Tensor Decompositions and Algorithms for Efficient Multidimensional Signal Processing

Liana Khamidullina

TECHNISCHE UNIVERSITÄT
ILMENAU

Communications Research
Laboratory

Dissertation zur Erlangung des akademischen Grades Doktor-Ingenieur
(Dr.-Ing.)

Tensor Decompositions and Algorithms for Efficient Multidimensional Signal Processing

Liana Khamidullina

Anfertigung im: Fachgebiet Nachrichtentechnik
Institut für Informationstechnik
Fakultät für Elektrotechnik und Informationstechnik

Gutachter: Univ.-Prof. Dr.-Ing. Martin Haardt
Prof. Dr. André L. F. de Almeida
Prof. Dr.-Ing. Marius Pesavento

Tag der wissenschaftlichen Aussprache: 28. November 2023

DOI: 10.22032/dbt.59389
URN: urn:nbn:de:gbv:ilm1-2023000390

Liana Khamidullina

Tensor Decompositions and Algorithms for Efficient Multidimensional Signal Processing

Dissertation zur Erlangung des akademischen Grades Doktor-Ingenieur (Dr.-Ing.)
2024 Reviewers: Univ.-Prof. Dr.-Ing. Martin Haardt, Prof. Dr. André L. F. de Almeida, and Prof. Dr.-Ing. Marius Pesavento
Supervisor: Univ.-Prof. Dr.-Ing. Martin Haardt

Technische Universität Ilmenau
Fachgebiet Nachrichtentechnik
Helmholtzplatz 2
98693 Ilmenau

Acknowledgments

I want to express my deepest gratitude to my supervisor, Prof. Martin Haardt, for the opportunity to work on this thesis, for constant guidance and creative ideas. Your passion for research is inspiring. Thank you for your advice and discussions that helped me grow as a researcher and significantly improved my work.

Likewise, I want to express my gratitude to Prof. Andre L. F. de Almeida for the cooperation, ideas, and discussions online and in person over the past years. Thank you also for the friendliest atmosphere during our visits to Fortaleza and GTEL.

I want to thank Prof. Jens Haueisen for his collaboration on biomedical signal processing, explanations, and insights about the EEG-MEG data and results.

Moreover, I would like to thank Prof. Marius Pesavento for volunteering to review this thesis, as well as Prof. Jochen Seitz, Prof. Giovanni Del Galdo, and Prof. Dr. Gerald Schuller for forming my examination committee.

I sincerely thank my former and current colleagues, Marjan, Gerald, Mikus, Joseph, Damir, Mike, Bilal, Alexey, Marko, Kristina, Casey, Andreas, and Madlen, for the wonderful working environment. I want to give special thanks to Ivan and Yao for guiding me at the very beginning of my PhD journey, and to Alla for her support when I was finalizing my thesis. I want to thank Sepideh, Khaled, and Bruno, the people with whom I started this journey, for their kindest personalities and our coffee break conversations.

I also want to thank all my students, especially Gabriela and Ildar, for their collaboration on the implementation of the block-term decomposition algorithms.

Last but not least, I want to thank my friends, parents, and sister for their unconditional love and constant support throughout these years.

Abstract

Due to the extensive growth of big data applications, the widespread use of multisensor technologies, and the need for efficient data representations, multidimensional techniques are a primary tool for many signal processing applications. Multidimensional arrays or tensors allow a natural representation of high-dimensional data. Therefore, they are particularly suited for tasks involving multi-modal data sources such as biomedical sensor readings or multiple-input and multiple-output (MIMO) antenna arrays. While tensor-based techniques were still in their infancy several decades ago, nowadays, they have already proven their effectiveness in various applications.

There are many different tensor decompositions in the literature, and each finds use in diverse signal processing fields. In this thesis, we focus on two tensor factorization models: the rank-$(L_r, L_r, 1)$ Block-Term Decomposition (BTD) and the Multilinear Generalized Singular Value Decomposition (ML-GSVD) that we propose in this thesis.

The ML-GSVD is an extension of the Generalized Singular Value Decomposition (GSVD) of two matrices to the tensor case. The properties of the original matrix GSVD render it an attractive tool for different applications, including genomic signal processing, MIMO relaying, coordinated beamforming, physical layer security, and multiuser MIMO systems. Yet, since the GSVD is restricted to two matrices, its use in wireless communications is limited to two users. Furthermore, this also ties into the fact that the literature lacked the extension of the GSVD for more than two matrices that would also inherit the properties of the original decomposition. Therefore, in this thesis, we extend the GSVD of two matrices to the tensor case while preserving its orthogonality properties and demonstrate its efficient application to multi-user MIMO communication systems. We provide a detailed discussion of the ML-GSVD subspace structure and propose an algorithm to compute it.

Furthermore, we present three applications of the ML-GSVD in MIMO communication systems: multiuser downlink MIMO systems with joint unicast and multicast transmissions; non-orthogonal multiple access (NOMA); and multi-user MIMO broadcast systems with rate splitting at the transmitter. For these applications, we exploit the structure of the ML-GSVD with common and private subspaces and show how the factors of the ML-GSVD can be used for the design of the precoders and decoders.

In the other part of the thesis, we focus on the rank-$(L_r, L_r, 1)$ Block-Term Decomposition. In contrast to the more common Canonical Polyadic (CP) decomposition, the rank-$(L_r, L_r, 1)$ decomposition has not yet been investigated as extensively and still has unexplored areas, such as its efficient computation. This thesis provides the algorithms to calculate both single and coupled rank-$(L_r, L_r, 1)$ decompositions by exploiting the connection of the BTD and CP decompositions. The proposed SECSI-BTD (SEmi-algebraic framework for approximate Canonical polyadic decompositions via SImultaneous Matrix Diagonalizations) algorithm includes the initial calculation of the factor estimates, followed by clustering and refinement procedures that return the appropriate rank-$(L_r, L_r, 1)$ BTD terms. Moreover, we introduce a new approach to estimate the multilinear rank structure of the tensor based on the higher-order singular value decomposition (HOSVD) and k-means clustering. Since the proposed SECSI-BTD algorithm does not require a known rank structure but can still take advantage of the known ranks when available, it is more flexible than the existing techniques in the literature.

As an application of the coupled rank-$(L_r, L_r, 1)$ decomposition, we consider near-field localization in multi-static MIMO radar systems. We show how the BTD can be employed for parameter estimation in 3D space based on the exact spherical wavefront model.

Finally, we consider the application of the coupled rank-$(L_r, L_r, 1)$ BTD to the Electroencephalogram (EEG) and Magnetoencephalogram (MEG) recordings of somatosensory evoked electrical potentials (SEPs) and somatosensory evoked magnetic fields (SEFs) to separate the signal components related to the 200 Hz band activity. In contrast to state-of-the-art works on the EEG and MEG recordings, we perform the fusion of the complete data set, including the gradiometer measurements, i.e., yielding a coupled rank-$(L_r, L_r, 1)$ BTD of four tensors (EEG, MEG-MAG, MEG-GRAD1, and MEG-GRAD2).

Additionally, this thesis provides the background material on the fundamentals of multilinear algebra, reviews the basic matrix and tensor decompositions, and identifies future research directions.

Zusammenfassung

Aufgrund des starken Wachstums von Big-Data-Anwendungen, der weit verbreiteten Nutzung von Multisensortechnologien und der Notwendigkeit einer effizienten Datendarstellung sind mehrdimensionale Techniken ein primäres Werkzeug für viele Anwendungen der Signalverarbeitung. Mehrdimensionale Arrays oder Tensoren ermöglichen eine natürliche Darstellung hochdimensionaler Daten. Daher eignen sie sich besonders für Aufgaben mit multimodalen Datenquellen wie biomedizinischen Sensorwerten oder MIMO-Antennenarrays (Multiple Input and Multiple Output). Während tensorbasierte Techniken vor einigen Jahrzehnten noch in den Kinderschuhen steckten, haben sie heute bereits ihre Wirksamkeit in verschiedenen Anwendungsgebieten unter Beweis gestellt.

In der Literatur gibt es viele verschiedene Tensorzerlegungen, die jeweils in verschiedenen Bereichen der Signalverarbeitung Anwendung finden. In dieser Arbeit konzentrieren wir uns auf zwei Tensorfaktorisierungsmodelle: die Rang-$(L_r, L_r, 1)$ Block-Term Decomposition (BTD) und die Multilinear Generalized Singular Value Decomposition (ML-GSVD), die wir in dieser Arbeit vorschlagen.

Die ML-GSVD ist eine Erweiterung der Generalized Singular Value Decomposition (GSVD) zweier Matrizen auf den Tensorfall. Die Eigenschaften der ursprünglichen Matrix GSVD machen sie zu einem attraktiven Werkzeug für verschiedene Anwendungen, einschließlich genomischer Signalverarbeitung, MIMO-Relaying, koordinierter Strahlformung, Sicherheit der physikalischen Schicht und Mehrbenutzer-MIMO-Systemen. Da die GSVD jedoch auf zwei Matrizen beschränkt ist, ist ihre Verwendung in der drahtlosen Kommunikation auf zwei Teilnehmer beschränkt. Darüber hinaus hängt dies auch mit der Tatsache zusammen, dass in der Literatur eine Erweiterung der GSVD für mehr als zwei Matrizen fehlte, die auch die Eigenschaften der ursprünglichen Zerlegung erben würde. Daher erweitern wir in dieser Arbeit die GSVD zweier Matrizen auf den Tensorfall unter Beibehaltung ihrer Orthogonalitätseigenschaften und demonstrieren ihre effiziente Anwendung auf Mehrbenutzer-MIMO-Kommunikationssysteme. Wir bieten eine detaillierte Diskussion der ML-GSVD-Unterraumstruktur und schlagen einen Algorithmus zu ihrer Berechnung vor.

Darüber hinaus stellen wir drei Anwendungen der ML-GSVD in MIMO Kommunikationssystemen vor: Multiuser Downlink MIMO Systeme mit gemeinsamer Unicast- und Multicast-Übertragung; nicht-orthogonaler Mehrfachzugriff (NOMA); und Mehrbenutzer-MIMO-Broadcast-Systeme mit Ratenaufteilung am Sender (RSMA). Für diese Anwendungen nutzen wir die Struktur der ML-GSVD mit gemeinsamen und privaten Unterräumen und zeigen, wie die Faktoren des ML-GSVD für den Entwurf der Precoder und Decoder genutzt werden können.

Im anderen Teil der Arbeit konzentrieren wir uns auf die Rang-$(L_r, L_r, 1)$ Block-Term Decomposition. Im Gegensatz zur häufigeren Canonical Polyadic (CP)-Zerlegung wurde die Rang-$(L_r, L_r, 1)$ BTD noch nicht so umfassend untersucht und weist noch unerforschte Bereiche auf, beispielsweise ihre effiziente Berechnung. Diese Arbeit stellt die Algorithmen zur Berechnung sowohl einzelner als auch gekoppelter Rang-$(L_r, L_r, 1)$ Zerlegungen bereit, indem sie die Verbindungen der BTD- mit CP-Zerlegungen ausnutzt. Der vorgeschlagene SECSI-BTD-Algorithmus (SEmi-algebraic Framework for approximate Canonical polyadic decompositions via SImultaneous Matrix Diagonalizations) umfasst die anfängliche Berechnung der Faktorschätzungen, gefolgt von Clustering- und Verfeinerungsverfahren, die den entsprechenden Rang det BTD-Terme zurückgeben. Darüber hinaus stellen wir einen neuen Ansatz zur Schätzung der multilinearen Rangstruktur des Tensors vor, der auf der Singulärwertzerlegung höherer Ordnung (HOSVD) und k-Means-Clustering basiert. Da der vorgeschlagene SECSI-BTD-Algorithmus keine bekannte Rangstruktur erfordert, aber dennoch die bekannten Ränge nutzen kann, sofern verfügbar, ist er flexibler als die in der Literatur vorhandenen Techniken.

Als Anwendung der gekoppelten Rang-$(L_r, L_r, 1)$ Zerlegung betrachten wir die Nahfeldlokalisierung in multistatischen MIMO-Radarsystemen. Wir zeigen, wie die BTD zur Parameterschätzung im 3D-Raum basierend auf dem exakten sphärischen Wellenfrontmodell verwendet werden kann.

Abschließend betrachten wir die Anwendung des gekoppelten Rang-$(L_r, L_r, 1)$ BTD auf die Elektroenzephalogramm- (EEG) und Magnetoenzephalogramm- (MEG) Aufzeichnungen somatosensorisch evozierter elektrischer Potentiale (SEPs) und somatosensorisch evozierter Magnetfelder (SEFs), um die damit verbundenen Signalkomponenten im 200 Hz Band zu trennen. Im Gegensatz zu aktuellen Arbeiten zu diesen Daten faktorisieren wir gemeinsam den gesamten EEG-MEG-Datensatz, einschließlich der Gradiometermessungen, d. h. wir erhalten eine gekoppelte Rang-$(L_r, L_r, 1)$ BTD von vier Tensoren (EEG, MEG-MAG, MEG-GRAD1 und MEG-GRAD2).

Darüber hinaus liefert diese Arbeit Hintergrundmaterial zu den Grundlagen der multilinearen Algebra, gibt einen Überblick über die grundlegenden Matrix- und Tensorzerlegungen und identifiziert zukünftige Forschungsrichtungen.

Contents

Introduction and scope of the thesis

1

1.1 Introduction and Motivation

Over the last few decades, the amount of data people and machines generate, process, store, and use has grown unimaginably. The extensive internet usage, availability, and continuous evolution of telecommunication technology significantly contributed to this data explosion. Moreover, its scientific and commercial value has also increased: the machine learning models that are trained on large data sets provide personalized services, produce art, can help recognize diseases or abnormalities in a body function, and many more. Yet, these remarkable advancements would not be possible without progress in signal processing techniques that allow extracting significant information from the data.

In many signal processing applications, data is commonly represented as matrices, capturing relationships along two dimensions. However, the advancements in multisensor technology and the emergence of large data sets have highlighted the limitations of standard two-dimensional models when dealing with the increasingly complex, high-dimensional data encountered in modern applications. The tensors, multidimensional equivalents of matrices [DDV00a], provide a more natural way to represent higher-order data without destroying its multilinear structure. For example, the received data in multiple-input multiple-output (MIMO) systems or electroencephalography (EEG) and magnetoencephalography (MEG) recordings are inherently three-dimensional (for instance, space × space × frequency [CAVP21] or time × sensors × trials [NLK+20; CRHH21]), and the employment of tensors in these fields shows a clear benefit. Moreover, the data that is not multidimensional by nature can still be tensorized [DD15] and benefit from tensor-based processing techniques such as various tensor decompositions.

Tensor factorizations allow the representation of data tensors by other matrices and/or tensors, often of much smaller dimensions than the original tensor, thereby enabling their efficient storage and manipulation.

While tensor-based techniques were still in their infancy several decades ago, nowadays, they have already proven their effectiveness in various applications and continue gaining attention from researchers in different fields, from image [SL01] and sensor data processing [SBG00] to machine learning [SDF+17] and big data analysis [SGCH19].

Wireless communication is one of the areas where tensor-based techniques have shown significant promise [CAVP21]. In modern wireless communication systems, multiple antennas are often used to transmit and receive signals simultaneously, which results in high-dimensional complex data and channel models. Higher order techniques allow exploiting the underlying structure of these data to improve the performance of communication systems, for instance, by enabling efficient MIMO channel estimation and precoding. Tensor-based algorithms have also found their use in relay systems [DYHL15], joint communications and sensing [PZNH19], satellite systems [dLA+19], and many others.

Considering more interdisciplinary fields, another great example is the application of tensor-based techniques to biomedical data processing. The extensive development of biomedical research, the broad spectrum and diversity of the data, and the requirement for fast analysis opened up many signal processing questions. The commonly large, inherently multidimensional, and highly noise-corrupted data require efficient processing solutions developed at the intersection of biomedical engineering and data analysis. Higher-order techniques have already shown a high potential in this area. Tensor representation allows capturing the multidimensional structure of these signals and a more accurate and robust feature extraction. For example, when applied to EEG-MEG signals, tensor-decompositions enable artifact detection, feature extraction, and can serve as a denoising tool or provide a low-dimensional input for further processing using deep learning [TJR19].

Probably the most prominent and extensively used tensor decompositions are the higher-order singular value decomposition (HOSVD) [DDV00a; Tuc66] and the canonical polyadic decomposition (CPD), sometimes also referred to as CANDECOMP (canonical decomposition) [CC70] or PARAFAC (parallel factors) [Har70]. Their applications include statistics, communications, localization, biomedical signal processing, source separation, and others [KB09; SDF+17].

Generally, tensor decompositions can be considered as multidimensional extensions of matrix factorizations, such as eigendecomposition and singular value decomposition (SVD). However, unlike the matrix methods, tensor decompositions offer uniqueness under mild and natural conditions. Moreover, the variety of higher-order decom-

positions and their flexibility in the choice of constraints allow finding and adapting the decomposition to specific data. This, in turn, leads to an improved performance in, for instance, feature extraction compared to the matrix-based methods.

As previously noted, tensors have a high potential in different areas of signal processing and optimization: they can preserve the data structure, enable lower-rank compact representations, and can be employed in independent component analysis, deep learning, and wireless systems optimization. Yet, various challenges still have to be addressed when going toward higher dimensions. These include, for example, the efficiency, reliability, and complexity of the algorithms in the scaled-up settings of future wireless communication systems with their huge data rates and extremely large antenna arrays.

There are many different tensor decompositions in the literature, and each finds use in diverse signal processing fields. In this thesis, we focus on two tensor factorization models: the rank-$(L_r, L_r, 1)$ block-term decomposition (BTD) and the multilinear generalized singular value decomposition (ML-GSVD) that we propose in this thesis. The latter, the ML-GSVD, extends the generalized singular value decomposition (GSVD) [PS81; Van76] of two matrices to the tensor case. The properties of the original matrix GSVD render it an attractive tool for different applications, including genomic signal processing, MIMO relaying, coordinated beamforming, physical layer security, and multi-user MIMO systems. Yet, since the GSVD is restricted to two matrices, its use in wireless communications is limited to two users. Furthermore, this also ties into the fact that the literature lacked the extension of the GSVD for more than two matrices that would also inherit the properties of the original decomposition. Therefore, in this thesis, we extend the GSVD of two matrices to the tensor case while preserving its orthogonality properties and demonstrate its efficient application to multi-user MIMO communication systems.

In the other part of the thesis, we focus on the rank-$(L_r, L_r, 1)$ block-term decomposition. In contrast to the more common canonical polyadic (CP) decomposition, the rank-$(L_r, L_r, 1)$ decomposition has not yet been investigated as extensively and still has unexplored areas, such as its efficient computation. This thesis provides the algorithms to calculate both single and coupled rank-$(L_r, L_r, 1)$ decompositions by exploiting the connection of the BTD and CP decompositions. Moreover, we introduce the application of the BTD to near-field localization in multi-static MIMO radar systems and to the analysis of EEG and MEG recordings.

1.2 Outline

This thesis contains four main parts: (I) an introduction to fundamental concepts and notations used in the thesis; (II,III) the achieved scientific contributions in two tensor decomposition domains; and (IV) conclusions with a discussion on the future research directions. The complementary materials are provided in the appendix (V).

The main goal of Part I is to familiarize the readers, who are new to the subject, with the essentials of tensor-based signal processing and to revise them for the knowledgeable ones.

Chapter 2 of this part provides the background material on the basics of multilinear algebra and the notations used in the thesis. Moreover, it reviews the relevant matrix and tensor decompositions. Section 2.1 introduces the notations and essential definitions of tensor algebra, such as the tensor rank, slices and fibers of a tensor, and the concept of unfoldings. Section 2.2 focuses on some of the fundamental matrix and tensor factorizations. Specifically, we review the following decompositions

- *Singular Value Decomposition (SVD).* We start our review of essentials by summarizing the singular value decomposition and its truncated and economy-size versions in Section 2.2.1.
- *Generalized Singular Value Decomposition (GSVD).* Section 2.2.2 gives a short summary of the Generalized Singular Value Decomposition of two matrices. In Chapter 3, we propose an extension of the GSVD to factorize a set of $K \geq 2$ matrices.
- *Tucker and Higher-Order Singular Value Decompositions (HOSVD).* Section 2.2.3 reviews the Tucker decomposition and its most common representation, the Higher-Order Singular Value Decomposition. We also briefly cover the truncated model and the computation techniques.
- *Canonical Polyadic Decomposition (CPD).* In Section 2.2.4, we focus on the Canonical Polyadic Decomposition, another primary tensor factorization. In addition to the representation of the CPD and its computation, this section also provides the essentials on the uniqueness of the decomposition and the ambiguities, such as matrix product, permutation, and scaling ambiguities.
- *PARAFAC2.* Section 2.2.5 is devoted to the PARAFAC2 (parallel factors 2) decomposition to factorize a set of matrices with one common dimension. We cover the uniqueness properties of the decomposition, some of the applications, and

the computation. We exploit the PARAFAC2 decomposition later in Chapter 3 as it has some similarities to our proposed ML-GSVD.

- *Block-Term Decompositions (BTD).* Section 2.2.6 revises three different types of block-term decompositions, the decomposition in rank-$(L_r, L_r, 1)$ terms, the decomposition in rank-(L_r, M_r, N_r) terms, and the type-2 decomposition in rank-$(L_r, M_r, \cdot)$ terms. We provide the tensor and matrix representations that are utilized for their computation. We will go into more detail on the decomposition in rank-$(L_r, L_r, 1)$ terms in Chapter 7.

- *Coupled decompositions.* In Section 2.2.7 we briefly discuss another category of factorizations, the coupled tensor decompositions, an effective data fusion tool.

Part II tackles this thesis' goal to extend the GSVD of two matrices to the tensor case while preserving its original properties and to demonstrate its efficient application to multi-user MIMO communication systems.

Generally, the chapters in Parts II and III are divided into four main sections. We start the chapter with an introduction that provides a state-the-art review and summarizes the contributions. Then, we present the data model, the problem, and the proposed solution. Next, we provide the numerical simulation results, and summarize the results in the conclusions section.

In Chapter 3, we introduce a multilinear generalized singular value decomposition (ML-GSVD) as an extension of the generalized singular value decomposition (GSVD) of two matrices to higher orders. The proposed decomposition allows a joint factorization of 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 the orthogonality of the second-mode factor matrices and the subspace structure of the third-mode factor matrices. The subspace structure of the decomposition depends on the dimensionality and the ranks of the slices of the decomposed tensor. We will discuss it in detail and introduce three different cases of the ML-GSVD depending on the existence of common and private subspaces. For the computation of the ML-GSVD, we introduce an ALS-based algorithm, which has been inspired by PARAFAC2 decomposition algorithms. Moreover, we briefly describe some of the applications of the proposed ML-GSVD, which we examine in more detail in the subsequent chapters.

In Chapter 4, we consider an application of the proposed ML-GSVD to multi-user downlink MIMO orthogonal frequency division multiplexing (OFDM) systems with joint unicast and multicast transmissions. We show how the common and individual

factor matrices of the ML-GSVD can be utilized as the precoding and decoding matrices, respectively. We also demonstrate that the three cases of the decomposition correspond to the transmission of private or common messages (or both). The simulation results show that the ML-GSVD-based precoding scheme outperforms the reference multicast and unicast precoding techniques in terms of the sum rate [KAH22].

> Chapters 3 and 4 are based on the following journal paper: L. Khamidullina, A. L. F. de Almeida, and M. Haardt, "Multilinear Generalized Singular Value Decomposition (ML-GSVD) and Its Application to Multiuser MIMO Systems", *IEEE Transactions on Signal Processing*, vol. 70, 2022 [KAH22].

In Chapter 5 of the thesis, we focus on non-orthogonal multiple access (NOMA), which has been considered a promising technique for 5G and beyond wireless networks. We consider a power-domain downlink MIMO-NOMA system with an arbitrary number of users and propose the design of the precoding and decoding matrices based on the ML-GSVD that we have presented in Chapter 3. Moreover, we demonstrate how the generalized singular values of the ML-GSVD can be used for power allocation. We also compare the proposed MIMO-NOMA scheme with orthogonal multiple access (OMA) techniques and provide various numerical results [KAH21].

> Chapter 5 is based on the following peer-reviewed conference publication: L. Khamidullina, A. L. F. de Almeida, and M. Haardt, "ML-GSVD-based MIMO-NOMA Networks", in *Proc. 25th International ITG Workshop on Smart Antennas (WSA 2021)*, Nov. 2021 [KAH21].

In Chapter 6, we explore multi-user multiple-input multiple-output (MU-MIMO) broadcast systems with rate splitting at the transmitter. We propose a precoder design that applies to both underloaded and overloaded communication systems and supports the transmission of multiple common and private streams. We show how the generalized singular value and multilinear generalized singular value decompositions can be used to define the number of common and private streams and how to adjust the message split. Additionally, we present transmit precoding and receive combining designs that allow the simultaneous transmission of common and private streams but do not require successive interference cancellation (SIC) at the receivers and can be used in cases where the total number of streams does not exceed the number of transmit antennas.

Chapter 6 is based on the following peer-reviewed conference publication: L. Khamidullina, A. L. F. de Almeida, and M. Haardt, "Rate Splitting and Precoding Strategies for Multi-User MIMO Broadcast Channels with Common and Private Streams", in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2023)*, 2023 [KAH23].

Part III of the thesis is dedicated to another type of tensor decompositions: multilinear-rank block-term decompositions. If the ML-GSVD can be considered more like a "structure revealing" factorization (as the full-size SVD or HOSVD), the block-term decompositions, similar to, for example, the approximate CPD or truncated HOSVD of the noise-corrupted data, belong to the "low-rank (compact) models" that allow extracting the significant information from the data. Therefore, we use it in the context of denoising and feature extraction. In general, the block-term decompositions have not yet been investigated as extensively as, for example, the CP decomposition. In Part III, we concentrate on one particular type of BTD, the rank-$(L_r, L_r, 1)$ decomposition and its applications. We provide the efficient computation algorithms for both versions, the single BTD and the coupled BTD. Moreover, we demonstrate its application to localization problems and EEG-MEG data analysis.

In Chapter 7, we introduce the SECSI-BTD framework, which exploits the connection between Canonical Polyadic and rank-$(L_r, L_r, 1)$ decompositions to estimate the block-terms of the rank-$(L_r, L_r, 1)$ BTD. The proposed SECSI-BTD algorithm includes the initial calculation of the factor estimates using the semi-algebraic framework for approximate canonical polyadic decompositions via simultaneous matrix diagonalizations (SECSI), followed by clustering and refinement procedures that return the appropriate rank-$(L_r, L_r, 1)$ BTD terms. Moreover, we introduce a new approach to estimate the multilinear rank structure of the tensor based on the HOSVD and k-means clustering. Since the proposed SECSI-BTD algorithm does not require a known rank structure but can still take advantage of the known ranks when available, it is more flexible than the existing techniques in the literature. Additionally, our algorithm does not require multiple initializations, and the simulation results show that it provides more accurate results and a better convergence behavior for an extensive range of SNRs (signal-to-noise ratio).

Chapter 7 is based on the following journal paper: L. Khamidullina, G. Seidl, I. A. Podkurkov, A. A. Korobkov, and M. Haardt, "Enhanced Solutions for the Block-Term Decomposition in Rank-$(L_r, L_r, 1)$ Terms", *IEEE Transactions on Signal Processing*, vol. 71, 2023 [KSP+23].

The coupled decompositions prove to be attractive in many signal processing fields and data fusion tasks where multiple sets of data exhibit common properties, such as joint EEG-MEG analysis and data fusion. Therefore, in Chapter 8, we focus on the coupled extension of the rank-$(L_r, L_r, 1)$ decomposition. We present an algorithm to calculate the coupled rank-$(L_r, L_r, 1)$ decomposition of multiple three-dimensional tensors with a coupled mode and possibly an unknown rank structure. The proposed approach is composed of three main parts. The first is an extension of the linear regression of global eigenvalues (LaRGE) [KDHH21] technique to estimate the number of blocks and the sum of the multilinear ranks in the coupled tensors. The second part accounts for the calculation of the preliminary multilinear factors of the coupled BTD by means of the coupled SECSI. The last part contains the final refinement procedures that return the estimated BTD factors. Moreover, we provide some synthetic data simulations showing that the proposed algorithm demonstrates more stable and accurate results than the schemes from the literature.

> Chapter 8 is based on the following peer-reviewed conference publication: I. Safiullin, L. Khamidullina, A. A. Korobkov, and M. Haardt, "Enhanced Computation of the Coupled Block-Term Decomposition in Multilinear Rank Terms", in *Proc. IEEE 12th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, Trondheim, Norway, Jun. 2022 [SKKH22].

In Chapter 9, we present a high-resolution coupled rank-$(L_r, L_r, 1)$ decomposition-based near-field localization scheme for multi-static MIMO radar systems. The proposed COBRAS (COupled Block-term decomposition for multi-static RAdar Systems) algorithm estimates the target location parameters in 3D space based on the exact wavefront model and is applicable to arbitrary array geometries. Compared to the far-field models, the exact near-field wavefront model allows exploiting the distance information for high-accuracy positioning. Moreover, we consider a system with massive antenna arrays, which increases the Fresnel region and renders the near-field assumption applicable in practical scenarios. The COBRAS algorithm includes the initial tensor decomposition of the data and further postprocessing steps that allow extracting the location parameters. The simulation results demonstrate that the employment of coupling vastly improves the localization performance compared to the non-coupled solutions. Moreover, we show that the coupled BTD algorithm proposed in Chapter 8 outperforms its alternatives from the literature.

Chapter 10 is devoted to one of the common and promising applications of tensor decompositions, joint EEG and MEG data processing. In particular, we show how the coupled rank-$(L_r, L_r, 1)$ decomposition can be applied to the EEG-MEG recordings

of the somatosensory evoked electrical potentials (SEPs) and somatosensory evoked magnetic fields (SEFs) to separate the signal components related to the 200 Hz band activity. In our study, we use the SEP and SEF recordings evoked through electrical median nerve stimulations conducted on eight healthy volunteers. In this study, we extend the work in [CRHH21], where the authors utilized the multilinear rank-$(L_r, L_r, 1)$ decomposition to process EEG and MEG magnetometer (MAG) recordings. In contrast to the aforementioned paper, we focus on the complete data set, which additionally includes the gradiometer (GRAD) measurements, resulting in a coupled decomposition of four tensors: EEG, MEG-MAG, MEG-GRAD-1, and MEG-GRAD-2. Furthermore, we address the problem of the proper initialization of the coupled rank-$(L_r, L_r, 1)$ decomposition and use the SECSI-based approach that we propose in Chapter 8 instead of multiple random initializations employed in [CRHH21]. In essence, this chapter validates the application of the coupled BTD algorithm proposed in Chapter 8 for the analysis of neural oscillations via an efficient data fusion of four tensors which, in turn, holds the potential to provide enhanced insights into brain function and dynamics.

Part IV summarizes the findings of this thesis and identifies future research directions. Finally, Part V contains the appendix with the lists of acronyms, figures, notations, and some additional materials such as derivations, selected matrix products definitions, and notable references.

Part I.

Background

Multilinear Algebra Fundamentals

2

For the most part, this thesis focuses on multi-dimensional signal processing, more precisely, on tensor decompositions and their communication, sensing, and biomedical applications. Therefore, before transitioning to the core content, let us begin the exploration by briefly revising multilinear algebra's fundamental concepts and notations used in this thesis, building a foundation for the following chapters.

2.1 Basic concepts and notation

In the context of this thesis, we use the term (higher-order) "tensor" to refer to an array of numerical values in three or more dimensions. The tensors can also be intuitively imagined as multidimensional analogs of matrices (second-order tensors) or vectors (first-order tensors) [DDV00a], (Figure 2.1). The order of a tensor corresponds to the number of its dimensions or, equivalently, its modes or ways.

In the following, we provide some basic definitions and notations that generally apply to n-way tensors. Since in this thesis we primarily focus on 3-way arrays, some concepts are demonstrated using third-order tensors as an example.

Notation. Throughout this thesis, we denote the matrices and vectors by upper-case ($\boldsymbol{A}$) and lower-case ($\boldsymbol{a}$) bold-faced letters, respectively. Bold-faced calligraphic letters denote tensors ($\boldsymbol{\mathcal{A}}$). The superscripts $\{\cdot\}^{\mathrm{T}}$, $\{\cdot\}^{\mathrm{H}}$, $\{\cdot\}^{-1}$, and $\{\cdot\}^{+}$ denote the transpose, Hermitian transpose, matrix inverse, and Moore-Penrose pseudoinverse, respectively. Moreover, the mth column and the lth row of $\boldsymbol{A}$ is denoted as $\boldsymbol{A}_{(:,m)}$ and $\boldsymbol{A}_{(l,:)}$, respectively. The field of real ($\mathbb{R}$) or complex ($\mathbb{C}$) numbers is represented by $\mathbb{F}$ when the difference is unimportant. We use $\otimes$, $\diamond$, and $\circ$ to denote the Kronecker, Khatri-Rao (column-wise Kronecker), and outer products, respectively. The operator $\|\cdot\|_{\mathrm{F}}$ denotes the Frobenius norm, and $\|\cdot\|$ denotes the Euclidean norm. The operator $\mathrm{diag}(\cdot)$ denotes the construction of a diagonal matrix with diagonal elements being the entries of the input vector, whereas $\mathrm{bdiag}(\cdot)$ is the operation of constructing a block-diagonal

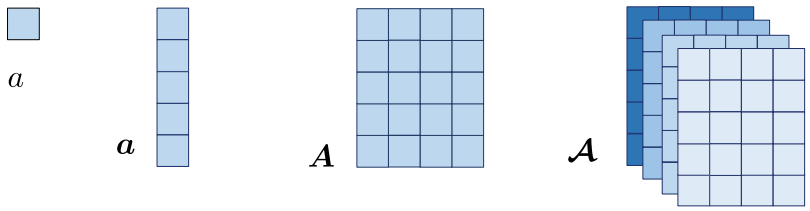


Figure 2.1. Left to right: scalar, vector, matrix, and a third-order tensor.

matrix with the input matrices on the main diagonal. The matrix $\boldsymbol{O}_{M\times N}$ denotes the $M \times N$ matrix of zeros. An identity matrix of size $d \times d$ is denoted as $\boldsymbol{I}_d$, and a column vector of ones of size R is denoted as $\mathbf{1}_R$. The mathematical notation used in this thesis is also summarized in the Section "Symbols & Notation" on page 187.

Rank-1 tensor. As the rank-1 matrix can be represented as an outer product of two vectors

$$\boldsymbol{A} = \boldsymbol{a}_1 \circ \boldsymbol{a}_2, \tag{2.1}$$

the rank-1 N-way tensor $\boldsymbol{\mathcal{A}} \in \mathbb{F}^{I_1\times\ldots\times I_N}$ can be viewed as an outer product N vectors [KB09]

$$\boldsymbol{\mathcal{A}} = \boldsymbol{a}_1 \circ \boldsymbol{a}_2 \circ \ldots \circ \boldsymbol{a}_N. \tag{2.2}$$

Fibers. Analogously to matrix rows and columns, the n-mode vectors or fibers of a higher-order tensor are obtained by fixing every index but one [KB09]. A 1-mode fiber of a matrix is a matrix column, and a 2-mode fiber is its row. Taking third-order tensors as an example, in addition to the 1- and 2-mode fibers, they have 3-mode (tube) fibers [KB09]. The third-order tensors in terms of the different mode fibers are illustrated in Figures 2.3 and 2.4. When extracted from the tensor, fibers are assumed to be oriented as column vectors.

Slices. The slices of a tensor are the matrices defined by fixing all but two indices in a tensor [KB09]. A third-order tensor $\boldsymbol{\mathcal{A}} \in \mathbb{F}^{I\times J\times K}$ has horizontal, lateral, and frontal slices (see Figure 2.2) commonly denoted as $\boldsymbol{\mathcal{A}}_{(i,:,:)}$, $\boldsymbol{\mathcal{A}}_{(:,j,:)}$, and $\boldsymbol{\mathcal{A}}_{(:,:,k)}$, respectively. In the thesis, we also use $\boldsymbol{A}_k$ to denote the frontal slices of $\boldsymbol{\mathcal{A}} \in \mathbb{F}^{I\times J\times K}$.

Elements. The element $(i_1,\ldots,i_N)$ of an N-way tensor $\boldsymbol{\mathcal{A}} \in \mathbb{F}^{I_1\times\ldots\times I_N}$ is denoted as $\boldsymbol{\mathcal{A}}_{(i_1,\ldots,i_N)}$, or equivalently, as $a_{i_1,\ldots,i_N}$. For the third-order tensor $\boldsymbol{\mathcal{A}} \in \mathbb{F}^{I\times J\times K}$, its (i,j,k)th element is denoted as $\boldsymbol{\mathcal{A}}_{(i,j,k)} = a_{i,j,k}$

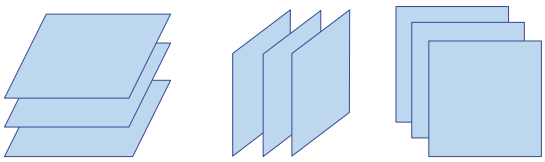
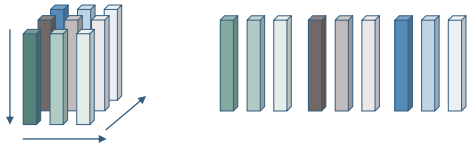
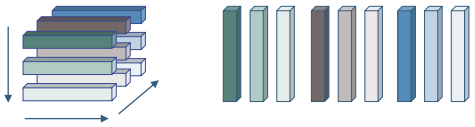


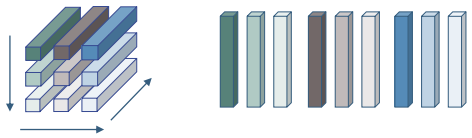
Figure 2.2. Slices of a third-order tensor. Left to right: horizontal, lateral, and frontal slices.



(a) 1-mode (column) fibers and 1-mode unfolding.



(b) 2-mode (row) fibers and 2-mode unfolding.



(c) 3-mode (tube) fibers and 3-mode unfolding.

Figure 2.3. Fibers and unfoldings of a $3 \times 3 \times 3$ tensor in the forward column ordering.

Unfolding. Furthermore, the process of matricization, denoting the transformation of a tensor into a matrix, is alternatively referred to as unfolding or flattening [KB09; DDV00a]. For the third-order tensors $\mathcal{A} \in \mathbb{F}^{I \times J \times K}$, we can define the 1-mode, 2-mode, and 3-mode unfoldings. These are represented as $[\mathcal{A}]_{(1)} \in \mathbb{F}^{I \times JK}$, $[\mathcal{A}]_{(2)} \in \mathbb{F}^{J \times IK}$, and $[\mathcal{A}]_{(3)} \in \mathbb{F}^{K \times IJ}$. The unfolding process involves arranging the respective n-mode vectors into matrices. Different orders of the n-mode vectors result in varying tensor unfoldings. Following the *forward column ordering*, the n-mode unfolding of an N-way tensor $\mathcal{A} \in \mathbb{F}^{I_1 \times \ldots \times I_N}$ involves arranging the n-mode vectors in ascending order. Consequently, we start with varying the first index, keep increasing up to the $(n-1)$th index, then continue with the $(n+1)$th up to the Nth index. An illustration of the three distinct unfoldings of a 3-way tensor in the forward ordering is shown in Figure 2.3.

An alternative ordering method is the *reverse cyclical ordering* [DDV00a], where the unfolding $\left[\mathcal{A}\right]_{(n)}$ of an N-way tensor follows the sequence that starts with the $(n-1)$th index, keeps decreasing to the first index, continue with the Nth index, and proceeds backward until reaching the $(n+1)$th index (visualized in Figure 2.4).

The different ordering approaches lead to differing definitions of tensor operations. Therefore, meticulous attention must be given to the chosen ordering, maintaining consistency when deriving new concepts or outcomes. Throughout this thesis, we use the reverse cyclical ordering unless explicitly stated otherwise.

***n*-ranks.** The n-rank of a tensor $\mathcal{A}$ is equal to the rank of its n-mode unfolding

$$\text{rank}_n\left(\mathcal{A}\right) = \text{rank}\left(\left[\mathcal{A}\right]_{(n)}\right). \tag{2.3}$$

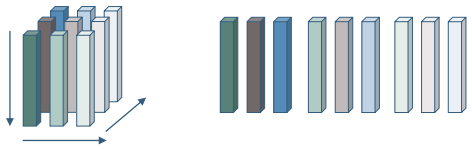
In contrast to matrices, where the column and the row ranks are equal, the tensors' n-ranks may differ.

Generalized unfolding. In addition to single n-mode unfoldings, we can define the generalized matrix unfoldings using two subsets of any of the N dimensions [RSH12; LA11]. For instance, the set of modes $(1,2,\ldots,N)$ of an N-way tensor $\mathcal{A}$ can be divided into two non-overlapping, P and $N-P$ dimensional subsets, where the indices contained in the first subset vary along the rows and the indices contained in second subset vary along the columns [RSH12]. However, since this type of unfoldings is not common, and in this thesis, we focus on 3-dimensional tensors, we refer the reader to [RSH12] for further information on the generalized unfoldings. For the tensors of order three, the generalized unfoldings correspond to n-mode unfoldings.

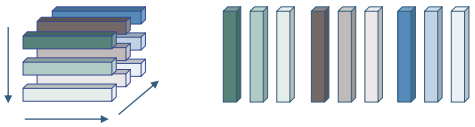
Super-diagonal tensor. An N-dimensional tensor $\mathcal{D} \in \mathbb{R}^{R\times R\times\ldots\times R}$ is called super-diagonal if its elements are non-zero only if all N indices are equal and zeros otherwise. We use $\mathcal{I}_{N,R}$ to denote a super-diagonal tensor with ones along its super-diagonal. Figure 2.5 ilustrates the super-diagonal tensor $\mathcal{I}_{3,5}$.

Vectorization. Vectorization of a tensor is a linear transformation that converts a tensor into a column vector. The $\text{vec}\left(\mathcal{A}\right)$ can be obtained by stacking the n-mode fibers of $\mathcal{A}$ into one column

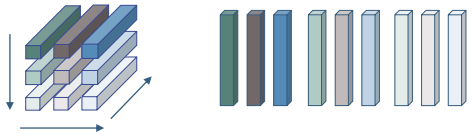
$$\text{vec}\left(\mathcal{A}\right) = \text{vec}\left(\left[\mathcal{A}\right]_{(n)}\right) \tag{2.4}$$



(a) 1-mode (column) fibers and 1-mode unfolding.



(b) 2-mode (row) fibers and 2-mode unfolding.



(c) 3-mode (tube) fibers and 3-mode unfolding.

Figure 2.4. Fibers and unfoldings of a $3 \times 3 \times 3$ tensor in the reverse cyclical ordering.

Similarly to the unfoldings, the ordering of the fibers should be consistent through the derivations [KB09].

Norm of a tensor. We denote the Frobenius norm of an N-way tensor $\mathcal{A} \in \mathbb{F}^{I_1 \times I_2 \ldots \times I_N}$ by $\|\mathcal{A}\|_{\mathrm{F}}$ defined as the square root of the sum of the squares of all its elements, the 2-norm of the vectorized tensor, or equally, as the Frobenius norm of an arbitrary n-mode unfolding of the tensor [KB09; DDV00a]

$$\|\mathcal{A}\|_{\mathrm{F}} = \sqrt{\sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} \left|\mathcal{A}_{(i_1,i_2,\ldots,i_N)}\right|^2} = \|\mathrm{vec}\,(\mathcal{A})\| = \left\|\left[\mathcal{A}\right]_{(n)}\right\|_{\mathrm{F}}. \tag{2.5}$$

The n-mode product. The n-mode product between a tensor $\mathcal{A} \in \mathbb{F}^{I_1 \times I_2 \ldots \times I_N}$ and a matrix $\boldsymbol{U} \in \mathbb{F}^{J \times I_n}$ is denoted by

$$\mathcal{A} \times_n \boldsymbol{U} \in \mathbb{F}^{I_1 \times \ldots \times I_{n-1} \times J \times I_{n+1} \ldots \times I_N}. \tag{2.6}$$

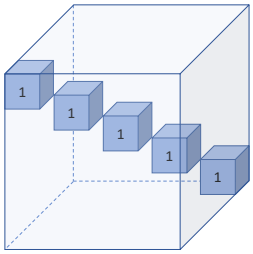


Figure 2.5. Super-diagonal tensor $\mathcal{I}_{3,5}$ with ones along its super-diagonal.

In terms of the element-wise notation, it can be expressed as

$$(\mathcal{A} \times_n \boldsymbol{U})_{(i_1,i_2,\ldots,i_{n-1},j,i_{n+1},\ldots,i_N)} = \sum_{i_n=1}^{I_n} \mathcal{A}_{(i_1,i_2,\ldots,i_N)} \cdot \boldsymbol{U}_{(j,i_n)}, \tag{2.7}$$

and terms of an n-mode tensor unfoldings as

$$[\mathcal{A} \times_n \boldsymbol{U}]_{(n)} = \boldsymbol{U} \cdot [\mathcal{A}]_{(n)}. \tag{2.8}$$

Considering a third-order tensor $\mathcal{A} \in \mathbb{F}^{I \times J \times K}$ as an example, the three n-mode products between tensor $\mathcal{A}$ and the matrices $\boldsymbol{U}_1 \in \mathbb{F}^{M_1 \times I}$, $\boldsymbol{U}_2 \in \mathbb{F}^{M_2 \times J}$, and $\boldsymbol{U}_3 \in \mathbb{F}^{M_3 \times K}$ can be expressed as follows

$$\begin{aligned} &\mathcal{B} = \mathcal{A} \times_1 \boldsymbol{U}_1 \in \mathbb{F}^{M_1 \times J \times K}, \\ &\updownarrow \\ &\mathcal{B}_{(i,j,k)} = \sum_{\ell=1}^{I} \mathcal{A}_{(\ell,j,k)} \cdot \boldsymbol{U}_{1,(i,\ell)}, \\ &\updownarrow \\ &[\mathcal{B}]_{(1)} = \boldsymbol{U}_1 \cdot [\mathcal{A}]_{(1)}, \end{aligned} \tag{2.9}$$

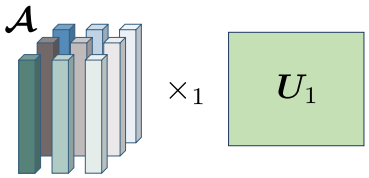


Figure 2.6. 1-mode product.

$$\begin{aligned} &\mathcal{B} = \mathcal{A} \times_2 \boldsymbol{U}_2 \in \mathbb{F}^{I \times M_2 \times K}, \\ &\updownarrow \\ &\mathcal{B}_{(i,j,k)} = \sum_{\ell=1}^{J} \mathcal{A}_{(i,\ell,k)} \cdot \boldsymbol{U}_{2,(j,\ell)}, \\ &\updownarrow \\ &[\mathcal{B}]_{(2)} = \boldsymbol{U}_2 \cdot [\mathcal{A}]_{(2)}, \end{aligned} \tag{2.10}$$

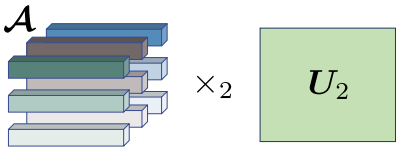


Figure 2.7. 2-mode product.

$$\begin{aligned}
&\mathcal{B} = \mathcal{A} \times_3 \boldsymbol{U}_3 \in \mathbb{F}^{I \times J \times M_3}, \\
&\updownarrow \\
&\mathcal{B}_{(i,j,k)} = \sum_{\ell=1}^{K} \mathcal{A}_{(i,j,\ell)} \cdot \boldsymbol{U}_{3,(k,\ell)}, \\
&\updownarrow \\
&[\mathcal{B}]_{(3)} = \boldsymbol{U}_3 \cdot [\mathcal{A}]_{(3)}.
\end{aligned} \tag{2.11}$$

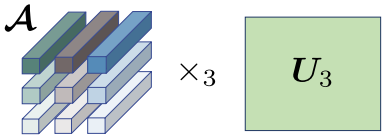


Figure 2.8. 3-mode product.

Properties of n-mode products and unfoldings. Given N-way tensors $\mathcal{A} \in \mathbb{F}^{I_1 \times \ldots \times I_N}$ and $\mathcal{B} \in \mathbb{F}^{J_1 \times \ldots \times J_N}$, and the matrices $\boldsymbol{U}_n \in \mathbb{F}^{I_n \times J_n}$, $n \in \{1, \ldots, N\}$, the n-mode unfolding of

$$\mathcal{A} = \mathcal{B} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \times_3 \cdots \times_N \boldsymbol{U}_N, \tag{2.12}$$

in reverse cyclical ordering, takes the form [DDV00a]

$$[\mathcal{A}]_{(n)} = \boldsymbol{U}_n \cdot [\mathcal{B}]_{(n)} \cdot \left(\boldsymbol{U}_{n+1} \otimes \cdots \otimes \boldsymbol{U}_N \otimes \boldsymbol{U}_1 \otimes \cdots \otimes \boldsymbol{U}_{n-1}\right)^{\mathrm{T}}, \tag{2.13}$$

and in forward ordering, it follows

$$[\mathcal{A}]_{(n)} = \boldsymbol{U}_n \cdot [\mathcal{B}]_{(n)} \cdot \left(\boldsymbol{U}_N \otimes \cdots \otimes \boldsymbol{U}_{n+1} \otimes \boldsymbol{U}_{n-1} \otimes \cdots \otimes \boldsymbol{U}_1\right)^{\mathrm{T}}. \tag{2.14}$$

Moreover, for the series of multiplications

$$\mathcal{X} \times_n \boldsymbol{U}_n \times_m \boldsymbol{V}_m = \mathcal{X} \times_m \boldsymbol{V}_m \times_n \boldsymbol{U}_n, \text{ for } m \neq n, \tag{2.15}$$

the order of multiplication along different modes is irrelevant. Furthermore, for the products in the same mode

$$\mathcal{X} \times_n \boldsymbol{U}_n \times_n \boldsymbol{V}_n = \mathcal{X} \times_n (\boldsymbol{V}_n \cdot \boldsymbol{U}_n), \tag{2.16}$$

where $\mathcal{X} \in \mathbb{F}^{I_1 \times \ldots \times I_N}$, $\boldsymbol{U}_n \in \mathbb{F}^{J_n \times I_n}$ and $\boldsymbol{V}_n \in \mathbb{F}^{N_n \times J_n}$ are the tensor and matrices of compatible dimensions [KB09].

Taking the third-order tensors $\mathcal{Y} \in \mathbb{F}^{I_1 \times I_2 \times I_3}$ and $\mathcal{X} \in \mathbb{F}^{J_1 \times J_2 \times J_3}$, and matrices $\boldsymbol{A} \in \mathbb{F}^{I_1 \times J_1}$, $\boldsymbol{B} \in \mathbb{F}^{I_2 \times J_2}$, and $\boldsymbol{C} \in \mathbb{F}^{I_3 \times J_3}$ as an example, if $\mathcal{Y} = \mathcal{X} \times_1 \boldsymbol{A} \times_2 \boldsymbol{B} \times_3 \boldsymbol{C}$, we can express the tensor unfoldings in reverse cyclical ordering as follows [KB09]

$$[\mathcal{Y}]_{(1)} = \boldsymbol{A} \cdot [\mathcal{X}]_{(1)} \cdot (\boldsymbol{B} \otimes \boldsymbol{C})^{\mathrm{T}}, \tag{2.17}$$

$$[\mathcal{Y}]_{(2)} = \boldsymbol{B} \cdot [\mathcal{X}]_{(2)} \cdot (\boldsymbol{C} \otimes \boldsymbol{A})^{\mathrm{T}}, \tag{2.18}$$

$$[\mathcal{Y}]_{(3)} = \boldsymbol{C} \cdot [\mathcal{X}]_{(3)} \cdot (\boldsymbol{A} \otimes \boldsymbol{B})^{\mathrm{T}}, \tag{2.19}$$

where the symbol $\otimes$ denotes the Kronecker product. For the definitions of Kronecker and other tensor-related products, see Appendix A.1.

2.2 Matrix and tensor decompositions

Compared to matrices, tensors allow capturing the original complex and high-dimensional structure of the data, such as those that are often encountered in real-life applications. Therefore, higher-dimensional decompositions have shown to be a powerful tool that can be effectively utilized in different areas of signal processing and optimization. Tensor decompositions play a significant role in modern data analysis and computation, providing mechanisms to uncover latent structures and relationships within complex data sets. These decompositions involve expressing a given tensor as a product of simpler and more interpretable matrices or tensors, thereby bringing complex and big data into a more comprehensible form. Many tensor decompositions, such as canonical polyadic decomposition (CPD) or higher-order singular value decomposition (HOSVD), expand the matrix decompositions to more dimensions. Therefore, in the following, we will review some matrix and tensor decompositions that we directly or indirectly touch upon in this thesis.

2.2.1 Singular Value Decomposition

Given a matrix $\boldsymbol{A} \in \mathbb{F}^{I \times J}$, the singular value decomposition (SVD) decomposes it into a product of three matrices

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{H}}, \tag{2.20}$$

where $\boldsymbol{U} \in \mathbb{F}^{I \times I}$ is a square unitary matrix, containing the left singular vectors of $\boldsymbol{A}$ as its columns, $\boldsymbol{\Sigma} \in \mathbb{F}^{I \times J}$ is the diagonal matrix of singular values, with r non-zero entries $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r$ along its diagonal, where r is the rank of $\boldsymbol{A}$, and $\boldsymbol{V} \in \mathbb{F}^{J \times J}$ is a square unitary matrix, containing the right singular vectors of $\boldsymbol{A}$ as its columns.

Since only the first r diagonal elements of $\boldsymbol{\Sigma}$ are non-zero, the SVD in (2.20) can be expressed in a more compact ("economy-size") form

$$\boldsymbol{A} = \boldsymbol{U}_{\mathrm{s}}\boldsymbol{\Sigma}_{\mathrm{s}}\boldsymbol{V}_{\mathrm{s}}^{\mathrm{H}}, \tag{2.21}$$

where the matrices $\boldsymbol{U}_{\mathrm{s}}$ and $\boldsymbol{V}_{\mathrm{s}}$ contain the first r columns of $\boldsymbol{U}$ and $\boldsymbol{V}$, respectively, and $\boldsymbol{\Sigma}_{\mathrm{s}} \in \mathbb{R}^{r \times r}$ is the upper-left submatrix of $\boldsymbol{\Sigma}$.

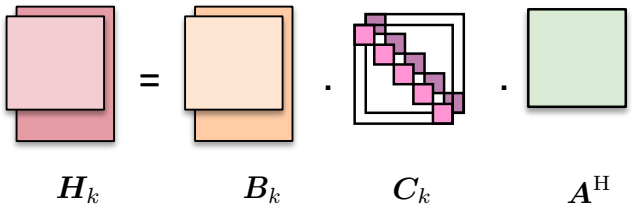


Figure 2.9. Generalized singular value decomposition (GSVD).

Moreover, the optimal rank-k approximation of $\boldsymbol{A}$ in the Frobenius norm is given by truncating the SVD to $k < \operatorname{rank}(\boldsymbol{A})$ leading columns in $\boldsymbol{U}$ and $\boldsymbol{V}$, and the $k \times k$ upper-left submatrix of $\boldsymbol{\Sigma}$ [EY36]

$$\arg\min_{\hat{\boldsymbol{A}}} \left\| \boldsymbol{A} - \hat{\boldsymbol{A}} \right\|_{\mathrm{F}} = \boldsymbol{U}_k \boldsymbol{\Sigma}_k \boldsymbol{V}_k^{\mathrm{H}}. \tag{2.22}$$

The SVD is a fundamental matrix factorization technique that plays a primary role in many applications, such as solving linear systems of equations, computing pseudoinverses, and performing principal component analysis. The tensor decompositions that we review in the following can be considered as multidimensional extensions of the SVD.

2.2.2 Generalized Singular Value Decomposition

Generalized singular value decomposition (GSVD) is an extension of the SVD for two matrices and is proposed in [PS81], [Van76], and [GVL96]. Let $\boldsymbol{H}_1 \in \mathbb{F}^{J_1 \times I}$ and $\boldsymbol{H}_2 \in \mathbb{F}^{J_2 \times I}$ 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

$$\begin{aligned} \boldsymbol{H}_1 &= \boldsymbol{B}_1 \cdot \boldsymbol{C}_1 \cdot \boldsymbol{A}^{\mathrm{H}}, \\ \boldsymbol{H}_2 &= \boldsymbol{B}_2 \cdot \boldsymbol{C}_2 \cdot \boldsymbol{A}^{\mathrm{H}}, \end{aligned} \tag{2.23}$$

where $\boldsymbol{B}_1 \in \mathbb{F}^{J_1 \times J_1}$ and $\boldsymbol{B}_2 \in \mathbb{F}^{J_2 \times J_2}$ have orthogonal columns, $\boldsymbol{A} \in \mathbb{F}^{I \times q}$ is non-singular and common for both matrices with $q = \operatorname{rank}\left(\left[\boldsymbol{H}_1^{\mathrm{H}}, \boldsymbol{H}_2^{\mathrm{H}} \right] \right)$. Moreover, $\boldsymbol{C}_1 \in \mathbb{R}^{J_1 \times q}$ and $\boldsymbol{C}_2 \in \mathbb{R}^{J_2 \times q}$ 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 $r = q - \operatorname{rank}(\boldsymbol{H}_2)$ and $s = \operatorname{rank}(\boldsymbol{H}_1) + \operatorname{rank}(\boldsymbol{H}_2) - q$, then, $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ have the following structures

$$\boldsymbol{C}_1 = \begin{bmatrix} \boldsymbol{O}_{(J_1-r-s)\times(q-r-s)} & & \\ & \hat{\boldsymbol{\Sigma}} & \\ & & \boldsymbol{I}_r \end{bmatrix}, \tag{2.24}$$

$$\boldsymbol{C}_2 = \begin{bmatrix} \boldsymbol{I}_{(q-r-s)} & & \\ & \hat{\boldsymbol{\Lambda}} & \\ & & \boldsymbol{O}_{(J_2-q+r)\times(r)} \end{bmatrix}, \tag{2.25}$$

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, $\hat{\boldsymbol{\Sigma}} = \operatorname{diag}(\sigma_1,\ldots,\sigma_s)$, $\hat{\boldsymbol{\Lambda}} = \operatorname{diag}(\lambda_1,\ldots,\lambda_s)$ are diagonal matrices, such that $0 < \sigma_n < 1, 0 < \lambda_n < 1$, and $\sigma_n^2 + \lambda_n^2 = 1$ for $n \in \{1,\ldots,s\}$. The Matlab implementation of the GSVD additionally provides the "economy-size" version of the GSVD, where $\boldsymbol{C}_1 \in \mathbb{R}^{\min(J_1,I)\times q}$ and $\boldsymbol{C}_2 \in \mathbb{R}^{\min(J_2,I)\times q}$. An example illustration of the "economy-size" GSVD for a case where $q = I = 5$ is given in Figure 2.9.

The matrices $\hat{\boldsymbol{\Sigma}}$, $\hat{\boldsymbol{\Lambda}}$, and the identity matrices in $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ allow us to distinguish the common subspaces (columns $(q-r-s+1,\ldots,q-r-s+s)$ corresponding to $\hat{\boldsymbol{\Sigma}}$ and $\hat{\boldsymbol{\Lambda}}$) and the private (columns corresponding to identity matrices in $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$) subspaces in $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$. Depending on the values of q, r, and s, the private or common subspaces can be empty. Moreover, note that the private subspaces correspond to generalized singular values that are either 0 or infinity.

Choosing $\boldsymbol{H}_2 = \boldsymbol{I}_I$ and $\boldsymbol{A} = \boldsymbol{B}_2$, we can obtain the SVD of $\boldsymbol{H}_1 = \boldsymbol{B}_1\boldsymbol{C}_1\boldsymbol{B}_2$. Furthermore, in the scenario where the matrices $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$ have full column rank, the generalized singular values are equal to the singular values of $\boldsymbol{H}_1(\boldsymbol{H}_2)^+ = \boldsymbol{B}_1\boldsymbol{C}_1\boldsymbol{A}^{\mathrm{H}}\left(\boldsymbol{A}^{\mathrm{H}}\right)^+(\boldsymbol{C}_2)^+\boldsymbol{B}_2^{\mathrm{H}} = \boldsymbol{B}_1\boldsymbol{C}_1(\boldsymbol{C}_2)^+\boldsymbol{B}_2^{\mathrm{H}} = \boldsymbol{B}_1\operatorname{diag}(\sigma_1/\lambda_1,\ldots,\sigma_I/\lambda_I)\boldsymbol{B}_2^{\mathrm{H}}$.

There are some small variations in the representation of the GSVD, for instance, in [Van76] 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 the cosine-sine decompositions. For more details on the GSVD and its applications, we refer the reader to [Van76; PS81], and [Bai92].

Part II of this thesis is devoted to an extension of GSVD, the multilinear generalized singular value decomposition (ML-GSVD), and its applications.

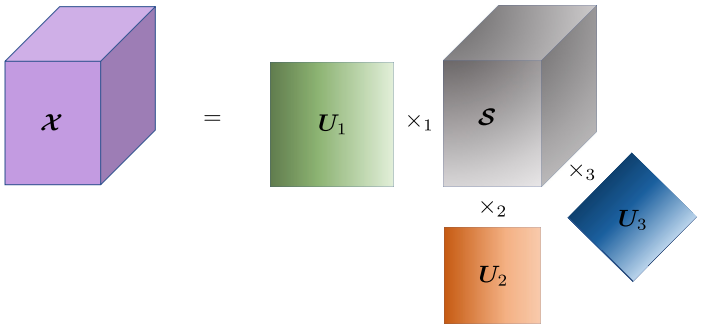


Figure 2.10. Higher-Order Singular Value Decomposition of a 3-way tensor.

2.2.3 Tucker and Higher-Order Singular Value Decompositions

The Tucker decomposition is a powerful extension of the classical matrix-based SVD designed to handle multi-dimensional data arrays. Given a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{F}^{I_1 \times I_2 \times \ldots \times I_N}$, the Tucker decomposition factorizes it into a core tensor $\boldsymbol{\mathcal{G}} \in \mathbb{F}^{P_1 \times \ldots \times P_N}$ and a set of factor matrices $\boldsymbol{V}_n \in \mathbb{F}^{I_n \times P_n}$ for each mode n as follows [KB09]

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{G}} \times_1 \boldsymbol{V}_1 \times_2 \boldsymbol{V}_2 \times_3 \ldots \times_N \boldsymbol{V}_N, \tag{2.26}$$

where the P_ns denote the number of components (columns). If $P_n < I_n$, the core tensor can be viewed as a compressed version of $\boldsymbol{\mathcal{X}}$. Equation (2.26) refers to a Tucker decomposition in its most general case, without any constraints on the core tensor and the factors.

However, the most common representation of the Tucker decomposition, also known as higher-order singular value decomposition (HOSVD) or multilinear singular value decomposition [DDV00a; KB09], takes the form

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{S}} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \times_3 \ldots \times_N \boldsymbol{U}_N, \tag{2.27}$$

with the orthogonal loading matrices $\boldsymbol{U}_n \in \mathbb{F}^{I_n \times I_n}$ representing the basis of the n-mode space of $\boldsymbol{\mathcal{X}}$, and a core tensor $\boldsymbol{\mathcal{S}} \in \mathbb{F}^{I_1 \times \ldots \times I_N}$ with the property of "all-orthogonality" [DDV00a]. The decomposition is illustrated in Figure 2.10. The loading

matrices of the HOSVD can be computed as the left singular vectors from the SVDs of the n-mode unfoldings as

$$[\boldsymbol{\mathcal{X}}]_{(n)} = \boldsymbol{U}_n \boldsymbol{\Sigma}_n \boldsymbol{V}_n^{\mathrm{H}} \in \mathbb{F}^{I_n \times \bar{I}_n}, \quad \bar{I}_n = I_1 \cdot I_2 \dots I_{n-1} \cdot I_{n+1} \dots I_N \tag{2.28}$$

where the matrices $\boldsymbol{\Sigma}_n \in \mathbb{R}^{I_n \times \bar{I}_n}$ and $\boldsymbol{V}_n \in \mathbb{F}^{\bar{I}_n \times \bar{I}_n}$ correspond to the singular values and the right singular vectors of the matrix $[\boldsymbol{\mathcal{X}}]_{(n)}$, respectively.

Taking into account the unitary property of the matrices $\boldsymbol{U}_n, n \in \{1, \dots, N\}$, the core tensor can be expressed as

$$\boldsymbol{\mathcal{S}} = \boldsymbol{\mathcal{X}} \times_1 \boldsymbol{U}_1^{\mathrm{H}} \times_2 \boldsymbol{U}_2^{\mathrm{H}} \times_3 \dots \times_N \boldsymbol{U}_N^{\mathrm{H}}. \tag{2.29}$$

Furthermore, the n-mode unfolding of the core tensor $\boldsymbol{\mathcal{S}}$ is related to the singular values of the n-mode unfolding of $\boldsymbol{\mathcal{X}}$, such that $[\boldsymbol{\mathcal{S}}]_{(n)} \cdot [\boldsymbol{\mathcal{S}}]_{(n)}^{\mathrm{H}} = \boldsymbol{\Sigma}_n \boldsymbol{\Sigma}_n^{\mathrm{T}} \in \mathbb{R}^{I_n \times I_n}$, and the $\boldsymbol{\Sigma}_n$ is defined in (2.28). Similar to the SVD, we can define the truncated HOSVD as follows

$$\boldsymbol{\mathcal{X}} \approx \boldsymbol{\mathcal{S}}^{[\mathrm{s}]} \times_1 \boldsymbol{U}_1^{[\mathrm{s}]} \times_2 \boldsymbol{U}_2^{[\mathrm{s}]} \times_3 \dots \times_N \boldsymbol{U}_N^{[\mathrm{s}]}, \tag{2.30}$$

where the factor matrices $\boldsymbol{U}_n^{[\mathrm{s}]} \in \mathbb{F}^{I_n \times r_n}$ hold the first $r_n \leq \operatorname{rank}\left([\boldsymbol{\mathcal{X}}]_{(n)}\right)$ columns of the matrices $\boldsymbol{U}_n$ in equation (2.28).

In case of the three-dimensional tensor $\boldsymbol{\mathcal{X}} \in \mathbb{F}^{I \times J \times K}$, the HOSVD can be expressed as [KB09]

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{S}} \times_1 \boldsymbol{A} \times_2 \boldsymbol{B} \times_3 \boldsymbol{C} = \sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{k=1}^{K} s_{i,j,k} \boldsymbol{a}_i \circ \boldsymbol{b}_j \circ \boldsymbol{c}_k = [\![\boldsymbol{\mathcal{S}}; \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]\!], \tag{2.31}$$

where $\boldsymbol{A} \in \mathbb{F}^{I \times I}$, $\boldsymbol{B} \in \mathbb{F}^{J \times J}$, and $\boldsymbol{C} \in \mathbb{F}^{K \times K}$ are the orthogonal factor matrices and $\boldsymbol{\mathcal{S}} \in \mathbb{F}^{P \times Q \times R}$ is a core tensor with the property of "all-orthogonality" [DDV00a].

The most straightforward way of computing the HOSVD, as shown in (2.28), is via SVD of the tensor unfoldings. Consequently, the core tensor can be computed according to (2.29). The truncated version of the HOSVD can be calculated by taking into account only the first $r_n \leq \operatorname{rank}\left([\boldsymbol{\mathcal{X}}]_{(n)}\right)$ columns of the loading matrices. There are also more solutions in the literature, for example, Higher-order Orthogonal Iteration (HOOI) or sequentially truncated HOSVD algorithms from [DDV00b; VVM12].

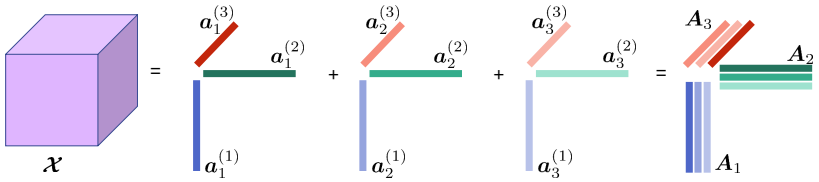


Figure 2.11. Canonical Polyadic Decomposition of a 3-way tensor with rank $R = 3$.

2.2.4 Canonical Polyadic Decomposition

The canonical polyadic decomposition (CPD), also referred to as CANDECOMP (canonical decomposition) and PARAFAC (parallel factors), is a fundamental tensor decomposition technique that expresses a multi-dimensional tensor as the sum of the minimum number of rank-one tensors [Kie00; KB09].

Given a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{F}^{I_1 \times I_2 \times \ldots \times I_N}$, CPD represents it as the sum of R rank-one tensors (assuming the noiseless case):

$$\boldsymbol{\mathcal{X}} = \sum_{r=1}^{R} \boldsymbol{a}_r^{(1)} \circ \boldsymbol{a}_r^{(2)} \circ \ldots \circ \boldsymbol{a}_r^{(N)} = \boldsymbol{\mathcal{I}}_{N,R} \times_1 \boldsymbol{A}_1 \times_2 \boldsymbol{A}_2 \times_3 \ldots \times_N \boldsymbol{A}_N, \tag{2.32}$$

where the vectors $\boldsymbol{a}_r^{(n)} \in \mathbb{F}^{I_n}$ represent the rth columns of the CPD factor matrices $\boldsymbol{A}_n \in \mathbb{F}^{I_n \times R}$. In contrast to the HOSVD, the factor matrices of the canonical polyadic (CP) decomposition do not have to be unitary, which makes it attractive for many applications.

Considering the third-order tensor $\boldsymbol{\mathcal{X}} \in \mathbb{F}^{I \times J \times K}$, the CP decomposition can be expressed as

$$\boldsymbol{\mathcal{X}} = \sum_{r=1}^{R} \boldsymbol{a}_r^{(1)} \circ \boldsymbol{a}_r^{(2)} \circ \boldsymbol{a}_r^{(3)} = \boldsymbol{\mathcal{I}}_{3,R} \times_1 \boldsymbol{A}_1 \times_2 \boldsymbol{A}_2 \times_3 \boldsymbol{A}_3, \tag{2.33}$$

where the vectors $\boldsymbol{a}_r^{(1)}$, $\boldsymbol{a}_r^{(2)}$, and $\boldsymbol{a}_r^{(3)}$ represent the rth columns of the matrices $\boldsymbol{A}_1 \in \mathbb{F}^{I \times R}$, $\boldsymbol{A}_2 \in \mathbb{F}^{J \times R}$, and $\boldsymbol{A}_3 \in \mathbb{F}^{K \times R}$, respectively. The value R denoted the tensor rank (also called CPD rank), i.e., the minimum number of rank-one components that sum up to $\boldsymbol{\mathcal{X}}$. The tensor rank does not have to be equal to the n-ranks. Moreover, there is no straightforward way to determine the tensor rank of a given tensor. More details on the tensor rank can be found in [KB09].

Using the factor matrices in (2.33), the n-mode unfolding of $\boldsymbol{\mathcal{X}}$ can be expressed as follows (in reverse cyclical order)

$$\begin{aligned}
[\boldsymbol{\mathcal{X}}]_{(1)} &= \boldsymbol{A}_1(\boldsymbol{A}_2 \diamond \boldsymbol{A}_3)^{\mathrm{T}}, \\
[\boldsymbol{\mathcal{X}}]_{(2)} &= \boldsymbol{A}_2(\boldsymbol{A}_3 \diamond \boldsymbol{A}_1)^{\mathrm{T}}, \\
[\boldsymbol{\mathcal{X}}]_{(3)} &= \boldsymbol{A}_3(\boldsymbol{A}_1 \diamond \boldsymbol{A}_2)^{\mathrm{T}}.
\end{aligned} \tag{2.34}$$

Another common notation of the CPD is

$$\boldsymbol{\mathcal{X}} = \sum_{r=1}^{R} \lambda_r \boldsymbol{a}_r^{(1)} \circ \boldsymbol{a}_r^{(2)} \circ \boldsymbol{a}_r^{(3)}, \tag{2.35}$$

in which the factor matrices are normalized to length one, and the scaling factors are absorbed in a vector $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_R]$. Additionally, the CPD of $\boldsymbol{\mathcal{X}}$ can also be expressed in terms of its frontal slices as

$$\boldsymbol{X}_k = \boldsymbol{A}_1 \boldsymbol{D}^{(k)} \boldsymbol{A}_2^{\mathrm{T}}, \tag{2.36}$$

where $\boldsymbol{D}^{(k)} \in \mathbb{F}^{R \times R} = \operatorname{diag}\big((\boldsymbol{A}_3)_{(k,:)}\big)$, $k \in \{1, \ldots, K\}$.

As the most straightforward way to compute the CP decomposition, we can consider the alternating least squares (ALS) scheme from [CC70; Har70; KB09]. It is an iterative algorithm that computes one factor matrix at a time, assuming that the other factor matrices are already known. The factor matrices are approximated in the least squares sense from the tensor unfoldings in (2.34). It is important to note that the ALS algorithm does not guarantee convergence, and its required number of iterations can be quite high. Some improved versions of the ALS algorithm include the incorporation of dimensionality reduction techniques, better initialization, and imposing the constraints [TB06].

Other more advanced methods for computing the (approximate) CP decomposition from noise-corrupted data include, for example, the line search [RCH08], pencil-based [DD14], simultaneous diagonalization [RH13], gradient-based [SDF+17], and other optimization algorithms [TB06]. Note that most of the CP decomposition algorithms assume prior knowledge of the tensor rank. However, estimating the tensor rank from noisy observations is a challenging problem. Some researchers propose estimating the tensor rank by fitting CP decompositions for varying ranks and selecting the rank corresponding to the smallest residual. Nonetheless, this approach does not consistently guarantee reliable outcomes. Addressing this issue, the CORCONDIA (core consistency diagnostic) algorithm, proposed in [BK03], employs a core consistency check to estimate the tensor rank. There are some subsequent algorithms that try

to mitigate the complexity of the algorithm in [BK03] and also algorithms that jointly estimate both the tensor rank and the factors of the CP decomposition [HAK+17a].

Uniqueness

A notable characteristic of higher-order tensors is their tendency for unique decompositions, unlike matrix decompositions that lack this property, for example, due to the matrix product ambiguity.

Matrix product ambiguity. Consider the matrix factorization $\boldsymbol{X} = \boldsymbol{A}\boldsymbol{B}^{\mathrm{T}}$. Replacing the matrix $\boldsymbol{A}$ by $\boldsymbol{A}\boldsymbol{H}$ and $\boldsymbol{B}$ by $\boldsymbol{H}^{-1}\boldsymbol{B}^{\mathrm{T}}$, where $\boldsymbol{H}$ is an arbitrary non-singular matrix, still results in a valid decomposition of $\boldsymbol{X}$:

$$\boldsymbol{X} = \boldsymbol{A}\boldsymbol{B}^{\mathrm{T}} = \big(\boldsymbol{A}\boldsymbol{H}\big)\big(\boldsymbol{H}^{-1}\boldsymbol{B}^{\mathrm{T}}\big) = \boldsymbol{A}'\boldsymbol{B}'^{\mathrm{T}}. \tag{2.37}$$

The ways to overcome this kind of ambiguity include, for instance, introducing additional constraints such as orthogonality, triangular form, sparsity, etc.

In general, the CP decomposition of a tensor $\boldsymbol{\mathcal{X}}$ is called essentially unique if there is only one possible combination of rank-one tensors that sums to $\boldsymbol{\mathcal{X}}$, except for the elementary indeterminacies of scaling and permutation [KB09; DeL06].

Permutation ambiguity. The permutation indeterminacy means that the rank-one component tensors (columns in the factor matrices) can be arbitrarily permuted by a permutation matrix[1] $\boldsymbol{P} \in \mathbb{R}^{R\times R}$

$$\begin{aligned}\boldsymbol{\mathcal{X}} &= \boldsymbol{\mathcal{I}}_{3,R} \times_1 \boldsymbol{A_1} \times_2 \boldsymbol{A}_2 \times_3 \boldsymbol{A}_3 \\ &= \boldsymbol{\mathcal{I}}_{3,R} \times_1 \boldsymbol{A}_1\boldsymbol{P} \times_2 \boldsymbol{A}_2\boldsymbol{P} \times_3 \boldsymbol{A}_3\boldsymbol{P} \\ &= \boldsymbol{\mathcal{I}}_{3,R} \times_1 \boldsymbol{A}'_1 \times_2 \boldsymbol{A}'_2 \times_3 \boldsymbol{A}'_3,\end{aligned} \tag{2.38}$$

where $\boldsymbol{A}'_1, \boldsymbol{A}'_2$, and $\boldsymbol{A}'_3$ are equally valid factors of the CP decomposition.

[1] A permutation matrix $\boldsymbol{P}$ is a square matrix such that each row and column contains one element equal to 1, and the remaining elements are equal to 0 (identity matrix with re-ordered rows) [GVL96].

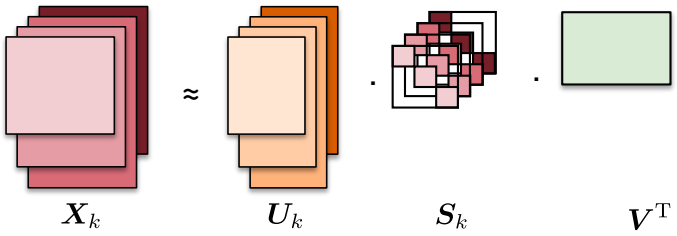


Figure 2.12. PARAFAC2 decomposition.

Scaling ambiguity. The scaling ambiguity means that the individual columns of the rank-1 terms can be arbitrarily scaled, i.e.,

$$\boldsymbol{\mathcal{X}} = \sum_{r=1}^{R} \left(\alpha_r \boldsymbol{a}_r^{(1)}\right) \circ \left(\beta_r \boldsymbol{a}_r^{(2)}\right) \circ \left(\gamma_r \boldsymbol{a}_r^{(3)}\right), \tag{2.39}$$

as long as their product stays the same: $\alpha_r \beta_r \gamma_r = 1$, $r \in \{1, \ldots, R\}$ [KB09].

Kruskal Rank. The k-rank of a matrix $\boldsymbol{A}$, denoted $k_{\boldsymbol{A}}$, is defined as the maximum value k such that any k columns of $\boldsymbol{A}$ are linearly independent [Kru77].

The most well-known uniqueness condition for the CP decomposition is *Kruskal's condition*, which states that a sufficient condition for the uniqueness of the third-order CPD is

$$k_{\boldsymbol{A}_1} + k_{\boldsymbol{A}_2} + k_{\boldsymbol{A}_3} \geq 2R + 2, \tag{2.40}$$

where $k_{\boldsymbol{A}_1}$, $k_{\boldsymbol{A}_2}$, and $k_{\boldsymbol{A}_3}$ denote the k-rank (Kruskal Rank) of the matrices $\boldsymbol{A}_1$, $\boldsymbol{A}_2$, and $\boldsymbol{A}_3$, respectively.

For the N-way tensors, the sufficient condition for uniqueness is [SB00]

$$\sum_{n=1}^{N} k_{\boldsymbol{A}^{(n)}} \geq 2R + (N-1). \tag{2.41}$$

There are more works on the CPD uniqueness conditions in the literature, such as [DeL06; LS01].

Generally, the CP decomposition is one of the most commonly used decompositions. It found applications in a broad range of fields, including signal processing, chemometrics, data analysis, neuroscience, communications, and many more [CMD+15].

2.2.5 PARAFAC2

The PARAFAC2 (parallel factors 2) decomposition is an extension of a three-way CP decomposition that factorizes a collection of matrices $\boldsymbol{X}_k \in \mathbb{F}^{I_k \times J}$, $k \in \{1, \ldots, K\}$ with the same number of columns but (possibly) a different number of rows (I_k is allowed to vary with k). Therefore, the PARAFAC2 decomposition can also be seen as a coupled decomposition applied to k matrices or as a tensor decomposition of a tensor $\boldsymbol{\mathcal{X}}$ with matrices $\boldsymbol{X}_k$ being its frontal slices. In a slice-wise fashion, the PARAFAC2 model is expressed as follows [KB09]

$$\boldsymbol{X}_k \approx \boldsymbol{U}_k \boldsymbol{S}_k \boldsymbol{V}^{\mathrm{T}}, \tag{2.42}$$

where $\boldsymbol{U}_k$ is an $I_k \times R$ matrix, $\boldsymbol{S}_k$ is an $R \times R$ diagonal matrix, $k = \{1, \ldots, K\}$, and $\boldsymbol{V}$ is a $J \times R$ factor matrix that does not vary with k. Moreover, R corresponds to the number of components (rank). In general, this decomposition is not unique. The essential uniqueness of the PARAFAC2 decomposition is achieved by imposing the Harshman constraints [Har72], such that $\boldsymbol{U}_k^{\mathrm{H}} \boldsymbol{U}_k = \boldsymbol{H}^{\mathrm{H}} \boldsymbol{H}$, $\boldsymbol{U}_k = \boldsymbol{Q}_k \boldsymbol{H}$, and $\boldsymbol{Q}_k \in \mathbb{R}^{I_k \times R}$ has orthonormal rows.

One of the ways to compute the PARAFAC2 decomposition is proposed in [KBB99] and is called the direct fitting approach, in which the factor matrices are computed in a two-step iterative way, where first the matrix $\boldsymbol{Q}_k$ is calculated utilizing the SVD, and the remaining unknowns $\boldsymbol{H}$, $\boldsymbol{S}_k$, and $\boldsymbol{V}$ are updated via a CPD-ALS procedure [KBB99; KB09].

The applications of the PARAFAC2 include biomedical signal analysis [CHGH18; CNH+18], chromatograhic data analysis [BAK99], wireless communications [KCH19], and others [KB09].

2.2.6 Block-Term Decompositions

The block-term decompositions (BTD) allow the decomposition of a tensor into blocks of smaller sizes [DeL08a] and are considered as extensions of the HOSVD and the CP decompositions. There are several types of BTDs in the literature. In this section, we briefly review three of them, namely, the decomposition in rank-$(L_r, L_r, 1)$ terms, the decomposition in rank-(L_r, M_r, N_r) terms, and the type-2 decomposition in rank-$(L_r, M_r, \cdot)$ terms.

Rank-$(L_r, L_r, 1)$ decomposition

The rank-$(L_r, L_r, 1)$ BTD of a three-dimensional tensor $\mathcal{T} \in \mathbb{F}^{I \times J \times K}$ can be expressed as (in the noiseless case) [VDS+16]

$$\mathcal{T} = \sum_{r=1}^{R} \left(\boldsymbol{A}_r \cdot \boldsymbol{B}_r^{\mathrm{T}} \right) \circ \boldsymbol{c}_r, \tag{2.43}$$

where R and L_r are the number of block-terms and the multilinear ranks, respectively. Moreover, $\boldsymbol{A}_r \in \mathbb{F}^{I \times L_r}$ and $\boldsymbol{B}_r \in \mathbb{F}^{J \times L_r}$ are the r-th submatrices of $\boldsymbol{A}$ and $\boldsymbol{B}$ so that $\boldsymbol{A} = [\boldsymbol{A}_1, \boldsymbol{A}_2, ..., \boldsymbol{A}_R] \in \mathbb{F}^{I \times \Sigma L_r}$ and $\boldsymbol{B} = [\boldsymbol{B}_1, \boldsymbol{B}_2, ..., \boldsymbol{B}_R] \in \mathbb{F}^{J \times \Sigma L_r}$. The vectors $\boldsymbol{c}_r \in \mathbb{F}^{K \times 1}$, $r = \{1, ..., R\}$ are stacked in the matrix $\boldsymbol{C}$ such that $\boldsymbol{C} = [\boldsymbol{c}_1, \boldsymbol{c}_2, ..., \boldsymbol{c}_R] \in \mathbb{F}^{K \times R}$. For an illustration of the rank-$(L_r, L_r, 1)$ decompostion, we refer the reader to Figure 7.2 on page 100.

The rank-$(L_r, L_r, 1)$ BTD can be seen as a special case of the CP decomposition with collinearity in one of the factors (in the 3-mode, in case of the model in (2.43)) (we will give more details and exploit this connection in Chapter 7). Similar to the CPD, the rank-$(L_r, L_r, 1)$ decomposition can be written in terms of unfoldings and Khatri-Rao products as follows [DeL08a]:

$$\begin{aligned}
\left[\mathcal{T}\right]_{(1)} &= \boldsymbol{A} \cdot (\boldsymbol{B} \diamond_{\mathrm{s}} \boldsymbol{C})^{\mathrm{T}}, \\
\left[\mathcal{T}\right]_{(2)} &= \boldsymbol{B} \cdot (\boldsymbol{C} \diamond_{\mathrm{s}} \boldsymbol{A})^{\mathrm{T}}, \\
\left[\mathcal{T}\right]_{(3)} &= \boldsymbol{C} \cdot [(\boldsymbol{A}_1 \diamond \boldsymbol{B}_1) \cdot \mathbf{1}_{L_1}, \cdots, (\boldsymbol{A}_R \diamond \boldsymbol{B}_R) \cdot \mathbf{1}_{L_R}]^{\mathrm{T}},
\end{aligned} \tag{2.44}$$

where $\diamond_{\mathrm{s}}$ denotes the partition-wise Khatri-Rao product (see Appendix A.1 for the definition). Due to the matrix product ambiguities in the product $\left(\boldsymbol{A}_r \cdot \boldsymbol{B}_r^{\mathrm{T}} \right)$ (see the definitions of the ambiguities on page 27), the rank-$(L_r, L_r, 1)$ decomposition can also be expressed as follows

$$\mathcal{T} = \sum_{r=1}^{R} \boldsymbol{E}_r \circ \boldsymbol{c}_r, \tag{2.45}$$

where $\boldsymbol{A}_r$ and $\boldsymbol{B}_r$ are merged into $\boldsymbol{E}_r = \boldsymbol{A}_r \cdot \boldsymbol{B}_r^{\mathrm{T}}$. Since the ambiguity in $\left(\boldsymbol{A}_r \cdot \boldsymbol{B}_r^{\mathrm{T}} \right)$ is difficult to resolve, the representation in (2.45) is often used when considering the applications of the rank-$(L_r, L_r, 1)$ BTD. For the conditions under which the rank-$(L_r, L_r, 1)$ decomposition is unique, we refer the reader to Appendix A.4.

The rank-$(L_r, L_r, 1)$ decomposition is one of the main focuses of this thesis. We propose an algorithm to compute the decomposition, show some of the applications of the rank-$(L_r, L_r, 1)$ BTD, and discuss the state-of-the-art schemes and other details in Part III of the thesis.

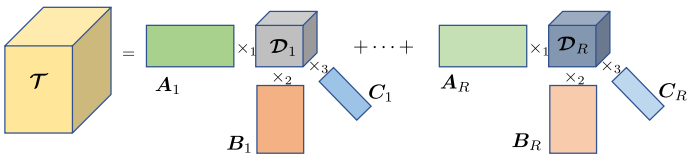


Figure 2.13. Decomposition in rank-(L_r, M_r, N_r) terms. In this example, $L_1 = \ldots = L_R$, $M_1 = \ldots = M_R$, and $N_1 = \ldots = N_R$.

Decomposition in rank-(L_r, M_r, N_r) terms

The decomposition of a (noisless) tensor $\mathcal{T} \in \mathbb{F}^{I \times J \times K}$ into a sum of rank-(L_r, M_r, N_r) terms, $r \in \{1, \ldots, R\}$, is expressed as

$$\mathcal{T} = \sum_{r=1}^{R} \mathcal{D}_r \times_1 \boldsymbol{A}_r \times_2 \boldsymbol{B}_r \times_3 \boldsymbol{C}_r \tag{2.46}$$

where the tensors $\mathcal{D}_r \in \mathbb{F}^{L_r \times M_r \times N_r}$ have full multidimensional rank (L_r, M_r, N_r) and $\boldsymbol{A}_r \in \mathbb{F}^{I \times L_r}$, $(I \geq L_r)$, $\boldsymbol{B}_r \in \mathbb{F}^{J \times M_r}$, $(J \geq M_r)$, and $\boldsymbol{C}_r \in \mathbb{F}^{K \times N_r}$ $(K \geq N_r)$ are of full column rank, $r \in \{1, \ldots, R\}$. The rank-(L_r, M_r, N_r) decomposition can be seen as a generalization of the CP and Tucker decompositions.

Using the partitioned matrices $\boldsymbol{A} = [\boldsymbol{A}_1, \ldots, \boldsymbol{A}_R]$, $\boldsymbol{B} = [\boldsymbol{B}_1, \ldots, \boldsymbol{B}_R]$, and $\boldsymbol{C} = [\boldsymbol{C}_1, \ldots, \boldsymbol{C}_R]$, the tensor $\mathcal{T}$ can be expressed as

$$\begin{aligned}
[\mathcal{T}]_{(1)} &= (\boldsymbol{B} \diamond_{\mathrm{s}} \boldsymbol{C}) \cdot \mathrm{bdiag}\Big([\mathcal{D}_1]_{(1)}, \ldots, [\mathcal{D}_R]_{(1)}\Big) \cdot \boldsymbol{A}^{\mathrm{T}}, \\
[\mathcal{T}]_{(2)} &= (\boldsymbol{C} \diamond_{\mathrm{s}} \boldsymbol{A}) \cdot \mathrm{bdiag}\Big([\mathcal{D}_1]_{(2)}, \ldots, [\mathcal{D}_R]_{(2)}\Big) \cdot \boldsymbol{B}^{\mathrm{T}}, \\
[\mathcal{T}]_{(3)} &= (\boldsymbol{A} \diamond_{\mathrm{s}} \boldsymbol{B}) \cdot \mathrm{bdiag}\Big([\mathcal{D}_1]_{(3)}, \ldots, [\mathcal{D}_R]_{(3)}\Big) \cdot \boldsymbol{C}^{\mathrm{T}},
\end{aligned} \tag{2.47}$$

where $\diamond_{\mathrm{s}}$ denotes the partition-wise Khatri-Rao product (see Appendix A.1 for the definition). The ALS algorithm to compute the rank-(L_r, M_r, N_r) BTD using the tensor unfoldings in (2.47) is presented in [DN08]. For more details on the rank-(L_r, M_r, N_r) BTD, its computation, and uniqueness, we refer the reader to [DeL08b; DeL08a; DN08]. The decomposition is visualized in Figure 2.13.

Decomposition in rank-$(L_r, M_r, \cdot)$ terms

A type-2 decomposition of a tensor $\mathcal{T} \in \mathbb{F}^{I \times J \times K}$ into a sum of rank-$(L_r, M_r, \cdot)$ terms, $r \in \{1, \ldots, R\}$, is expressed as [DeL08a]

$$\mathcal{T} = \sum_{r=1}^{R} \mathcal{C}_r \times_1 \boldsymbol{A}_r \times_2 \boldsymbol{B}_r, \tag{2.48}$$

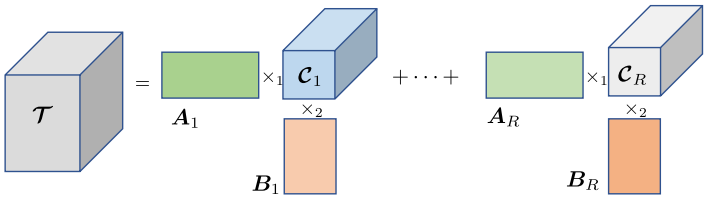


Figure 2.14. Decomposition in rank-$(L_r, M_r, \cdot)$ terms. In this example, $L_1 = \ldots = L_R$ and $M_1 = \ldots = M_R$.

where $\mathcal{C}_r \in \mathbb{F}^{L_r \times M_r \times K}$ has 1-rank and 2-rank equal to L_r and M_r, respectively. The matrices, $\boldsymbol{A}_r \in \mathbb{F}^{I \times L_r}$, $(I \geq L_r)$ and $\boldsymbol{B}_r \in \mathbb{F}^{J \times M_r}$, $(J \geq M_r)$ are of full column rank. The label "type-2" refers to the Tucker2 decomposition (the Tucker2 decomposition of a 3-way tensor assumes that one of the factor matrices is an identity matrix) [Tuc66; DeL08a; KL80]. The decomposition is illustrated in Figure 2.14.

Let $\boldsymbol{A} = [\boldsymbol{A}_1, \ldots, \boldsymbol{A}_R]$ and $\boldsymbol{B} = [\boldsymbol{B}_1, \ldots, \boldsymbol{B}_R]$. Then, the rank-$(L_r, M_r, \cdot)$ decomposition can be computed based on ALS using the unfoldings of $\mathcal{T}$

$$\begin{aligned}
[\mathcal{T}]_{(1)} &= \left[\; [\mathcal{C}_1 \times_2 \boldsymbol{B}_1]^{\mathrm{T}}_{(1)}, \quad \ldots, \quad [\mathcal{C}_R \times_2 \boldsymbol{B}_R]^{\mathrm{T}}_{(1)} \; \right] \cdot \boldsymbol{A}^{\mathrm{T}}, \\
[\mathcal{T}]_{(2)} &= \left[\; [\mathcal{C}_1 \times_1 \boldsymbol{A}_1]^{\mathrm{T}}_{(2)}, \quad \ldots, \quad [\mathcal{C}_R \times_1 \boldsymbol{A}_R]^{\mathrm{T}}_{(2)} \; \right] \cdot \boldsymbol{B}^{\mathrm{T}}, \\
[\mathcal{T}]_{(3)} &= (\boldsymbol{A} \diamond_{\mathrm{s}} \boldsymbol{B}) \cdot \begin{pmatrix} [\mathcal{C}_1]^{\mathrm{T}}_{(3)} \\ \vdots \\ [\mathcal{C}_R]^{\mathrm{T}}_{(3)} \end{pmatrix},
\end{aligned} \tag{2.49}$$

where $\diamond_{\mathrm{s}}$ denotes the partition-wise Khatri-Rao product (see Appendix A.1 for the definition).

More details on the decomposition, its computation, and its uniqueness can be found in [DeL08b; DeL08a; DN08].

2.2.7 Coupled decompositions

Coupled tensor decompositions represent the tensor factorization methods that allow extracting shared patterns and interdependencies from multiple tensors. Processing the data from multiple sources jointly often provides deeper insights and is more accurate. Moreover, data fusion can provide factor uniqueness in cases where the single tensor or matrix factorization is not unique. In general, the scenarios where several data sets exhibit shared underlying structures are not unusual, and coupled

tensor decompositions emerge as powerful tools to capture the shared and individual information [SVBD15].

Therefore, coupled decompositions and data fusion are increasingly attractive topics in signal processing and data analysis [SD15].

In the scope of this thesis, in Part III in particular, we will exploit the coupled rank-$(L_r, L_r, 1)$ decomposition and its application to radar systems and joint electroencephalography (EEG) and magnetoencephalography (MEG) signal processing.

In general, all tensor decompositions we have discussed in previous sections have their coupled extensions, for example, the coupled CP decomposition of L third-order tensors $\boldsymbol{\mathcal{X}}_\ell \in \mathbb{F}^{I_1 \times I_2^{(\ell)} \times I_3^{(\ell)}}$, $\ell \in \{1, \ldots, L\}$ can be expressed as (in the noiseless case)

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{I}}_{3,R} \times_1 \boldsymbol{A}_1 \times_2 \boldsymbol{A}_2^{(\ell)} \times_3 \boldsymbol{A}_3^{(\ell)}, \tag{2.50}$$

where $\boldsymbol{A}_1$ is a common factor matrix, and $\boldsymbol{A}_2^{(\ell)}$ and $\boldsymbol{A}_3^{(\ell)}$ are the individual factor matrices.

For more details on data fusion and different coupled decompositions and applications, we refer the reader to [SVBD15; SD15; SDD15; MGK+21; CNH+18].

Additionally, we collected some interesting overview papers and textbooks on tensor decompositions in Appendix A.8.

2.3 Conclusions

In this chapter, we introduced the notation and reviewed the fundamental concepts of tensor algebra used in the following chapters. The presented material included essential definitions and concepts of multilinear algebra, such as n-mode products, ranks, unfoldings, and vectorization. Moreover, in this chapter, we gave a brief overview of the basic tensor and matrix decompositions that will be exploited in the subsequent chapters. We have covered the generalized singular value decomposition of two matrices, which we extend to the tensor case in Chapter 3. Additionally, we reviewed the PARAFAC2 decomposition, which, as the readers will see, is also utilized in Chapter 3. Moreover, we reviewed the HOSVD and the CP decomposition, the most well-known tensor decompositions, and discussed some of the characteristics of the tensor factorizations, such as uniqueness, ambiguities, and matrix representations. Furthermore, we reviewed some of their computational algorithms and applications.

In addition, this chapter provided a brief introduction to the block-term decompositions and their different representations. The BTDs in this chapter can be considered as multirank extensions of the Tucker and CP decompositions. In Chapter 7 of this thesis, we will focus on the rank-$(L_r, L_r, 1)$ block-term decomposition, introduce an algorithm to compute it, and present its applications to wireless systems and biomedical signal processing in the subsequent chapters.

Part II.

Multilinear Generalized Singular Value Decomposition and Applications

Multilinear Generalized Singular Value Decomposition

3

This chapter introduces the multilinear generalized singular value decomposition (ML-GSVD) of 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 the orthogonality of the second-mode factor matrices as well as the subspace structure of the third-mode factor matrices. We introduce an alternating least squares (ALS)-based algorithm to compute the ML-GSVD, which has been inspired by PARAFAC2 (parallel factors 2) decomposition algorithms [KAH22]. Furthermore, we briefly present some of the applications of the proposed ML-GSVD that we will discuss in more detail in the subsequent chapters of the thesis.

3.1 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.

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 [KAH20] 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 [FS12; MCF16; HRML21], multiple-input

multiple-output (MIMO) relaying [ST10; ST13], physical layer security [KWWE07; WWW+17], and genomic signal processing [AA16; ABB03; LASA12]. In this chapter of the thesis, we present an extension of the GSVD [PS81], [Van76] 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. The initial investigations on the ML-GSVD and its calculation we have first presented in [KAH20]. Then in [KAH22], we introduced a more general definition of the ML-GSVD, which is applicable to a set of both full-rank and rank-deficient matrices. Moreover, we specified the corresponding optimization problem. Additionally, we provided an enhanced algorithm to calculate the ML-GSVD and its application to multi-user MIMO orthogonal frequency division multiplexing (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 in the literature. Two different multidimensional decompositions that extend the GSVD to the tensor case have been introduced in [PSVLA11] and [SSAA15]. The authors in [PSVLA11] 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 GSVD. The HO GSVD in [PSVLA11] is used for a comparative analysis of a global messenger ribonucleic acid (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 [KGD21] 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 [SSAA15], 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 the deoxyribonucleic acid (DNA) analysis. Both [PSVLA11] and [SSAA15] 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 appli-

cations, 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. In this thesis, we will focus on the following applications of the ML-GSVD:

- Downlink MIMO-OFDM systems with unicast and multicast transmissions (Chapter 4);
- Non-orthogonal multiple access systems(Chapter 5);
- Rate splitting multiple access systems (Chapter 6).

Additionally, it is important to acknowledge that while we have extensively explored and discussed certain applications of ML-GSVD, there exist additional intriguing applications that merit attention. For example, as an interesting perspective, we see a physical layer security application [KWWE07], where the ML-GSVD can be used to extend the existing results to more than two users. Furthermore, the similarity to the HO GSVD [PSVLA11] inspires the exploration of the ML-GSVD in the biomedical signal processing context. Yet, before coming to the applications, let us give an overview of the ML-GSVD.

Main contributions. In this chapter we:

- Define a new extension of the GSVD [PS81], [Van76] 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 two or more 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.
- Introduce a general ALS-based algorithm to compute the proposed ML-GSVD decomposition. In particular, we show that any algorithm to calculate the PARAFAC2 (parallel factors 2) decomposition can be modified to compute the proposed ML-GSVD.

Moreover, 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.

The rest of this chapter is organized as follows. We introduce the ML-GSVD in Section 3.2. Section 3.3 presents an algorithm to compute the ML-GSVD. The numerical results are shown in Section 3.4, and the conclusions are drawn in Section 3.5.

3.2 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

$$\begin{aligned} \boldsymbol{H}_1 &= \boldsymbol{B}_1 \cdot \boldsymbol{C}_1 \cdot \boldsymbol{A}^{\mathrm{H}} \in \mathbb{C}^{J_1 \times I}, \\ &\vdots \\ \boldsymbol{H}_K &= \boldsymbol{B}_K \cdot \boldsymbol{C}_K \cdot \boldsymbol{A}^{\mathrm{H}} \in \mathbb{C}^{J_K \times I}. \end{aligned} \tag{3.1}$$

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 $J = \max\left(J_1, \ldots, J_K\right)$ (zeros are added for those elements that are not defined in (3.1)). 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 $\boldsymbol{\mathcal{H}}$ (Figure 3.1) can be defined in a slice-wise fashion as

$$\boldsymbol{H}_k = \boldsymbol{B}_k \cdot \boldsymbol{C}_k \cdot \boldsymbol{A}^{\mathrm{H}} \in \mathbb{C}^{J_k \times I}, \tag{3.2}$$

where $\boldsymbol{H}_k$, $k \in \{1, \ldots, K\}$, is the kth slice of $\boldsymbol{\mathcal{H}}$. The matrix $\boldsymbol{A} \in \mathbb{C}^{I \times Q}$, $Q = \min\left(\sum_{k=1}^{K} J_k, I\right)$, is left invertible and common for all the K 3-mode slices of the tensor $\boldsymbol{\mathcal{H}}$. The non-zero subcolumns $\hat{\boldsymbol{B}}_k \in \mathbb{C}^{J_k \times r_k}$ of the matrix $\boldsymbol{B}_k \in \mathbb{C}^{J_k \times Q}$, corresponding to the k-th slice of $\boldsymbol{\mathcal{H}}$, have orthogonal columns such that $\hat{\boldsymbol{B}}_k^{\mathrm{H}} \cdot \hat{\boldsymbol{B}}_k = \boldsymbol{I}_{r_k}$, $r_k = \operatorname{rank}\left(\boldsymbol{H}_k\right)$. The matrices $\boldsymbol{C}_k \in \mathbb{R}^{Q \times Q}$ are diagonal with non-negative entries satisfying $\sum_{k=1}^{K} \boldsymbol{C}_k^2 = \boldsymbol{I}_Q$. In contrast to the GSVD in (2.23), in the ML-GSVD, the matrices $\boldsymbol{C}_k$ are square such that the zero matrices in (2.24) and (2.25) are filled with zeros when they have no rows and are empty if they have no columns. 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., $\boldsymbol{C}_k = \operatorname{diag}\left(\boldsymbol{C}_{(k,:)}\right)$. Then, $\boldsymbol{C}$ has

[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 thesis.

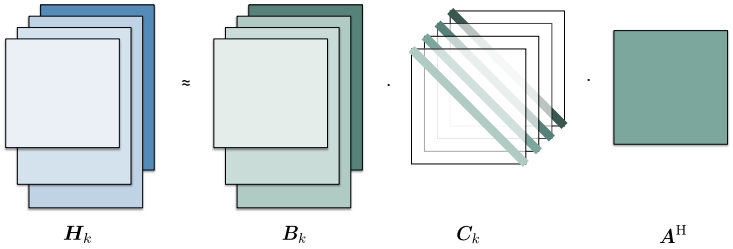


Figure 3.1. Illustration of Multilinear Generalized Singular Value Decomposition.

unit column norms: $\sum_{k=1}^{K} c_{k,i}^2 = 1$ for all $i = 1 \dots Q$. Consequently, the optimization problem to be solved can be stated as follows

$$\begin{aligned} &\underset{\boldsymbol{A},\boldsymbol{B}_k,\boldsymbol{C}_k}{\arg\min} \quad \sum_{k=1}^{K} \left\| \boldsymbol{H}_k - \boldsymbol{B}_k \boldsymbol{C}_k \boldsymbol{A}^{\mathrm{H}} \right\|_{\mathrm{F}}^2 \\ &\text{subject to} \quad \hat{\boldsymbol{B}}_k^{\mathrm{H}} \cdot \hat{\boldsymbol{B}}_k = \boldsymbol{I}_{r_k}, \quad \sum_{k=1}^{K} \boldsymbol{C}_k^2 = \boldsymbol{I}_Q. \end{aligned} \tag{3.3}$$

Note that due to the imposed constraints, the model in (3.2) is an approximation of $\boldsymbol{H}_k$ in the 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 [GVL13], we assume that the null spaces of the $\boldsymbol{H}_k$s do not overlap, i.e., $\mathrm{null}(\boldsymbol{H}_1) \cap \mathrm{null}(\boldsymbol{H}_2) \cap \dots \cap \mathrm{null}(\boldsymbol{H}_K) = \emptyset$. 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

$$\boldsymbol{C}_{(1,:)} = \begin{bmatrix} \mathbf{0}_{p_2+\dots+p_K}^{\mathrm{T}} & \boldsymbol{\sigma}_1^{\mathrm{T}} & \mathbf{1}_{p_1}^{\mathrm{T}} \end{bmatrix} \in \mathbb{R}^{1 \times Q}, \tag{3.4}$$

where $\boldsymbol{C}_{(1,:)} = \mathrm{diag}\left(\boldsymbol{C}_1\right)$, $\mathbf{1}_{p_1}$ is a vector of ones, and $\mathbf{0}_{p_2+\dots+p_K}$ 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 (3.5) and (3.6) 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 columns one, two, and three that coincide with zeros in the second row.

$$\boldsymbol{C}_{\text{before reordering}} = \begin{bmatrix} 1.0000 & 0.7050 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.7092 & 0.0000 & 0.0000 & 1.0000 & 0.0000 \\ 0.0000 & 0.0000 & 1.0000 & 0.9193 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.3935 & 0.0000 & 1.0000 \end{bmatrix}, \tag{3.5}$$

$$\boldsymbol{C}_{\text{after reordering}} = \begin{bmatrix} 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.7050 & 1.0000 \\ 0.0000 & 0.0000 & 0.0000 & 1.0000 & 0.7092 & 0.0000 \\ 0.0000 & 0.9193 & 1.0000 & 0.0000 & 0.0000 & 0.0000 \\ 1.0000 & 0.3935 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \end{bmatrix}. \tag{3.6}$$

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

Case 1

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

$$\boldsymbol{C}_{(k,:)} = \left[\boldsymbol{\sigma}_k^{\mathrm{T}}\right] \in \mathbb{R}^{1 \times I}, \tag{3.7}$$

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

$$\boldsymbol{H}_k = \boldsymbol{B}_k \cdot \underbrace{\begin{bmatrix} \sigma_{k,1} & & \\ & \ddots & \\ & & \sigma_{k,I} \end{bmatrix}}_{\mathrm{diag}(\boldsymbol{C}_{(k,:)})} \cdot \boldsymbol{A}^{\mathrm{H}} \in \mathbb{C}^{J_k \times I}, \tag{3.8}$$

where $\mathrm{diag}\left(\boldsymbol{C}_{(k,:)}\right)$ is a full-rank diagonal matrix. The columns of $\boldsymbol{A} \in \mathbb{C}^{I \times I}$ 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 (3.8) 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 subspaces, while the other non-zero terms correspond to the common subspaces. 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

$$\boldsymbol{H}_k = \begin{bmatrix} \boldsymbol{O}_{J_k \times I - r_k} & \hat{\boldsymbol{B}}_k \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{O}_{I-r_k} & & \\ & \boldsymbol{\Sigma}_{c_k} & \\ & & \boldsymbol{I}_{p_k} \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{A}_{o_k}^{\mathrm{H}} \\ \boldsymbol{A}_{c_k}^{\mathrm{H}} \\ \boldsymbol{A}_{p_k}^{\mathrm{H}} \end{bmatrix}, \tag{3.9}$$

where $\boldsymbol{\Sigma}_{c_k} = \mathrm{diag}\left(\boldsymbol{\sigma}_k\right)$ and $\hat{\boldsymbol{B}}_k \in \mathbb{C}^{J_k \times r_k}$. The matrix $\boldsymbol{A} \in \mathbb{C}^{I \times I}$ 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 subspaces 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 $\boldsymbol{A}_{o_k}$ corresponds to 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 (3.9). 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 $I - r_2$ for $\boldsymbol{H}_1$ and $I - r_1$ for $\boldsymbol{H}_2$, 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

$$\boldsymbol{C} = \begin{bmatrix} \mathbf{0}_{p_K}^{\mathrm{T}} & \cdots & \mathbf{0}_{p_2}^{\mathrm{T}} & \mathbf{1}_{p_1}^{\mathrm{T}} \\ \mathbf{0}_{p_K}^{\mathrm{T}} & \cdots & \mathbf{1}_{p_2}^{\mathrm{T}} & \mathbf{0}_{p_1}^{\mathrm{T}} \\ & \vdots & & \\ \mathbf{1}_{p_K}^{\mathrm{T}} & \cdots & \mathbf{0}_{p_2}^{\mathrm{T}} & \mathbf{0}_{p_1}^{\mathrm{T}} \end{bmatrix} \in \mathbb{R}^{K \times Q}, \tag{3.10}$$

where $Q = \min\left(\sum_{k=1}^{K} J_k, I\right)$. The dimensions of the private subspaces ($\mathbf{1}_{p_k}$) are equal to $r_k = p_k = \operatorname{rank}\left(\boldsymbol{H}_k\right)$. As it can be seen from (3.10), this case provides only private subspaces for each matrix $\boldsymbol{H}_k$, and therefore, the common factor matrix can be rewritten as

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_K, & \ldots, & \boldsymbol{A}_1 \end{bmatrix}. \tag{3.11}$$

Then, we obtain:

$$\begin{aligned} \boldsymbol{H}_1 &= \begin{bmatrix} \boldsymbol{O}_{J_1 \times Q - p_1} & \hat{\boldsymbol{B}}_1 \end{bmatrix} \cdot \underbrace{\begin{bmatrix} \boldsymbol{O}_{Q-p_1} & \\ & \boldsymbol{I}_{p_1} \end{bmatrix}}_{\boldsymbol{C}_k} \cdot \begin{bmatrix} \boldsymbol{A}_K^{\mathrm{H}} \\ \vdots \\ \boldsymbol{A}_1^{\mathrm{H}} \end{bmatrix} \\ &= \hat{\boldsymbol{B}}_1 \boldsymbol{A}_1^{\mathrm{H}} \\ \boldsymbol{H}_2 &= \hat{\boldsymbol{B}}_2 \boldsymbol{A}_2^{\mathrm{H}} \\ &\vdots \\ \boldsymbol{H}_K &= \hat{\boldsymbol{B}}_K \boldsymbol{A}_K^{\mathrm{H}} \end{aligned} \tag{3.12}$$

As it can be observed from (3.12), the matrix $\boldsymbol{C}_k$ can be viewed as a selection matrix, which separates $\boldsymbol{A} \in \mathbb{C}^{I \times Q}$ 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 (3.12) 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., $\operatorname{rank}(\boldsymbol{H}_k) = J_k$, the decomposition has the form as in (3.10) if $\sum_{k=1}^{K} J_k \leq I$.

3.3 Computation of the ML-GSVD inspired by PARAFAC2 algorithms

We can see from equation (3.2), that the ML-GSVD has a similar representation (see Figure 2.12 on page 28) as the PARAFAC2 decomposition [Har72]. It is known from [Har72] 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 non-zero columns of matrices $\boldsymbol{B}_k$ are orthogonal, we set $\boldsymbol{F}$ to the identity matrix, which implies $\hat{\boldsymbol{B}}_k^{\mathrm{H}} \cdot \hat{\boldsymbol{B}}_k = \boldsymbol{I}_{r_k}$. 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

$$\begin{aligned} \underset{\boldsymbol{A},\boldsymbol{B}_k,\boldsymbol{C}_k}{\arg\min} \quad & \sum_{k=1}^{K} \left\| \boldsymbol{H}_k - \boldsymbol{B}_k\boldsymbol{C}_k\boldsymbol{A}^{\mathrm{H}} \right\|_{\mathrm{F}}^2 \\ \text{subject to} \quad & \hat{\boldsymbol{B}}_k^{\mathrm{H}} \cdot \hat{\boldsymbol{B}}_k = \boldsymbol{I}_{r_k}, \quad \sum_{k=1}^{K} \boldsymbol{C}_k^2 = \boldsymbol{I}_Q, \end{aligned} \tag{3.13}$$

where $\boldsymbol{C}_k = \operatorname{diag}(\boldsymbol{C}_{(k,:)})$ 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 [KBB99]. To this end, we alternately minimize (3.13) 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 I left singular vectors of $\sum_{k=1}^{K} \boldsymbol{H}_k^{\mathrm{H}}\boldsymbol{H}_k$ (singular value decomposition (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

$$\sum_{k=1}^{K} \left\| \boldsymbol{B}_k^{\mathrm{H}} \boldsymbol{H}_k - \boldsymbol{C}_k \boldsymbol{A}^{\mathrm{H}} \right\|_{\mathrm{F}}^2 \tag{3.14}$$

subject to $\boldsymbol{B}_k^{\mathrm{H}} \boldsymbol{B}_k = \boldsymbol{I}_Q$, which corresponds to the Orthogonal Procrustes Problem [Sch66] with the solution

$$\boldsymbol{B}_k = (\boldsymbol{T}_k \boldsymbol{T}_k^{\mathrm{H}})^{-\frac{1}{2}} \boldsymbol{T}_k, \tag{3.15}$$

where $\boldsymbol{T}_k = \boldsymbol{H}_k \widetilde{\boldsymbol{H}}_k$, and $\widetilde{\boldsymbol{H}}_k = \boldsymbol{H}_k^{\mathrm{H}} \boldsymbol{B}_k \approx \boldsymbol{A}\boldsymbol{C}_k$. For more details on the Orthogonal Procrustes Problem, see Appendix A.2. Next, we update the matrices $\boldsymbol{A}$ and $\boldsymbol{C}_k$ by solving (3.14) 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 canonical polyadic (CP) decomposition of $\widetilde{\boldsymbol{\mathcal{H}}}$ in terms of its frontal slices is written as

$$\widetilde{\boldsymbol{H}}_k = \boldsymbol{A}\boldsymbol{C}_k \boldsymbol{I}_Q. \tag{3.16}$$

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

$$[\widetilde{\boldsymbol{\mathcal{H}}}]_{(1)} = \boldsymbol{A}(\boldsymbol{I}_Q \diamond \boldsymbol{C})^{\mathrm{T}}, \tag{3.17}$$

$$[\widetilde{\boldsymbol{\mathcal{H}}}]_{(3)} = \boldsymbol{C}(\boldsymbol{A} \diamond \boldsymbol{I}_Q)^{\mathrm{T}}, \tag{3.18}$$

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

$$\boldsymbol{A} = [\widetilde{\boldsymbol{\mathcal{H}}}]_{(1)} (\boldsymbol{I}_Q \diamond \boldsymbol{C})^{\mathrm{T}+} \tag{3.19}$$

$$= [\widetilde{\boldsymbol{\mathcal{H}}}]_{(1)} \left(\mathrm{bdiag}\left(\frac{\boldsymbol{C}_{(:,1)}{}^{\mathrm{H}}}{\left\| \boldsymbol{C}_{(:,1)} \right\|^2}, \ldots, \frac{\boldsymbol{C}_{(:,Q)}{}^{\mathrm{H}}}{\left\| \boldsymbol{C}_{(:,Q)} \right\|^2} \right) \right)^{\mathrm{T}}, \tag{3.20}$$

$$\boldsymbol{C} = [\widetilde{\boldsymbol{\mathcal{H}}}]_{(3)} (\boldsymbol{A} \diamond \boldsymbol{I}_Q)^{\mathrm{T}+} \tag{3.21}$$

$$= [\widetilde{\boldsymbol{\mathcal{H}}}]_{(3)} \left(\left[\mathrm{diag}\left(\frac{\boldsymbol{A}_{(1,:)}{}^{\mathrm{H}}}{\left\| \boldsymbol{A}_{(:,1)} \right\|^2} \right), \ldots, \mathrm{diag}\left(\frac{\boldsymbol{A}_{(Q,:)}{}^{\mathrm{H}}}{\left\| \boldsymbol{A}_{(:,Q)} \right\|^2} \right) \right] \right)^{\mathrm{T}}. \tag{3.22}$$

Equations (3.20) and (3.22) allow us to avoid the explicit computation of the pseudo-inverse in (3.19) and (3.21), see Appendix A.3 for details.

In the original PARAFAC2 algorithm, the matrices $\boldsymbol{A}$ and $\boldsymbol{C}$ are computed by calculating the CP decomposition of $\widetilde{\boldsymbol{\mathcal{H}}}$. In contrast to PARAFAC2, by taking into account the ML-

GSVD constraints in (3.13), we can directly update the matrices $\boldsymbol{A}$ and $\boldsymbol{C}$ without computing the CP decomposition.

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 $\boldsymbol{C}$
2: **repeat**
3: **for** $k = 1, 2 \ldots, K$ **do**
4: Reconstruct $\widetilde{\boldsymbol{\mathcal{H}}}$ with $\boldsymbol{A}$ and $\boldsymbol{C}$:

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

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

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

6: Update $\widetilde{\boldsymbol{\mathcal{H}}}$: $\widetilde{\boldsymbol{H}}_k = \boldsymbol{H}_k^{\mathrm{H}} \cdot \boldsymbol{B}_k$
7: **end for**
8: Update $\boldsymbol{A}$ and $\boldsymbol{C}$

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

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

9: Normalize the columns of $\boldsymbol{C}$
10: To ensure that 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 $\boldsymbol{A}$
11: **for** $i = 1, 2 \ldots, I$ **do**
12: solve $\left|\boldsymbol{C}_{(:,i)}\right| = \alpha \boldsymbol{C}_{(:,i)}$, $\boldsymbol{A}_{(:,i)} = \frac{1}{\alpha} \boldsymbol{A}_{(:,i)}$
13: **end for**
14: **for** $k = 1, 2 \ldots, K$ **do**
15: $\hat{\boldsymbol{H}}_k = \boldsymbol{A} \cdot \operatorname{diag}\left(\boldsymbol{C}_{(k,:)}\right) \cdot \boldsymbol{B}_k^{\mathrm{H}} \in \mathbb{C}^{I \times J_k}$
16: **end for**
17: $E_{\mathrm{R}} = \left(\left\| \hat{\boldsymbol{\mathcal{H}}} - \boldsymbol{\mathcal{H}} \right\|_{\mathrm{F}}^2 \right) / \left\| \boldsymbol{\mathcal{H}} \right\|_{\mathrm{F}}^2$
18: **until** $\Delta E_{\mathrm{R}} = \left(E_{\mathrm{R}}^{\mathrm{old}} - E_{\mathrm{R}} \right) / E_{\mathrm{R}}^{\mathrm{old}}$ is smaller than a predefined threshold or the maximum number of iterations is reached. $E_{\mathrm{R}}^{\mathrm{old}}$ is the residual in the previous iteration
19: Order the columns of $\boldsymbol{C}$ as in (3.4), and reorder the columns of $\boldsymbol{A}$ and $\boldsymbol{B}_k$ accordingly.

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 $\operatorname{diag}\left(\boldsymbol{C}_{(k,:)}\right)$ 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 (3.4) and in the description below this equation, while the columns of $\boldsymbol{A}$ and the rows of $\boldsymbol{B}_k$ are reordered accordingly.

3.3.1 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 (3.13). 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 (3.10), for Case 3, the algorithm can be directly initialized with the matrix of ones and zeros as in (3.10) ("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 (3.20) and (3.22) for the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ has complexity $\mathcal{O}(KI^2)$.

3.3.2 Alternative ways to update the factor matrices

In this subsection, we consider three alternative ways to compute the matrices $\boldsymbol{A}$ and $\boldsymbol{C}$.

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 [NCdH18]. In contrast to a slice-wise representation, the generalized tensor contractions allow representing

an N-way array explicitly in tensor form. Following the derivations in [NCdH18], we can express the tensor $\mathcal{H} \in \mathbb{C}^{I \times K \times J}$ (the dimensions are permuted) in the constrained CP format

$$\mathcal{H} = \mathcal{I}_{3,Q} \times_1 \bar{\boldsymbol{A}} \times_2 \bar{\boldsymbol{C}} \times_3 \bar{\boldsymbol{B}}, \tag{3.23}$$

which is equal to a constrained CP decomposition [FA14], where $\bar{\boldsymbol{A}} = \boldsymbol{A}\left(\mathbf{1}_K^{\mathrm{T}} \otimes \boldsymbol{I}_Q\right)$, $\bar{\boldsymbol{C}} = (\boldsymbol{I}_K \otimes \mathbf{1}_Q^{\mathrm{T}}) \diamond \mathrm{vec}\left(\boldsymbol{C}^{\mathrm{T}}\right)^{\mathrm{T}}$, and $\bar{\boldsymbol{B}} = \left[\mathcal{B}\right]_{(1)}$. The kth slice of $\mathcal{B}$ is the matrix $\boldsymbol{B}_k$, $k \in \{1, \ldots, K\}$. Consequently, the matrix $\boldsymbol{A}$ can be estimated as follows

$$\boldsymbol{A} = \left[\mathcal{H}\right]_{(1)} \cdot \left((\mathbf{1}_K^{\mathrm{T}} \otimes \boldsymbol{I}_Q) \cdot (\bar{\boldsymbol{B}} \diamond \bar{\boldsymbol{C}})^{\mathrm{T}}\right)^{+}. \tag{3.24}$$

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

$$\mathrm{vec}\left(\left[\mathcal{H}\right]_{(3)}^{\mathrm{T}}\right) \approx \left(\bar{\boldsymbol{B}} \diamond (\boldsymbol{I}_K \otimes \mathbf{1}_R^{\mathrm{T}}) \diamond \bar{\boldsymbol{A}}\right) \cdot \mathrm{vec}\left(\boldsymbol{C}^{\mathrm{T}}\right). \tag{3.25}$$

For more details on PARAFAC2 via tensor contractions, we refer the reader to [NCdH18].

Least squares Khatri-Rao factorization

The 2-mode unfolding of the tensor $\widetilde{\mathcal{H}}$ in (3.16) can be written as follows

$$[\widetilde{\mathcal{H}}]_{(2)} = (\boldsymbol{C} \diamond \boldsymbol{A})^{\mathrm{T}}, \tag{3.26}$$

which means that we can employ the least squares Khatri-Rao factorization (LSKRF) [RH10] to estimate the matrices $\boldsymbol{A}$ and $\boldsymbol{C}$.

Joint diagonalization (JD)

Let us consider the Gramian matrix $\boldsymbol{H}_k^{\mathrm{H}} \boldsymbol{H}_k$

$$\begin{aligned} \boldsymbol{H}_k^{\mathrm{H}} \boldsymbol{H}_k &= (\boldsymbol{B}_k \boldsymbol{C}_k \boldsymbol{A}^{\mathrm{H}})^{\mathrm{H}} (\boldsymbol{B}_k \boldsymbol{C}_k \boldsymbol{A}^{\mathrm{H}}) && (3.27)\\ &= \boldsymbol{A} \boldsymbol{C}_k \boldsymbol{B}_k^{\mathrm{H}} \boldsymbol{B}_k \boldsymbol{C}_k \boldsymbol{A}^{\mathrm{H}} && (3.28)\\ &= \boldsymbol{A} \boldsymbol{C}_k^2 \boldsymbol{A}^{\mathrm{H}} = \widetilde{\boldsymbol{H}}_k \widetilde{\boldsymbol{H}}_k^{\mathrm{H}}. && (3.29) \end{aligned}$$

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 [Yer02]

$$\underset{\boldsymbol{A},\boldsymbol{\Lambda}_k}{\arg\min} \sum_{k=1}^{K} \left\| \widetilde{\boldsymbol{H}}_k \widetilde{\boldsymbol{H}}_k^{\mathrm{H}} - \boldsymbol{A}\boldsymbol{\Lambda}_k \boldsymbol{A}^{\mathrm{H}} \right\|_{\mathrm{F}}^2, \tag{3.30}$$

where $\boldsymbol{\Lambda}_k = \boldsymbol{C}_k^2$ is a diagonal matrix. The author in [Yer02] proposes the "AC-DC" (alternating columns-diagonal centers) algorithm that iteratively minimizes (3.30) 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 (3.30) is only essentially unique (permutation and scaling ambiguities are present), and the convergence of (3.30) does not guarantee the convergence of the parameters $\boldsymbol{A}$ and $\boldsymbol{\Lambda}_k$. However, we have observed in practice that this joint diagonalization (JD) 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.

3.4 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)

$$\mathrm{SRE} = \frac{\left\| \hat{\boldsymbol{\mathcal{H}}} - \boldsymbol{\mathcal{H}} \right\|_{\mathrm{F}}^2}{\left\| \boldsymbol{\mathcal{H}} \right\|_{\mathrm{F}}^2}, \tag{3.31}$$

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 3.2, 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 ((3.20) and (3.22)), 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

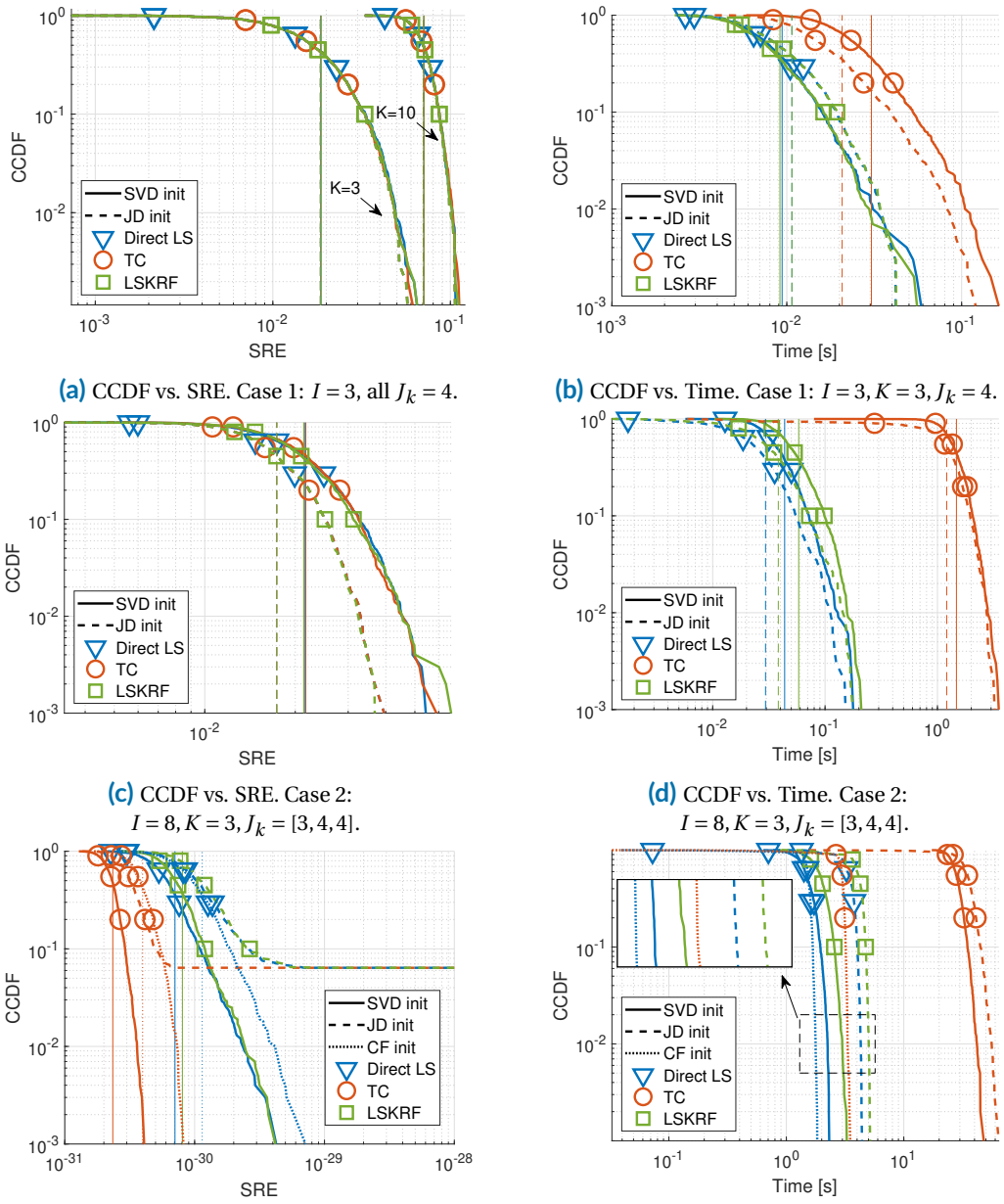
of the algorithms are represented by complementary cumulative distribution functions (CCDFs). In our simulations, the CCDF curves describe the probability that the SRE or the run time will be greater than a certain value on the x-axes. We present the CCDF of the error (time) since it allows an illustration of both the 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 Figures 3.2(a) and 3.2(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. Figure 3.2(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. Figure 3.2(b) shows that the Direct LS and LSKRF solutions have the lowest 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:

$$\boldsymbol{C} = \begin{bmatrix} \sigma_{1,1} & \dots & \sigma_{1,I} \\ \vdots & \ddots & \vdots \\ \sigma_{K,1} & \dots & \sigma_{K,I} \end{bmatrix} \in \mathbb{R}^{K \times I}, \tag{3.32}$$

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



(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: $I = 9, K = 3, J_k = [3,3,3]$.

Figure 3.2. CCDF vs. SRE and time for three cases of the ML-GSVD. Averaged over 1000 realizations of $\mathcal{H}$. 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.

case is of great interest in terms of the applications. Below is an example of the matrix $\boldsymbol{C}$ for the Case 2 with $I = 8$ and the J_ks are equal to 3, 4, and 4, respectively

$$\boldsymbol{C} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \sigma_{1,6} & \sigma_{1,7} & 1 \\ 0 & 0 & \sigma_{2,3} & 1 & 1 & \sigma_{2,6} & 0 & 0 \\ 1 & 1 & \sigma_{3,3} & 0 & 0 & 0 & \sigma_{3,7} & 0 \end{bmatrix} \in \mathbb{R}^{K \times I}. \tag{3.33}$$

As it can be observed from the third, sixth, and the seventh columns, the common subspace is shared by 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 Figures 3.2(c) and 3.2(d). In contrast to the previous results, for Case 2, the JD initialization leads to a smaller reconstruction error and converges faster.

Figures 3.2(e) and 3.2(f) show the CCDFs of the SRE and the time for Case 3, where $I = 9$, $K = 3$, and all $J_k = 3$. The resulting matrix $\boldsymbol{C}$ has the following structure

$$\boldsymbol{C} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \in \mathbb{R}^{K \times I}. \tag{3.34}$$

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 Figure 3.2(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 Figure 3.2(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 Figure 3.2(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.

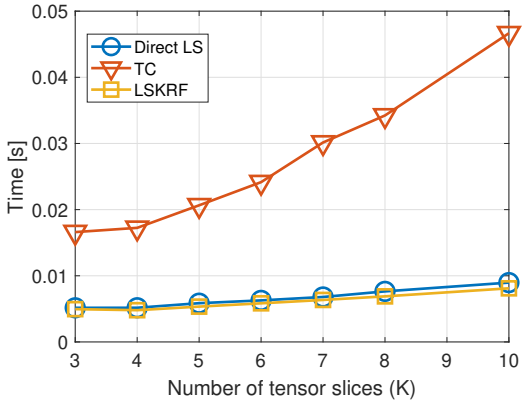


Figure 3.3. Simulation time. The third dimension (K) is changing, $I = 3$, $J_k = 3$.

Table 3.2 shows the algorithms and initializations that have the best reconstruction error and the lowest 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 Figure 3.3, the computational time increases with increasing K for the tensor contractions based estimation, but does not change significantly in case of the direct LS or LSKRF-based solutions.

	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.

Table 3.2. Algorithms with the lowest reconstruction error and the lowest time complexity.

3.5 Conclusions

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 ar-

bitrary 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 the 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. 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.

Joint Unicast and Multicast Transmission using the ML-GSVD

4

As one of the promising applications of the ML-GSVD, we consider a multi-user downlink MIMO orthogonal frequency division multiplexing (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 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 [KAH22].

4.1 Introduction

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 wireless multi-user multiple-input multiple-output (MIMO) systems. 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 several users. In [ST10], [ST13], and [KWWE07] the singular value decomposition (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 multi-user MIMO system, but the approach is limited to two users. The authors in [HRML21] propose the GSVD for polynomial matrices (PGSVD) and present its application to two-user frequency-selective MIMO channels. In [CDDS19; RDD20] 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 [KAH20], we show how the ML-GSVD can be used for coordinated beamforming in a multi-user MIMO downlink channel with more than two users. Moreover, in [KAH21] and in

Chapter 5 of this thesis, 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 to the number of transmit antennas in the downlink of a coordinated MIMO beamforming system.

The authors in [JNTS15] consider a MIMO orthogonal frequency division multiplexing (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 [JL06; SDL06; KLP11; ZPR12; XTW13; YD18; YD19; DNDT17; IKS20; SBL+18]. The authors in [SK09] 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 [JNTS15], the approach of [SK09] assumes a multigroup multicast scenario and a single-carrier system with single antenna users.

Main contributions. 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 system with multiple antenna users. In our work, 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 rest of this chapter is organized as follows. We introduce the system model and ML-GSVD-based beamforming Section 4.2. In Section 4.2.2 we discuss the state-of-the-art multicast techniques and present the numerical results. Section 4.3 draws the conclusions.

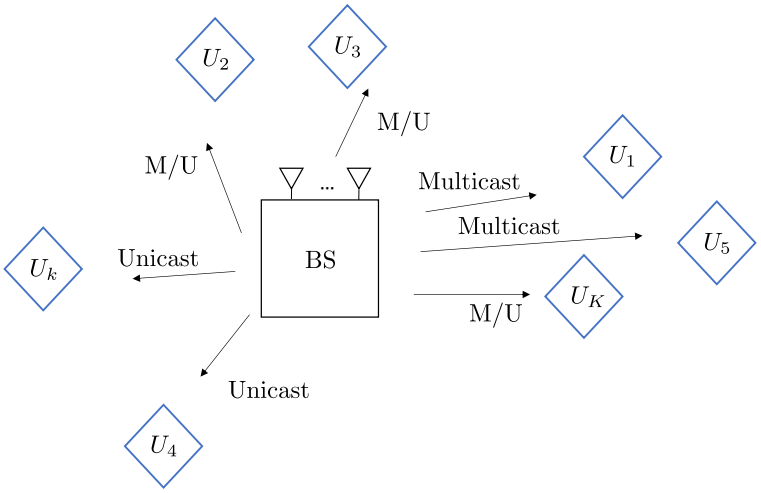


Figure 4.1. Joint multicast and unicast transmission.

4.2 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 unicast transmissions. We view the 3-mode slices of the tensor $\mathcal{H}$ 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 3.2, can be utilized to simultaneously send the common messages to a group of users and private messages to individual users.

4.2.1 System model and ML-GSVD-based beamforming

We consider a downlink MIMO-OFDM system with multicast transmission as depicted in Figure 4.1, 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 additive 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

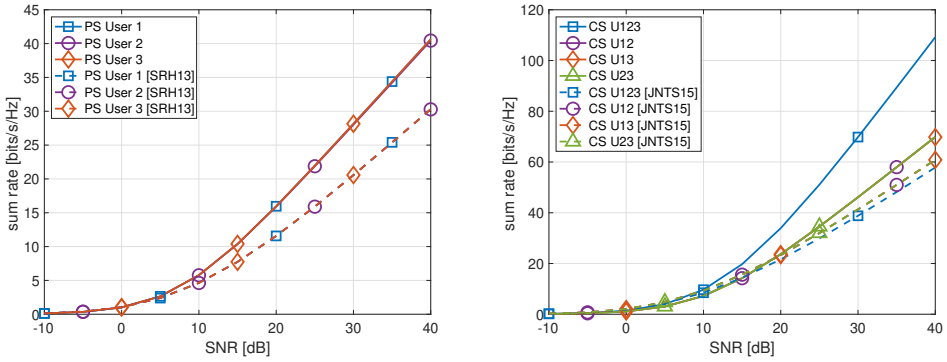
$$\boldsymbol{y}_{k,n} = \boldsymbol{H}_{k,n}\boldsymbol{F}_n\boldsymbol{x}_n + \boldsymbol{n}_{k,n}, \tag{4.1}$$

and at the detector we get

$$\hat{\boldsymbol{y}}_{k,n} = \boldsymbol{W}_{k,n}\left(\boldsymbol{H}_{k,n}\boldsymbol{F}_n\boldsymbol{x}_n + \boldsymbol{n}_{k,n}\right), \tag{4.2}$$

where $\boldsymbol{x}_n$ denotes the transmitted signal on the nth subcarrier that satisfies $\mathbb{E}\left\{\boldsymbol{x}_n\boldsymbol{x}_n^{\mathrm{H}}\right\} = \boldsymbol{I}_{M_T}/M_T$. The matrices $\boldsymbol{F}_n$ and $\boldsymbol{W}_{k,n}$ are the common transmit beamforming and the receive beamforming matrices, respectively. As in (3.2), 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 $\boldsymbol{F}_n = \alpha\left\{\boldsymbol{A}_n^{\mathrm{H}-}\right\}_{\mathcal{C}(\mathcal{Q})}$ (since we focus on the case where $\sum_{k=1}^{K} M_{R_k} > M_T$ (Case 2 of the ML-GSVD), the matrix $\boldsymbol{A}$ is $M_T \times M_T$), and the receive decoding matrices as $\boldsymbol{W}_{k,n} = \left\{\boldsymbol{B}_{k,n}^{\mathrm{H}}\right\}_{\mathcal{R}(\mathcal{Q})}$, 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 or a set of 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 $\boldsymbol{\mathcal{H}}_n \in \mathbb{C}^{M_R \times M_T \times K}$, $M_R = \max\left(M_{R_1},\ldots,M_{R_K}\right)$.

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



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

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

Figure 4.2. Multicast transmission: Private (PS) and common (CS) subspaces. 3 users, $M_T = 9$, $M_R = [6,6,6]$, $r_k = [3,3,3]$. 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.

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 $\boldsymbol{C}$), thus enabling 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.

4.2.2 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 [JNTS15], and in the case of the unicast transmission we use the flexible coordinated beamforming (FlexCoBF) [SRH10; SRH13] scheme as a reference.

The authors in [JNTS15] 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 (LinS)-based precoding matrix, which is given as follows

$$\boldsymbol{F}_n^{\text{LinS}} = \alpha \sum_{k \in \mathcal{S}} \beta_k \boldsymbol{V}_{n,k}, \tag{4.3}$$

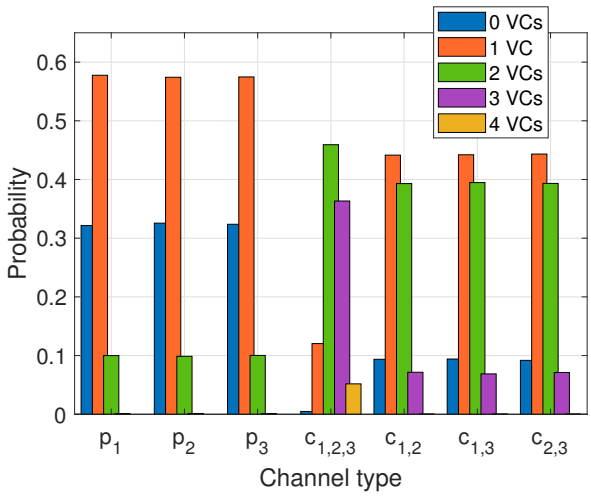


Figure 4.3. Histogram - Private and common channels (ML-GSVD). $M_T = 9$, $M_R = [6,6,6]$. Colors indicate the average number of VCs (streams) at each subcarrier: 0, 1, 2, 3, and 4 VCs, respectively. Labels on the x-axis of the histogram represent the users or the groups of users (p_1 is a private channel to user 1, $c_{1,2}$ is a common channel to users 1 and 2, etc.)

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 the 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

$$\left\| \boldsymbol{F}_n^{\mathrm{LinS}} \right\|_{\mathrm{F}}^2 = P_n. \tag{4.4}$$

We compare this linear sum based precoding in (4.3) with the ML-GSVD-based precoding

$$\boldsymbol{F}_n = \alpha \left\{ \boldsymbol{A}_n^{\mathrm{H}-} \right\}_{\mathcal{C}(\mathcal{Q})}, \tag{4.5}$$

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 [SRH10; SRH13]. 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 [SRH10; SRH13].

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

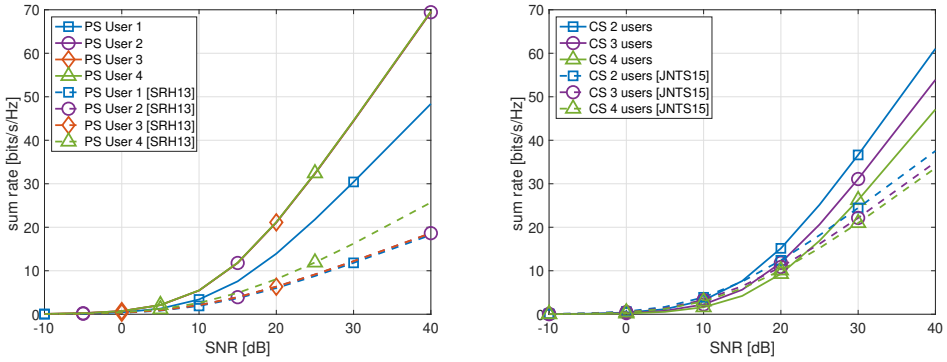
$$\mathrm{R}_{k,n} = \log_2 \det\left(\boldsymbol{I}_{N_r} + \frac{\rho_n}{M_T}\boldsymbol{H}_{n,k}\boldsymbol{F}_n\boldsymbol{F}_n^{\mathrm{H}}\boldsymbol{H}_{n,k}^{\mathrm{H}}\right), \tag{4.6}$$

where ρ_n is the SNR. 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.

Subcar. #	Channels	# of streams
1	p_1, p_2, p_3	r_1, r_2, r_3
2	c_{123}	$\min\left(M_T, M_{R_k}\right)_{k=1}^{3}$
3	c_{12}	$\min\left(M_T, M_{R_1}, M_{R_2}\right)$
4	c_{13}	$\min\left(M_T, M_{R_1}, M_{R_3}\right)$
5	c_{23}	$\min\left(M_T, M_{R_2}, M_{R_3}\right)$

Table 4.1. Joint multicast and unicast transmission using state-of-the-art algorithms (example with 5 subcarriers and 3 users): every subcarrier is utilized for a specific type of messages (private or particular common). In contrast, the ML-GSVD-based scheme allows simultaneous transmission of both private and common messages on one subcarrier, based on the channel conditions of the users. Therefore, in this case, the total number of streams for the kth user is $\sum_{n=1}^{N} r_{k,n}$, and the number of private and common streams on every subcarrier can vary as $p_k \in \{0,\ldots,M_{R_k}\}$ and $c_{\{l\}} \in \{0,\ldots,\min\left(M_{R_{\{l\}}}\right)\}$, $\{l\} \subseteq \{1,\ldots,K\}$, depending on the correlation of the users' channels.

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 3.2). 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 based on the channel conditions of the users. Therefore, in case of the ML-GSVD-based transmission, the total number of streams for the kth user is $\sum_{n=1}^{N} r_{k,n}$, and the number of private and common streams at every subcarrier can vary as $p_k \in \{0\ldots M_{R_k}\}$ and $c_{\{l\}} \in \{0\ldots\min\left(M_{R_{\{l\}}}\right)\}$, $\{l\} \subseteq \{1,\ldots,K\}$, depending on the correlation of the users' channels. For the reference algorithms, we use some subcarriers for the unicast and some for the multicast transmission as it is shown



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

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

Figure 4.4. 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.

in Table 4.1, 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 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 (3.2)). 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 4.1. 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 4.1. In case of the ML-GSVD-based beamforming, the number of virtual channels (VCs) depends on the realization of the channel and can vary for each subcarrier. We calculate the minimum achievable sum rate for each type of channel (common or private) as $\min\left(\sum_{n=1}^{N} R_{k,n}\right)_{k=1}^{K}$ and average the results over $T = 10000$ Monte-Carlo trials.

Figure 4.2(a) depicts the sum rate of the private channels to all three users and Figure 4.2(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 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

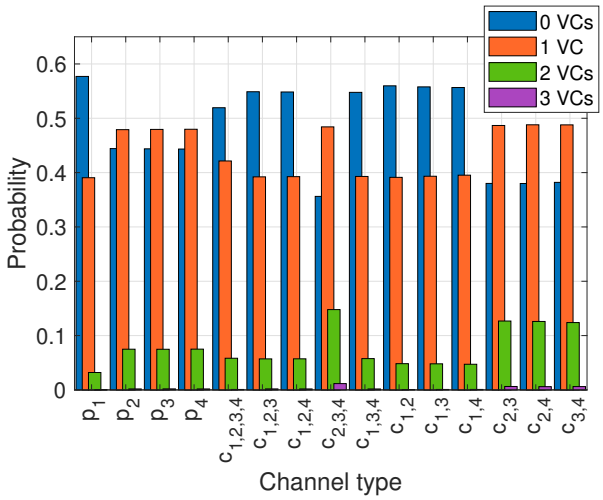


Figure 4.5. Histogram - Private and common channels (ML-GSVD). 4 users, $M_T = 9$, $M_R = [4,5,5,5]$. Colors indicate the average number of VCs (streams) at each subcarrier: 0, 1, 2, and 3 VCs, respectively. Labels on the x-axis of the histogram represent users or groups of users (p_1 is a private channel to user 1, $c_{1,2}$ is a common channel to users 1 and 2, etc.). Averaged over 10000 trials.

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

$$\hat{r}_k = \frac{1}{T} \sum_{n=1}^{N} s_{k,n}, \tag{4.7}$$

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, 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.

Figure 4.3 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 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 3.2, 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 Figure 4.3, 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 the 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 4.1. 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 $\binom{4}{2}$). Figure 4.4(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 [SRH10] and [SRH13]. Figure 4.4(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 [JNTS15] (dashed lines). As it can be seen, the proposed scheme outperforms the

reference techniques for all channel types. Figure 4.5 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 of obtaining a common channel to all users decreases, and there is a higher chance of having 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 user scheduling since it can be performed automatically based on the generalized singular values of the ML-GSVD.

4.3 Conclusions

In this chapter, as an application of the ML-GSVD proposed in Chapter 3, we have considered multi-user 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 three cases of the ML-GSVD correspond to the transmission of private or common messages (or both) to different sets of users. Our numerical results show that the ML-GSVD outperforms the reference multicast and unicast precoding schemes in terms of the average sum rate.

Utilizing the ML-GSVD for non-orthogonal multiple access

5

Future generations of wireless networks become more and more demanding due to the exponential growth of mobile traffic with billions of connected smart devices. In this chapter of the thesis, we focus on non-orthogonal multiple access (NOMA), which has been considered as a promising technique for 5G and beyond wireless networks. We consider a power-domain downlink MIMO-NOMA system with an arbitrary number of users and propose to design the precoding and decoding matrices based on the multilinear generalized singular value decomposition (ML-GSVD) that we have presented in Chapter 3. Moreover, we demonstrate how the generalized singular values of the ML-GSVD can be used for the power allocation. We also compare the proposed MIMO-NOMA scheme with orthogonal multiple access (OMA) techniques and provide various numerical results [KAH21].

5.1 Introduction

The substantial growth of the global data traffic puts higher demands on 5G and beyond networks. The extensive use of smart devices and the increasing interest in the Internet of things (IoT) require reliable low-latency communication, massive connectivity, and increased data rates. Due to limited bandwidth and limited power resources, providing a sufficient quality of service (QoS) becomes more and more challenging.

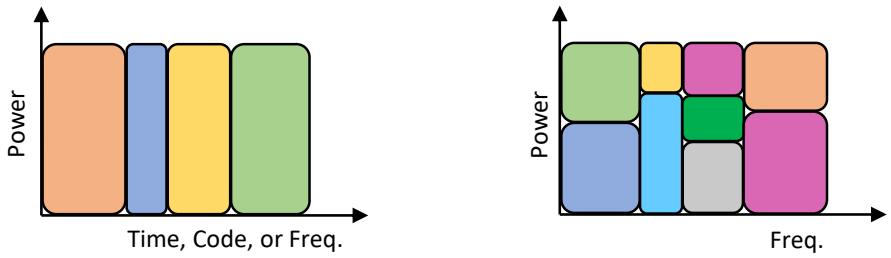
In conventional orthogonal multiple access (OMA) techniques (Figure 5.1(a)), the users exploit the orthogonal resources in time, frequency, code, or in the joint time-frequency domain to avoid multiple access interference. However, since each resource block can only support one user equipment (UE), the number of served users is limited. As an alternative, non-orthogonal multiple access (NOMA) has been proposed to be employed in future wireless communication systems [Kiz16]. NOMA allows using the same non-orthogonal resources to transmit the data of different users simultaneously.

It has gained a significant attention due to its potential to considerably improve spectral efficiency, enhance connectivity, and reduce latency [DWD+18]. In general, NOMA solutions can be separated into two main groups, power-domain and code-domain multiplexing [DLK+17]. However, the majority of the NOMA contributions focus on the power-domain NOMA (Figure 5.1(b)) due to its relatively simple implementation, tolerable user fairness, and spectral efficiency trade-off [ATG+18].

Several overviews and surveys [ATG+18; DWD+18; IADK17; ZLL+18; MCBA20; DLK+17; BSL+15; CQC+18] on NOMA techniques have been published in recent years. They present the basic principles of the NOMA technique, discuss the open issues, and compare it with the performance of OMA-based transmission. The authors in [DWD+18] additionally highlight the key challenges and the future research trends in NOMA exploration, which includes performance analysis, beamforming and receiver design, channel estimation, resource allocation, and an extension to MIMO. The study in [AHA19] investigates efficient user clustering and power allocation algorithms for MIMO-NOMA systems. The authors derive the sum rate capacity for the uplink system with a massive antenna array at the base station (BS) and single antenna UEs. A MIMO-NOMA downlink scenario with multiple antenna users has been considered in [DAP16]. The authors propose to use user pairing to enhance the NOMA performance and consider the identity matrix as a precoding matrix to reduce the system overhead. More sophisticated precoding and decoding based on QR and generalized singular value decomposition (GSVD) decompositions have been proposed in [DDP16; CDDS19; RDD21; RDD20; QV20; HD19; MDFT16]. However, the aforementioned schemes are restricted to systems with two users in one resource block due to the limitation that the GSVD uses only two matrices.

In Chapter 3 and in [KAH22], we have proposed the ML-GSVD as an extension of the GSVD to decompose more than two matrices with one common dimension. Compared to the other multidimensional extensions of the GSVD, the ML-GSVD inherits the properties of the original decomposition, such as orthogonality and the structure of the diagonal matrices.

Main contributions. This chapter of the thesis focuses on power-domain NOMA, which employs superposition coding (SC) at the transmitter and successive interference cancellation (SIC) at the receiver. We propose a new ML-GSVD-based NOMA scheme for downlink MIMO systems with more than two users in one resource block. We show how the ML-GSVD can be applied to MIMO-NOMA systems with an arbitrary number of users. Additionally, we propose a new power allocation scheme based on the generalized singular values of the ML-GSVD and compare it to the schemes from the literature.



(a) OMA. **(b)** NOMA.

Figure 5.1. Orthogonal and non-orthogonal multiple access (power domain).

The remainder of this chapter is organized as follows. We first provide the system model of the considered power-domain downlink MIMO-NOMA system in Section 5.2. Then, we present the design of the precoding and decoding matrices based on the ML-GSVD and consider different scenarios exploiting the properties of this recently proposed decomposition in Section 5.3. At the end of Section 5.3, we demonstrate how the generalized singular values of the ML-GSVD can be used for power allocation. In Section 5.5, we also show the numerical results to evaluate the performance of the ML-GSVD-based NOMA scheme in terms of the achievable rate and compare it to OMA.

5.2 System model

Let us consider a downlink MIMO-NOMA communication system with one base station (BS) and K users, where the BS and the kth user are equipped with M_T and M_{R_k} antennas, respectively (Figure 5.2). Assuming flat fading, the $M_{R_k} \times M_T$ channel matrix from the BS to the kth user is denoted as $\boldsymbol{H}_k$. In a downlink NOMA network, the BS transmits a superposition of the desired signals of K users with different allocated power to all K users. The power allocation coefficients α_k are inversely proportional to the channel conditions of the users (e.g., path loss). The precoding matrix at the BS is denoted as $\boldsymbol{P} \in \mathbb{C}^{M_T \times M_T}$. Then, the transmitted signal can be written as $\boldsymbol{x} = \boldsymbol{P}\boldsymbol{s}$, and $\boldsymbol{s} = \left[s_1, \ldots, s_{M_T}\right]^{\mathrm{T}} \in \mathbb{C}^{M_T \times 1}$ is a combined signal for all users, given by

$$\boldsymbol{s} = \sum_{k=1}^{K} \boldsymbol{\Lambda}_k \tilde{\boldsymbol{s}}_k, \tag{5.1}$$

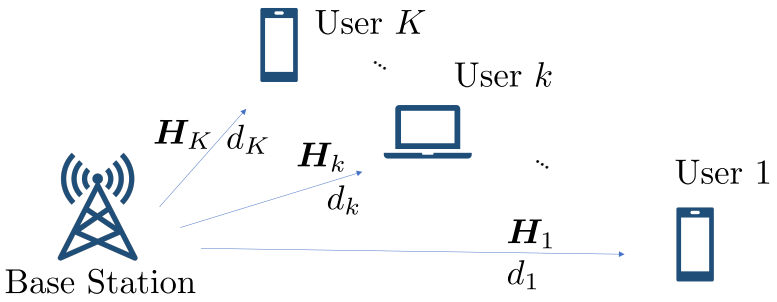


Figure 5.2. Downlink MIMO-NOMA system with K users.

where $\boldsymbol{\Lambda}_k = \text{diag}\left(\alpha_{k,1}, \ldots, \alpha_{k,M_T}\right) \in \mathbb{R}^{M_T \times M_T}$ and $\tilde{\boldsymbol{s}}_k = \left[\tilde{s}_{k,1}, \ldots, \tilde{s}_{k,M_T}\right]^{\mathrm{T}} \in \mathbb{C}^{M_T \times 1}$ contain the power allocation coefficients and the data for user k, respectively. The power allocation coefficients satisfy $\sum_{k=1}^{K} \boldsymbol{\Lambda}_k^2 = \boldsymbol{I}_{M_T}$. Consequently, the received signal at the kth user can be expressed as

$$\boldsymbol{y}_k = \frac{1}{\sqrt{d_k^{\eta}}} \boldsymbol{H}_k \boldsymbol{P} \boldsymbol{s} + \boldsymbol{n}_k \in \mathbb{C}^{M_{R_k} \times 1}, \tag{5.2}$$

where $\boldsymbol{n}_k$ denotes the additive Gaussian noise vector of user k, whose elements are i.i.d. complex Gaussian random variables with zero mean and unit variance. The quantity $\frac{1}{\sqrt{d_k^{\eta}}}$ denotes the large-scale fading, where d_k is the distance between the BS and the kth user, and η is the path loss exponent [CDDS19]. We assume that $\mathbb{E}\{\tilde{\boldsymbol{s}}\tilde{\boldsymbol{s}}^{\mathrm{H}}\} = \boldsymbol{I}_{M_T}$. After applying the decoding matrix $\boldsymbol{D}_k$ at the kth user we get

$$\boldsymbol{D}_k \boldsymbol{y}_k = \frac{1}{\sqrt{d_k^{\eta}}} \boldsymbol{D}_k \boldsymbol{H}_k \boldsymbol{P} \boldsymbol{s} + \boldsymbol{D}_k \boldsymbol{n}_k. \tag{5.3}$$

In the following section we discuss the design of the precoding and the decoding matrices.

5.3 ML-GSVD-based NOMA scheme

In the following, we will show how the ML-GSVD can be used for defining the precoding and decoding matrices $\boldsymbol{P}$ and $\boldsymbol{D}_k$ that will jointly diagonalize the channel matrices $\boldsymbol{H}_k$. Yet before we move to the proposed scheme, we encourage the readers to refer back to the definition (3.2) and the Cases 1, 2, and 3 of the ML-GSVD provided earlier in

Section 3.2 to enhance the comprehension of the upcoming discussions. Recalling the three cases briefly, in the first case, the columns of the matrix $\boldsymbol{A}$ in (3.2) are shared for all factorizations, and the ML-GSVD provides only the *common* subspace for all jointly decomposed matrices. Therefore, in this chapter, we mostly focus on this case. We use the aforementioned common subspace for the MIMO-NOMA transmission of the combined signals to all K users. The second case provides both *private* and *common* subspaces for all K matrices or for groups of matrices. In this scenario, the common subspace can be used for the NOMA transmission of the combined signals, and the private subspace might be used for confidential messages to ensure security. In the third case, the entries of $\boldsymbol{C}_k$ only contain ones and zeros, i.e., only a private subspace exists, and the tensor slices do not share any common factors. Thus, we do not consider Case 3 of the ML-GSVD for the NOMA transmission. However, since it allows isolating the private channels to users, Case 3 can be utilized in conventional OMA systems without superposition coding.

Using the ML-GSVD of the channel matrices $\boldsymbol{H}_k$, the precoding matrix $\boldsymbol{P}$ can be defined as $\boldsymbol{P} = \frac{\sqrt{P}}{\beta}\boldsymbol{A}^{\mathrm{H}-}$, where P is the total transmit power at the BS, and β is a power normalization coefficient. Since we focus on the case with only common subspaces, (Case 1 of the ML-GSVD), the matrix $\boldsymbol{A}$ is $M_T \times M_T$ and invertible. The decoding matrix of the kth user is defined as $\boldsymbol{D}_k = \boldsymbol{B}_k^{\mathrm{H}}$. Consequently, the received data at the kth user can be written as

$$\boldsymbol{B}_k^{\mathrm{H}}\boldsymbol{y}_k = \frac{\sqrt{P}}{\beta\sqrt{d_k^{\eta}}}\boldsymbol{B}_k^{\mathrm{H}}\boldsymbol{H}_k\boldsymbol{A}^{\mathrm{H}-}\boldsymbol{s} + \boldsymbol{B}_k^{\mathrm{H}}\boldsymbol{n}_k \tag{5.4}$$

$$= \frac{\sqrt{P}}{\beta\sqrt{d_k^{\eta}}}\boldsymbol{C}_k\boldsymbol{s} + \tilde{\boldsymbol{n}}_k, \tag{5.5}$$

where $\tilde{\boldsymbol{n}}_k = \boldsymbol{B}_k^{\mathrm{H}}\boldsymbol{n}_k$. After applying the ML-GSVD, the MIMO channels $\boldsymbol{H}_k$ of K users can be considered as independent parallel sub-channels

$$y_{k,n} = \frac{\sqrt{P}}{\beta\sqrt{d_k^{\eta}}}c_{k,n}s_n + \tilde{n}_{k,n}, \tag{5.6}$$

where $y_{k,n}$, $c_{k,n}$, s_n, and $\tilde{n}_{k,n}$ are the nth elements of $\boldsymbol{y}_k$, $\mathrm{diag}(\boldsymbol{C}_k)$, $\boldsymbol{s}$, and $\tilde{\boldsymbol{n}}_k$, respectively. For notational simplicity, we assume that the first user is the farthest, and the Kth user is the closest (strongest), so that $\left\|\frac{1}{\sqrt{d_1^{\eta}}}\boldsymbol{H}_1\right\|_{\mathrm{F}} \leq \ldots \leq \left\|\frac{1}{\sqrt{d_K^{\eta}}}\boldsymbol{H}_K\right\|_{\mathrm{F}}$. In NOMA, the highest power is assigned to the user with the weakest channel [DLK+17]. Therefore, the power allocation coefficients satisfy $\alpha_{1,n}^2 > \cdots > \alpha_{K,n}^2$, $n \in \{1,\ldots,M_T\}$.

Depending on the ratio between the number of transmit and receive antennas, according to the three cases of the ML-GSVD described above, the value of $c_{k,n}$ in (5.6) can be equal to 0, 1, or $\sigma_{k,n}$, corresponding to private or common subspaces. Notice that signals that are transmitted on the sub-channels that correspond to private subspaces of the kth user will not be detected by other users. Therefore, if $c_{k,n} = 1$, on the nth sub-channel the user k receives

$$y_{k,n} = \frac{\sqrt{P}}{\beta\sqrt{d_k^\eta}}\tilde{s}_n + \tilde{n}_{k,n}. \tag{5.7}$$

Thus, the individual information rate of user k is given as

$$R_n^{(k)} = \log_2\left(1 + \frac{P}{\beta^2 d_k^\eta N_0}\right), \tag{5.8}$$

whereas the rate of other users $l \neq k$ is zero. In this case, the signal s_n in (5.1) can be designed in a way that it only contains the data of user k, which will correspond to a conventional OMA signal. For the NOMA transmission, we use the sub-channels that are shared between several users, i.e., we utilize the common subspace of the ML-GSVD to transmit the superimposed signals of multiple users. For simplicity, we assume that $M_T \leq M_{R_k}$ for all k. This scenario corresponds to the Case 1 of the ML-GSVD, where the common subspace is shared among all users, and no private subspace exists for any user. In general, Case 2 of the ML-GSVD can also be used in NOMA systems. Since this case combines both private and common subspaces, only the channels corresponding to the common subspace should be used for the NOMA communication. Assuming transmission via the common channel, we can rewrite (5.6) as

$$y_{k,n} = \frac{\sqrt{P}}{\beta\sqrt{d_k^\eta}}\sigma_{k,n}s_n + \tilde{n}_{k,n}, \tag{5.9}$$

where $\sigma_{k,n}^2$ is the generalized singular value of the ML-GSVD corresponding to the nth common sub-channel, and the signal s_n contains the combined data of all users. To recover the individual signals, users have to perform the SIC process. Therefore, the information rate of $\tilde{s}_{k,n}$ for the kth user is given by

$$R_n^{(k)} = \log_2\left(1 + \frac{P\alpha_{k,n}^2\sigma_{k,n}^2}{\sum_{j=k+1}^{K} P\alpha_{j,n}^2\sigma_{j,n}^2 + \beta^2 d_k^\eta N_0}\right), \tag{5.10}$$

where N_0 is the noise power. The closest user, user K, needs to decode the signals of all users before decoding his own. Consequently, assuming perfect SIC, the achievable rate of the Kth user is given by

$$R_n^{(K)} = \log_2\left(1 + \frac{P\alpha_{K,n}^2\sigma_{K,n}^2}{\beta^2 d_K^{\eta} N_0}\right). \tag{5.11}$$

Next, we consider the choice of the power allocation coefficients $\alpha_{k,n}$. In general, power allocation is considered one of the main problems in NOMA systems [ATG+18]. In this study, we propose to allocate the power based on the ML-GSVD generalized singular values.

It has been mentioned before that the $\sigma_{k,n}$s from the ML-GSVD satisfy $0 < \sigma_{k,n} < 1$, and the stronger channel gets a larger generalized singular value, i.e., $\sigma_{K,n} > \sigma_{1,n}$. This relation can be used to calculate the power allocation coefficients $\boldsymbol{\alpha}_n = \left[\alpha_{1,n}, \ldots, \alpha_{K,n}\right]$ that account for the channel conditions of each user. Let us define the matrix $\tilde{\boldsymbol{C}}$ as

$$\tilde{\boldsymbol{C}} = 1 \oslash \begin{bmatrix} \operatorname{diag}\left(\boldsymbol{C}_1\right)^{\mathrm{T}} \\ \vdots \\ \operatorname{diag}\left(\boldsymbol{C}_K\right)^{\mathrm{T}} \end{bmatrix}, \tag{5.12}$$

where the matrices $\boldsymbol{C}_k$ are defined in (3.2), and $\oslash$ denotes the element-wise division. The values $\tilde{\boldsymbol{C}}_{(:,n)} = \left[\frac{1}{\sigma_{1,n}}, \ldots, \frac{1}{\sigma_{K,n}}\right]^{\mathrm{T}}$ are inversely proportional to the generalized singular values of the channel matrices $\boldsymbol{H}_k$. Thus, we can express the power allocation as $\boldsymbol{\alpha}_n = \hat{\boldsymbol{C}}_{(:,n)}$, where $\hat{\boldsymbol{C}}$ denotes the matrix $\tilde{\boldsymbol{C}}$ with normalized columns. Such a choice of $\alpha_{k,n}$s means that the power will be allocated according to the channel conditions that are represented by the generalized singular values of the channel.

5.4 Simulation results

In this section, we present the numerical results to assess the performance of the proposed ML-GSVD-based MIMO-NOMA scheme. For the simulations, we consider a downlink MIMO-NOMA scenario with one BS and three users. The noise power is calculated as $N_0 = -174 + \log_{10} B$, where the bandwidth $B = 1\,\mathrm{MHz}$. The sum rate is expressed as $R_{\mathrm{sum}} = \sum_{k=1}^{K}\sum_{n=1}^{M_T} R_n^{(k)}$, while the individual user rates are calculated as $R^{(k)} = \sum_{n=1}^{M_T} R_n^{(k)}$, where $R_n^{(k)}$ is defined in (5.10). All results are averaged over 1000 Monte Carlo trials.

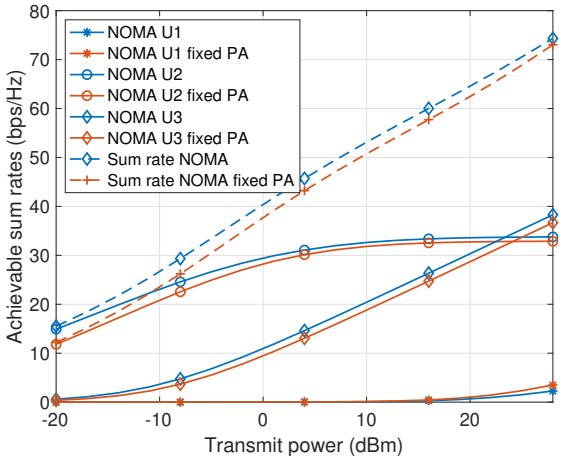


Figure 5.3. Rates vs. Transmit power. Comparison of the fixed and ML-GSVD-based power allocation schemes. Parameters: $d_1 = 200\,\text{m}, d_2 = 15\,\text{m}$, and $d_3 = 10\,\text{m}$, $\eta = 3$, $M_T = 3$, $M_{R_k} = 4$.

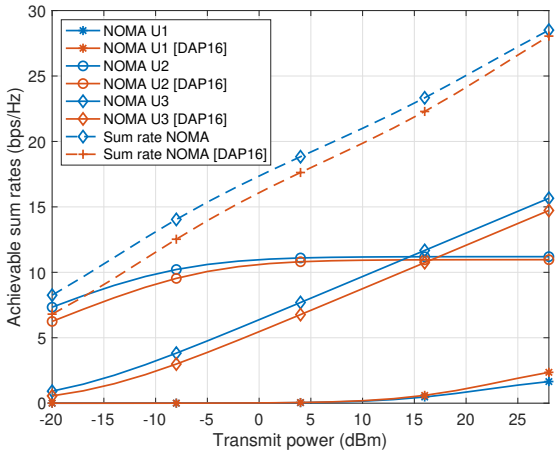


Figure 5.4. Rates vs. Transmit power. Parameters: $d_1 = 200\,\text{m}, d_2 = 15\,\text{m}$, and $d_3 = 10\,\text{m}$, $\eta = 3$, $M_T = 1$, $M_{R_k} = 4$.

In the first simulation, each user is equipped with $M_{R_k} = 4$ antennas, and the BS has $M_T = 3$ antennas. This scenario corresponds to the case where the ML-GSVD decomposition provides only the common subspace (Cases 1), i.e., the users receive superimposed data of all users. We set the distances between the BS and the kth user to $d_1 = 200\,\text{m}$, $d_2 = 15\,\text{m}$, and $d_3 = 10\,\text{m}$, respectively. The path loss exponent is equal to $\eta = 3$. Figure 5.3 shows the sum rate and the individual rates of each user with different power allocation coefficients. For the fixed power allocation, the coefficients are calculated according to the distances and a path loss, while in the case of the ML-GSVD-based power allocation, the coefficients are calculated dynamically

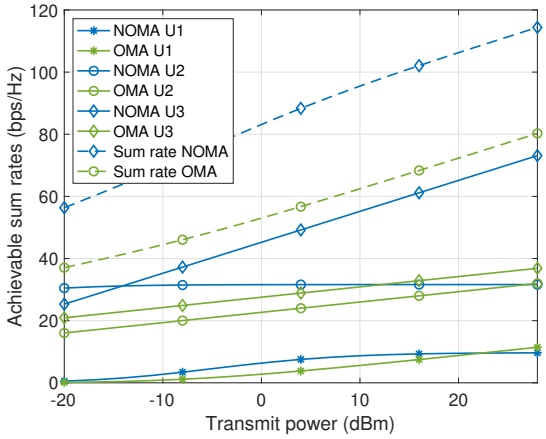


Figure 5.5. Rates vs. Transmit power. Comparison of the rates achieved by OMA and NOMA. Parameters: $d_1 = 300\,\text{m}, d_2 = 10\,\text{m}$, and $d_3 = 5\,\text{m}$, $\eta = 2$, $M_T = 3$, $M_{R_k} = [3,4,6]$.

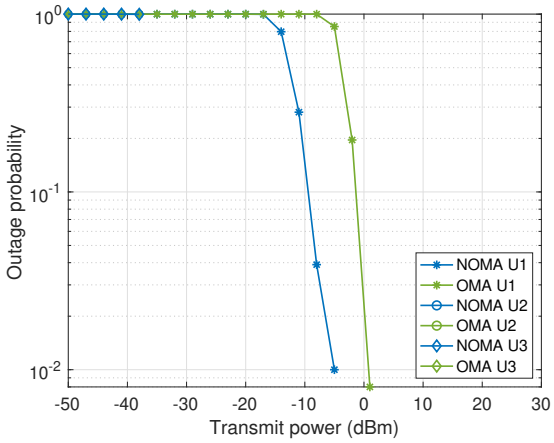


Figure 5.6. Outage probability vs. Transmit power. Comparison between OMA and NOMA. Parameters: $d_1 = 300\,\text{m}, d_2 = 10\,\text{m}$, and $d_3 = 5\,\text{m}$, $\eta = 2$, $M_T = 3$, $M_{R_k} = [3,4,6]$.

based on the generalized singular values. In both cases, the precoding and decoding matrices are determined based on the proposed ML-GSVD-NOMA scheme. As it can be observed, the ML-GSVD-based dynamic power allocation shows a better performance, especially for the sum rate and the individual rates of the second and third users, which is explained by the fact that the fixed power allocation does not adapt to the small changes in the channels.

In the second simulation, we compare the performances of the proposed ML-GSVD-based scheme with the technique proposed in [DAP16]. We consider the scenario,

where the BS has one antenna[1], and the users have four antennas each. The distances are $d_1 = 200\,\text{m}$, $d_2 = 15\,\text{m}$, and $d_3 = 10\,\text{m}$, respectively. The path loss exponent is set to $\eta = 3$. The results are shown in Figure 5.4. As it can be seen, the proposed scheme outperforms the reference scheme in terms of the achievable sum rate and the rates of the second and third users. It should be noted, that the scheme in [DAP16] is limited to the scenario where $M_T < M_R$ and assumes that all users have an equal number of antennas.

In the next simulation, we consider an asymmetric scenario, where the BS has $M_T = 3$ antennas, and three users are equipped with 3, 4, and 6 antennas, respectively, and compare the performance of the NOMA and OMA schemes. The simulation results are depicted in Figure 5.5. One can see that the proposed ML-GSVD outperforms the OMA scheme in terms of the achievable rates of all three users. As it can be observed, the achievable rate of the third (strongest) user is much higher in case of NOMA, especially with the increased power, due to the absence of interference from the weaker users. The rates of the first and the second users are approximately constant at high powers because they consider the signals of the stronger users as noise. Figure 5.6 depicts the outage probabilities for NOMA and OMA. The target rates are set to 2 bps/Hz, 4 bps/Hz, and 5 bps/Hz, respectively. As it can be observed, the outage probability of the weak NOMA user is smaller than for the OMA user which makes the NOMA transmission more reliable. Whereas with a given transmit power, the strong users maintain the target rates without outages.

5.5 Conclusions

In this chapter of the thesis, we have presented a new ML-GSVD-based NOMA transmission technique that can be applied in power-domain MIMO-NOMA downlink communication systems with multiple users. It utilizes the ML-GSVD proposed in Chapter 3 to design the precoding and decoding matrices that jointly diagonalize the channels between the BS and the users. Compared to the GSVD-based technique proposed in [CDDS19], the ML-GSVD can support more than two users on one frequency resource, and the common subspace of the ML-GSVD can be employed to transmit the combined signals of the K users. Additionally, we have presented a simple power allocation technique based on the generalized singular values of the ML-GSVD that outperforms the conventional fixed power allocation. The performance of the

[1] The authors in [DAP16] assume that the number of transmit antennas is equal to the number of clusters, which means that each cluster with K users is maintained by one antenna. Therefore, for a fair comparison, we assume a single antenna BS in this simulation.

proposed scheme has been evaluated in terms of the achievable rate and compared to traditional OMA and to the state-of-the-art NOMA techniques.

Precoding and rate splitting with the ML-GSVD for MIMO broadcast channels

6

In this chapter of the thesis, we present a precoder design for multi-user multiple-input multiple-output (MU-MIMO) broadcast systems with rate splitting at the transmitter. The proposed scheme applies to both underloaded and overloaded communication systems and supports the transmission of multiple common and private streams. We show how the generalized singular value (GSVD) and multilinear generalized singular value (ML-GSVD) decompositions can be used to define the number of common and private streams and adjust the message split. Additionally, we present transmit precoding and receive combining designs that allow the simultaneous transmission of common and private streams but do not require successive interference cancellation (SIC) at the receivers and can be used in cases where the total number of streams does not exceed the number of transmit antennas [KAH23].

6.1 Introduction

An increasing number of smart devices connected to the global network puts higher demands on every successive generation of mobile communications. The sixth generation (6G) networks are anticipated to offer massive connectivity with ultra-reliability and high quality of service, which in turn, requires more sophisticated technologies that will provide more efficient use of the bandwidth and power resources. To this end, the authors in [MCL18; MDC+22] propose the rate splitting multiple access (RSMA) framework, which encompasses the advantages of two extreme interference management strategies, namely, space division multiple access (SDMA) and non-orthogonal multiple access (NOMA), while enabling their joint and flexible use.

In recent years, the RSMA has attracted particular attention from researchers and continues to gain popularity. However, the technology has not yet reached its maturity, and there are still various challenges to be addressed and underdeveloped research directions to be investigated [MDC+22]. Most of the publications that overview and

explore rate splitting are dedicated to the multiple-input single-output (MISO) systems with single-antenna users [MCL18; MDC+22; JC16; MC20]. Yet, several papers discuss the extension of RSMA to multiple-input multiple-output (MIMO) systems. The authors in [HRC17] characterize the achievable degree-of-freedom (DoF) regions of a general two-receiver MIMO broadcast (BC) and MIMO interference channels (IC) with imperfect channel state information at the transmitter (CSIT). The achievable rates of linearly precoded BC RSMA schemes with K users have been investigated in [LYS21; LY18]. Moreover, rate splitting has found an application in massive MIMO systems in [DCGC16; DMC21]. The precoder design and optimization for the multi-user (MU) MIMO downlink with rate splitting has been studied in [MMDC22]. The authors additionally validate their results via link-level simulations. However, the model in [MMDC22] is limited to a scenario where all receivers have the same number of antennas. Linear and non-linear precoding and stream combining techniques for RSMA in MIMO systems with imperfect CSIT have been developed in [FLC20; FDLC21]. Moreover, in [FL22] the authors consider another essential RSMA problem and present adaptive power allocation schemes to distribute the transmit power between the common and private streams with a reduced computational cost. The precoder design for an underloaded and critically loaded[1] downlink MIMO system with rate splitting has been presented in [KS22; DKS22].

Main contributions. Similarly, in this contribution, our focus falls on the precoder design for MIMO broadcast systems with common and private streams where both the base station (BS) and the receivers are equipped with multiple antennas. However, in contrast to [FLC20; FDLC21; FL22], we do not limit the number of common streams to one and allow the transmission of multiple common and private streams. We propose a precoding design for the common and private streams based on the multicast precoder design in [JNTS15] and the FlexCoBF (flexible coordinated beamforming) scheme in [SRH13]. In contrast to the schemes in [KS22; DKS22], the proposed techniques are applicable in both under- and overloaded communication scenarios. Moreover, we show how the generalized singular value (GSVD) [Van76; PS81] and the multilinear generalized singular value (ML-GSVD) [KAH20; KAH22] decompositions can be used to define the number of common and private streams and adjust the message split. The ML-GSVD extends the GSVD to the multiple matrices case, enabling a joint decomposition of more than two matrices. Therefore, it applies to systems with more than two users where the common streams can also be transmitted to selected groups of users (generalized or multi-layer RS, where the number of SICs corresponds to the number of groups [MDC+22]). Additionally, we present the transmit precoding

[1] In the under- and critically loaded MU-MIMO systems, the total number of receive antennas is less or equal to the number of antennas at the BS.

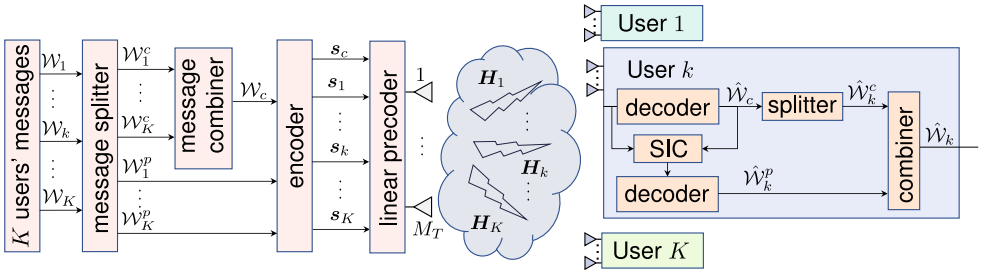


Figure 6.1. One-layer MIMO rate-splitting multiple access with K users.

and receive combining designs that enable the simultaneous transmission of common and private streams but do not require successive interference cancellation (SIC) at the receivers and can be used in cases where the total number of streams does not exceed the number of transmit antennas.

The remainder of this chapter is organized as follows. Section 6.2 provides the system model and a brief description of the considered RSMA system. Then, we present a technique to design precoders for common and private streams in Section 6.3. Section 6.3.2 considers the special case of rate splitting, where the common and private streams can be decoded jointly without SIC at the receivers. Section 6.4 shows the simulation results, and Section 6.5 concludes the chapter.

6.2 System model

We consider a multi-user MIMO downlink system where the base station (BS) equipped with M_T transmit antennas serves K users, each equipped with M_{R_k} receive antennas. The $M_{R_k} \times M_T$ channel matrix from the BS to the kth user is represented by $\boldsymbol{H}_k$, and the transmitted data is denoted as $\boldsymbol{x} \in \mathbb{C}^{M_T \times 1}$. The transmitted signal is subject to a total power constraint $\mathbb{E}\{\|\boldsymbol{x}\|^2\} \leq P_t$. Consequently, the signal received at the kth user is given by

$$\boldsymbol{y}_k = \boldsymbol{H}_k \boldsymbol{x} + \boldsymbol{n}_k \in \mathbb{C}^{M_{R_k} \times 1}, \tag{6.1}$$

where $\boldsymbol{n}_k \sim \mathcal{CN}\left(\boldsymbol{0}, \sigma_k^2 \boldsymbol{I}_{M_{R_k}}\right)$ is an Additive White Gaussian Noise (AWGN) vector whose elements are i.i.d. complex Gaussian random variables with zero mean and variance σ_k^2.

In this study, we consider the rate splitting technique to transmit the data to multiple users. In one-layer RSMA every user's message set $\mathcal{W}_k$ is split into common and

private parts denoted as $\mathcal{W}_k^c$ and $\mathcal{W}_k^p$, respectively (see Figure 6.1). The common parts are then combined and encoded together into a common stream $\boldsymbol{s}_c$ of size l_c, whereas the private message sets $\mathcal{W}_k^p$ are encoded independently into K private streams $\boldsymbol{s}_k$ of size l_k, $k = \{1, \ldots, K\}$. Consequently, the resulting data stream vector to be transmitted is equal to $\boldsymbol{s} = \left[\boldsymbol{s}_c^{\mathrm{T}}, \boldsymbol{s}_1^{\mathrm{T}}, \ldots, \boldsymbol{s}_K^{\mathrm{T}}\right]^{\mathrm{T}} \in \mathbb{C}^{l \times 1}$, where $l = l_c + \sum_{k=1}^{K} l_k$ denotes the total number of data streams. We assume that the total number of common streams l_c cannot exceed $\min\left(M_T, M_{R_i}\right)_{i=1}^{K}$ and the total number of private streams satisfies $\sum_{k=1}^{K} l_k \leq M_T$. Subsequently, the common stream $\boldsymbol{s}_c$ and the K private streams $\boldsymbol{s}_k$ are precoded via the precoding matrices $\boldsymbol{P}_c \in \mathbb{C}^{M_T \times l_c}$ and $\boldsymbol{P}_k \in \mathbb{C}^{M_T \times l_k}$, respectively. Thus, the combined transmit precoding matrix can be written as $\boldsymbol{P} = [\boldsymbol{P}_c, \boldsymbol{P}_1, \ldots, \boldsymbol{P}_K] \in \mathbb{C}^{M_T \times l}$. Then, the resulting transmitted signal is expressed as

$$\boldsymbol{x} = \boldsymbol{P}_c \boldsymbol{s}_c + \sum_{k=1}^{K} \boldsymbol{P}_k \boldsymbol{s}_k = \boldsymbol{P}\boldsymbol{s} \in \mathbb{C}^{M_T \times 1}. \tag{6.2}$$

The signal received at the kth user can be rewritten as

$$\begin{aligned} \boldsymbol{y}_k &= \boldsymbol{H}_k \boldsymbol{x} + \boldsymbol{n}_k && (6.3)\\ &= \boldsymbol{H}_k (\boldsymbol{P}_c \boldsymbol{s}_c + \sum_{k=1}^{K} \boldsymbol{P}_k \boldsymbol{s}_k) + \boldsymbol{n}_k && (6.4)\\ &= \boldsymbol{H}_k \boldsymbol{P}\boldsymbol{s} + \boldsymbol{n}_k && (6.5) \end{aligned}$$

with $\mathbb{E}\{\boldsymbol{s}\boldsymbol{s}^H\} = \boldsymbol{I}_l$. As a result, the total power is subject to the constraint $\|\boldsymbol{P}\|_{\mathrm{F}}^2 \leq P_t$. In the following sections, we present the precoder designs and a message splitting approach based on the GSVD (for two users) and the ML-GSVD (for more than two users). Therefore, to ensure clarity and avoid unnecessary repetition, we invite readers to review briefly the definition (3.2) and the Cases 1, 2, and 3 of the ML-GSVD provided in Section 3.2.

6.3 Precoder design for common and private streams

In this section, we present two methods to design the precoders for the RSMA system with private and common streams. Section 6.3.1 focuses on the general case, where the total number of streams can exceed the number of transmit antennas. Whereas, in Section 6.3.2 we consider a special case where the total number of data streams does not exceed the number of transmit antennas and show that common and pri-

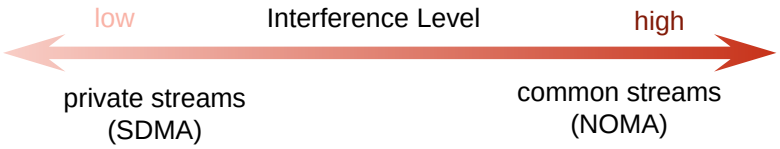


Figure 6.2. Interference level based strategies in RSMA.

vate messages can be precoded and decoded jointly without performing SIC at the receivers.

6.3.1 Channel assignment and precoder design

Considering the MIMO scenario where users can receive multiple data streams, the number of common and private streams transmitted in parallel has to be determined. Moreover, the authors in [MDC+22] showed (for the MISO case) that RSMA can be reduced to SDMA or NOMA depending on the strength or orthogonality of the users' channels (see Figure 6.2). In this regard, the ML-GSVD and the GSVD provide an elegant and straightforward way to determine the channels' strengths and alignment. As an example, let us consider the GSVD of two channel matrices $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$ with the diagonal terms as in (2.24) and (2.25). The GSVD separates the channels into three independent subspaces (or parallel virtual channels (VCs)): a common subspace ($\mathcal{S}_{12}$) shared between the two users and two private subspaces ($\mathcal{S}_1$ and $\mathcal{S}_2$) that are orthogonal to each other. Bases for these subspaces are given by the corresponding columns of the matrix $\boldsymbol{A}$ in (2.23). The presence or absence of the private and common subspaces depends on the channel conditions. Therefore, we use the GSVD to determine the existence of a common subspace to transmit the common messages and, consequently, the selected number of common streams, which is equal to the size c_k of the common subspace. For scenarios with more than two users, the ML-GSVD can be applied. It estimates the common subspace shared between all K users and between groups of users, which can be further used in multi-layer rate splitting. Again, the bases for these subspaces are given by the corresponding columns of the matrix $\boldsymbol{A}$ of the ML-GSVD in (3.2).

Subsequently, the defined number of common and private streams can be used for the design of the precoding matrices $\boldsymbol{P}_c$ and $\boldsymbol{P}_k$ in (6.4). The authors in [JNTS15] propose a multicast precoding scheme for MIMO orthogonal frequency division multiplexing (OFDM) systems to transmit common messages to a selected group of users. It can

also be generalized for MIMO-RSMA to transmit the common streams to all or selected groups of users as

$$\boldsymbol{P}_c = \frac{1}{\beta_c} \sum_{k \in \mathcal{S}_\mathcal{K}} g_k \boldsymbol{V}_k, \tag{6.6}$$

where the matrix $\boldsymbol{V}_k$ is the right singular matrix corresponding to the first $l_c = c_k$ singular values of the channel matrix $\boldsymbol{H}_k$, $\mathcal{S}_\mathcal{K}$ is the set of users $\mathcal{K}$ that decode the common message, g_k is a power normalization according to the effective channel gain, and β_c is a normalization constant to fulfill the transmit power constraint. All users will first decode the common streams $\boldsymbol{s}_c$ by treating the interference from the private streams as noise. Thus, the achievable rate for the common streams at user k can be expressed as

$$R_k^c = \log_2 \det\left(\boldsymbol{I}_{l_c} + \boldsymbol{P}_c^{\mathrm{H}} \boldsymbol{H}_k^{\mathrm{H}} (\boldsymbol{G}_k^c)^{-1} \boldsymbol{H}_k \boldsymbol{P}_c\right), \tag{6.7}$$

where the term $\boldsymbol{G}_k^c = \sigma_k^2 \boldsymbol{I}_{M_{R_k}} + \sum_{i=1}^K \boldsymbol{H}_k \boldsymbol{P}_i \boldsymbol{P}_i^{\mathrm{H}} \boldsymbol{H}_k^{\mathrm{H}}$ denotes the noise plus interference covariance matrices. To ensure that all K users successfully decode the common message, the resulting achievable common rate R^c should not exceed $\min\left(R_i^c\right)_{i=1}^K$.

The private streams $\boldsymbol{s}_k$ are decoded independently by the corresponding users after performing SIC by subtracting the successfully decoded common signal $\boldsymbol{H}_k \boldsymbol{P}_c \boldsymbol{s}_c$ from the received signal $\boldsymbol{y}_k$. The private precoding matrices $\boldsymbol{P}_k$ can be designed based on the FlexCoBF scheme proposed in [SRH13], which provides freedom in the choice of the linear transmit and receive beamforming strategies and can be applied in both under- and overloaded MU-MIMO systems. Thus, the achievable rates for the private streams can be expressed as

$$R_k^p = \log_2 \det\left(\boldsymbol{I}_{l_k} + \boldsymbol{P}_k^{\mathrm{H}} \tilde{\boldsymbol{H}}_k^{\mathrm{H}} (\boldsymbol{G}_k^p)^{-1} \tilde{\boldsymbol{H}}_k \boldsymbol{P}_k\right), \tag{6.8}$$

where the term $\boldsymbol{G}_k^p = \sigma_k^2 \boldsymbol{I}_{M_{R_k}} + \sum_{i=1, i \neq k}^K \tilde{\boldsymbol{H}}_k \boldsymbol{P}_i \boldsymbol{P}_i^{\mathrm{H}} \tilde{\boldsymbol{H}}_k^{\mathrm{H}}$ corresponds to the noise plus interference from the private streams of the users other than k, and $\tilde{\boldsymbol{H}}_k = \boldsymbol{W}_k^{\mathrm{H}} \boldsymbol{H}_k$ is the effective channel matrix, where $\boldsymbol{W}_k$ is the decoding matrix of user k. The resulting rate achieved at user k is written as $R_k = R_k^p + C_k$, where C_k is a portion of the common rate assigned to the kth user, such that $\sum_{k=1}^K C_k = R^c$. The precoding schemes presented above assume that the number of common streams l_c does not exceed $\min\left(M_T, M_{R_i}\right)_{i=1}^K$. The total number of private streams satisfies $\sum_{k=1}^K l_k \leq M_T$. Therefore, the described schemes are applicable in a wide range of scenarios, including cases where the total number of streams $l > M_T$.

6.3.2 Special case: ML-GSVD-based transmission

Next, let us introduce the precoding and decoding schemes for the scenario where the total number of streams transmitted in parallel satisfies $l = l_c + \sum_{k=1}^{K} l_k \leq M_T$. Based on the ML-GSVD of the channel matrices $\boldsymbol{H}_k$, the overall precoding matrix $\boldsymbol{P}$ can be defined as

$$\boldsymbol{P} = \frac{1}{\beta} \boldsymbol{A}^{\mathrm{H}-}, \tag{6.9}$$

where β is a normalization coefficient ensuring the total power constraint. When considering a scenario with both common and private channels, (Case 2 of the ML-GSVD), the matrix $\boldsymbol{A}$ is $M_T \times M_T$ and is invertible. The decoding matrix of the kth user is defined as $\boldsymbol{W}_k = \boldsymbol{B}_k$. Consequently, the received data at the kth user in (6.5) can be rewritten as

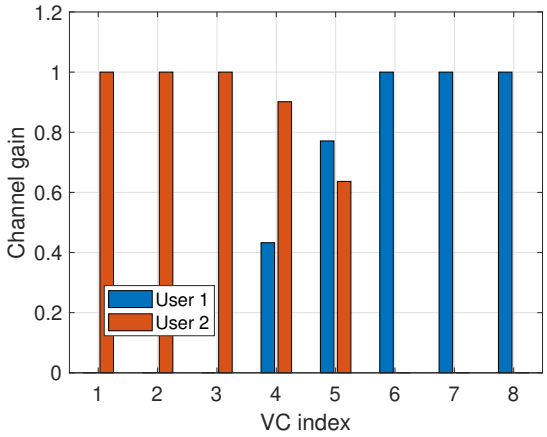
$$\boldsymbol{W}_k^{\mathrm{H}} \boldsymbol{y}_k = \boldsymbol{W}_k^{\mathrm{H}} \boldsymbol{H}_k \boldsymbol{P} \boldsymbol{s} + \boldsymbol{W}_k^{\mathrm{H}} \boldsymbol{n}_k \tag{6.10}$$

$$= \boldsymbol{C}_k \boldsymbol{s} + \tilde{\boldsymbol{n}}_k, \tag{6.11}$$

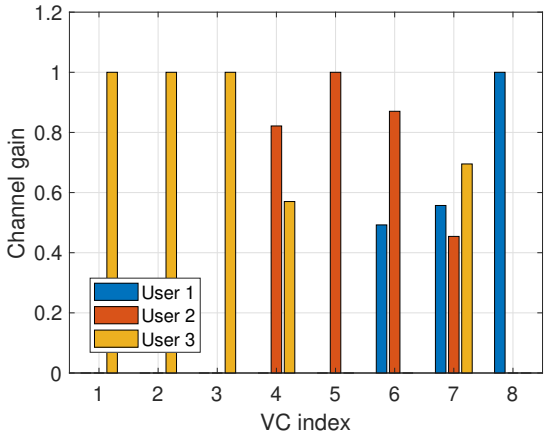
where $\tilde{\boldsymbol{n}}_k = \boldsymbol{B}_k^{\mathrm{H}} \boldsymbol{n}_k$, and $\boldsymbol{C}_k$ is a diagonal matrix as in (3.2) and (3.9). As it can be seen from (6.10), after applying the ML-GSVD, the MIMO channels $\boldsymbol{H}_k$ of K users can be considered as independent parallel virtual channels. Then, common streams can be transmitted on the common VCs (common ML-GSVD subspace) and decoded by the corresponding users. Private VCs (private ML-GSVD subspaces) are used to transmit the private streams that are detected individually by the intended users. It should be noted that by utilizing the ML-GSVD or GSVD for the transmission of private and common streams, we enable the joint detection of private and common streams without performing SIC at receivers, which reduces the complexity of the system and avoids error propagation, especially in multi-layer RSMA systems where several SIC steps have to be performed. Let us denote the entries of the diagonal matrix $\boldsymbol{\Sigma}_{c_k}$ in (3.9) as $\lambda_{k,n}$, $n \in \{1, \ldots, c_k\}$. Then since the common VC gains are equal to $\lambda_{k,n}$, $n \in \{1, \ldots, c_k\}$, the achievable rate of the nth common stream for the kth user is given by

$$R_{k,n}^{c} = \log_2 \left(1 + \frac{P_{c,n} \lambda_{k,n}^2}{\beta^2 \sigma_k^2} \right), \tag{6.12}$$

where $P_{c,n}$ is the portion of the total transmit power assigned to the common streams. As in (6.7), the resulting common rate R_n^c for the nth stream should not exceed $\min \left(R_{i,n}^{c} \right)_{i=1}^{K}$. The private channels have a unit channel gain. Thus, the individual



(a) $K = 2$. Channels 4 and 5 share a common subspace. One realization of the channel.



(b) $K = 3$. Channels 4, 6, and 7 share a common subspace. One realization of the channel.

Figure 6.3. Virtual channel gains of the users.

information rate of user k for the private stream m, $m \in \{1, \ldots, p_k\}$ can be expressed as

$$R_{k,m}^{p} = \log_2\left(1 + \frac{P_{p,m}}{\beta^2 \sigma_k^2}\right), \tag{6.13}$$

where $P_{p,m}$ is the portion of the total transmit power assigned to the mth private stream. The resulting rate achieved at user k is written as $R_k = \sum_{j=1}^{p_k} R_{k,j}^{p} + \sum_{i=1}^{c_k} C_{k,i}$, where $C_{k,n}$, $n \in \{1, \ldots, c_k\}$, is the portion of the common rate R_n^c assigned to the kth user, such that $\sum_{k=1}^{K} C_{k,n} = R_n^c$. We can choose the $C_{k,n}$s to be proportional to $\lambda_{k,n}$, i.e, the individual VC gains.

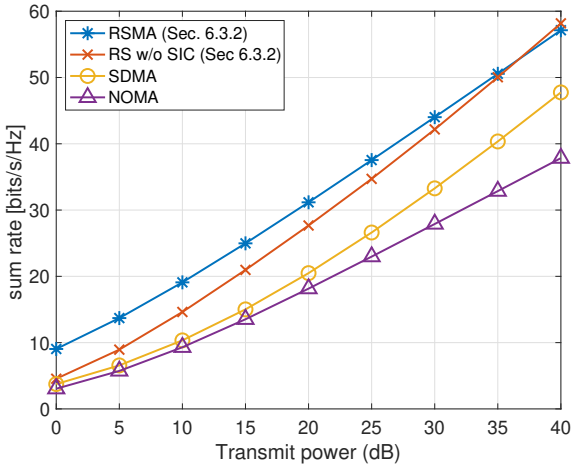


Figure 6.4. Sum Rates. $K = 2$. Results are averaged over 1000 Monte Carlo trials.

6.4 Simulation results

In this section, we present some numerical results to evaluate the performance of the proposed rate splitting schemes. The first simulation illustrates the private and common subspaces of the channel matrices provided by the GSVD and the ML-GSVD. For the simulations, we consider an uncorrelated Rayleigh fading MIMO channel model. Figure 6.3(a) shows the virtual channel gains obtained via the GSVD for an overloaded MIMO downlink system with $M_T = 8$ transmit antennas and two users equipped with $M_{R_k} = 5$ antennas each. As it can be observed, the two users have three private channels (with unit channel gain) and two common channels which can be used to transmit the common message. Since User 2 experiences a higher common channel gain on the VC 4, it can be assigned a larger portion of the common rate on that stream. Figure 6.3(b) demonstrates the virtual channel gains obtained with the ML-GSVD of the channels $\boldsymbol{H}_k$ for an asymmetrical scenario where the BS is equipped with $M_T = 8$ antennas, and three users are equipped with $M_{R_1} = 3$, $M_{R_2} = 4$, and $M_{R_3} = 5$ antennas, respectively. The ML-GSVD overcomes the limitation of the GSVD to two matrices and allows a joint decomposition of $K \geq 2$ users' channels. As it can be seen from Figure 6.3(b), in such a case the ML-GSVD distinguishes 1, 1, and 3 private VCs for the users 1, 2, and 3, respectively. Moreover, there are two common channels for a group of two users ($\mathcal{S}_{23}$ and $\mathcal{S}_{12}$) and one common channel for all three users ($\mathcal{S}_{123}$).

In the next simulation experiments, we assess the performance of the proposed precoding schemes. First, we consider a two-user downlink MIMO scenario with one

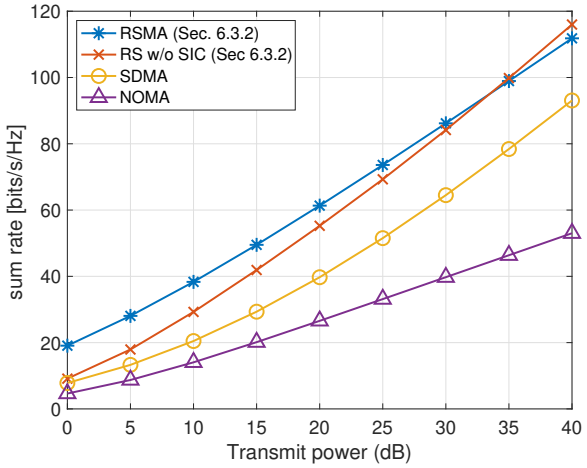


Figure 6.5. Sum Rates. $K = 3$. Results are averaged over 1000 Monte Carlo trials.

BS equipped with $M_T = 5$ transmit antennas and two receivers both equipped with $M_{R_k} = 3$ antennas. The total number of streams is equal to M_T. This scenario corresponds to the GSVD providing both common and private subspaces. Figure 6.4 shows the sum rate performances for the precoding schemes proposed in Sections 6.3.1 and 6.3.2 and their comparison to NOMA (data of the first user is transmitted on the private streams, and data of the second user is transmitted on the common streams [MDC+22]) and SDMA (zero power is assigned to the common streams). We perform the power allocation based on an exhaustive search, maximizing the sum rate and assuming that the users have equal weights.

Figure 6.5 shows the achievable sum rates for a scenario with $K = 3$ users, where the BS and every user is equipped with $M_T = 10$ and $M_{R_k} = 4$ antennas, respectively. In this case, the common streams can be sent to all three or to a group of two users (for $K > 2$ we employ the ML-GSVD to define the common and private subspaces as well as the number of streams). Such a scenario corresponds to the generalized RS where the number of SIC layers L can increase up to 4 for $K = 3$ users ($L = 2^{K-1}$, see [MDC+22] for details). The scheme described in Section 6.3.2 allows transmitting and receiving the common and private steams without SIC. As shown in Figures 6.4 and 6.5, it performs better than the RSMA scheme for the higher SNRs. In general, both proposed rate splitting techniques outperform the SDMA and NOMA for the presented two and three users scenarios.

6.5 Conclusions

In this chapter of the thesis, we have proposed a precoder design for MIMO broadcast systems with rate splitting at the transmitter. We have shown how the GSVD and the ML-GSVD can be employed to define the number of common and private streams as well as to adjust the message split. The proposed scheme applies to both underloaded and overloaded communication systems and supports the transmission of multiple common and private streams. Moreover, for the cases where the total number of streams does not exceed the number of transmit antennas, we have introduced transmit precoding and receive combining designs based on the ML-GSVD, which allow simultaneous transmission of common and private streams but do not require SIC at the receivers. The use of the ML-GSVD overcomes the two-user limitation of the GSVD, allowing its application to systems with more than two users where the common streams can also be transmitted to selected groups of users (multi-layer hierarchical or generalized RS). Simulation results have shown that both proposed rate splitting schemes outperform SDMA and NOMA.

Part III.

Exploring the Block-Term Decomposition

Rank-$(L_r, L_r, 1)$ Block-Term Decomposition

7

The block-term decompositions (BTD) represent tensors as a linear combination of low multilinear rank terms and can be explicitly related to the canonical polyadic decomposition (CPD). In this chapter of the thesis, we introduce the SECSI-BTD framework, which exploits the connection between these two decompositions to estimate the block-terms of the rank-$(L_r, L_r, 1)$ BTD. The proposed SECSI-BTD algorithm includes the initial calculation of the factor estimates using the SEmi-algebraic framework for approximate Canonical polyadic decompositions via SImultaneous Matrix Diagonalizations (SECSI), followed by clustering and refinement procedures that return the appropriate rank-$(L_r, L_r, 1)$ BTD terms. Moreover, we introduce a new approach to estimate the multilinear rank structure of the tensor based on the higher-order singular value decomposition (HOSVD) and k-means clustering. Since the proposed SECSI-BTD algorithm does not require a known rank structure but can still take advantage of the known ranks when available, it is more flexible than the existing techniques in the literature. Additionally, SECSI-BTD does not require multiple initializations, and the simulation results show that it provides more accurate results and a better convergence behavior for an extensive range of SNRs [KSP+23].

7.1 Introduction

Owing to multidimensionality-related benefits, tensor-based techniques have become a primary tool for many signal processing applications. Tensor decompositions are often identifiable under mild uniqueness conditions [DD17], and compared to classical matrix-based approaches, they allow preserving the data's original structure when it is inherently multidimensional. This enables modeling a signal across multiple domains and facilitates the interpretation of results. Additionally, a variety of tensor factorizations allow choosing a technique that best suits a given task or data.

Probably the most prominent and extensively used tensor decompositions are the higher-order singular value decomposition (HOSVD) [DDV00a; Tuc66] and the canonical polyadic decomposition (CPD), sometimes also referred to as CANDECOMP

(canonical decomposition) or PARAFAC (parallel factors) [CC70; Har70; KB09]. They have found their applications in a wide range of fields, including statistics, communications, localization, biomedical signal processing, source separation, and many others [KB09; SDF+17]. Furthermore, both decompositions have several variations and extensions, and the block-term decomposition (BTD) can be considered as one of them. Depending on the ranks of sub-blocks in the BTD, several types of this decomposition have been distinguished in the literature, for example, decomposition in rank-$(L, L, 1)$ terms, in rank-(L, M, N) terms, or in rank-$(L, M, \cdot)$ terms [DeL08a; DN08]. In this study, our focus falls on the block-term decomposition in rank-$(L_r, L_r, 1)$ terms, which has gained increasing attention from researchers in the last decade.

Whereas CPD techniques are quite well understood in the literature (there are elegant uniqueness theorems, stable and powerful tools for their computation and model order estimation), block-term decompositions are still a subject of active research since they pose a more complex problem due to inherent ambiguities and weaker uniqueness properties. Interestingly, exactly these properties render the block-term decompositions attractive for some applications since their uniqueness conditions are less strict, and they can be applied under more general circumstances than the CPD. However, state-of-the-art BTD frameworks as, for example, Tensorlab [VDS+16], do not always provide a stable decomposition performance, implying that they are prone to erroneous estimation of the factors for a great range of possible model parameters (these can be traced back to suboptimal initialization of the BTD-algorithm). An additional challenge that arises before computing the BTD is the estimation of the number of BTD terms and the multilinear ranks. For instance, in Tensorlab solutions, the number of rank-$(L_r, L_r, 1)$ terms as well as the L_rs have to be known beforehand, which might not be the case for some applications.

Recently, block-term decompositions of noise-corrupted tensors have received a lot of attention in different research areas. The authors in [DeL08b; DeL08a; DN08; ND09; DeL11; SVBD13] introduce the definitions, a link to the CPD, and the uniqueness conditions for different types of block-term decompositions as well as optimization-based and algebraic algorithms to estimate the block-term factors assuming that the rank structure is known beforehand. On the other hand, the authors in [RKG21b; RKG21a; RGK21; GRK22; GOF+20] present the hierarchical iteratively reweighted least squares and the alternating group lasso algorithms that estimate both the ranks and the factors of rank-$(L_r, L_r, 1)$ BTDs. However, they do not exploit prior knowledge of ranks when available and require multiple initializations to ensure convergence. Another recent paper on the multilinear rank decomposition investigates the conditions under which the decomposition in rank-$(L_r, L_r, 1)$ terms is unique and can be computed via

an eigenvalue decomposition [DD20]. The BTD algorithm based on the group sparsity property of the loading matrices introduced in [HAK+17b] also allows performing the decomposition and the model order estimation. Still, it is limited to a rank-$(L, L, 1)$ BTD with equal multilinear ranks.

Other research directions, that further exploit the BTD, include a variety of different applications and the extension to coupled decompositions. For instance, the uniqueness conditions and the algorithms for coupled CPD and BTD in rank-$(L_{r,n}, L_{r,n}, 1)$ terms are discussed in [SD15; SDD15], and structured data fusion by means of coupled tensor decompositions is presented in [SVBD15]. The authors in [GLD+16] use a coupled block simultaneous generalized Schur decomposition to calculate the coupled rank-$(L_m, L_n, \cdot)$ BTD. In [SKKH22] and in Chapter 8 of this thesis, we propose an algorithm to calculate the coupled rank-$(L_r, L_r, 1)$ BTD of multiple tensors with a common mode which often occurs in biomedical data applications [CRHH21]. Moreover, several authors have presented many other practical applications of the BTD, which include communications signal processing [DB08], image and graph analysis [GPP20], radar systems [YGL+19], and biomedical data processing [RHMZ15; OZ19; RAZ16; Zar17; HCS+14]. The nonnegative rank-$(L_r, L_r, 1)$ and coupled rank-$(L_r, L_r, 1)$ decompositions with application to hyperspectral imagery and cartography are investigated in [ZFHW19; XQZT19; ZFW+20]. The authors in [GP20] propose an algorithm to compute the rank-(L, M, N) BTD of large streaming tensor datasets. Another way of looking at the CPD and BTD is discussed in [TPC19; TPC21]. The authors view these decompositions as a special case of the Tucker decomposition and introduce the Krylov-Levenberg-Marquardt algorithm to compute it. Compared to the rank-$(L_r, L_r, 1)$ BTD studied in this thesis, the BTD in [TPC19] is considered as a sum of Tucker tensors with a block-diagonal core tensor which in turn requires prior knowledge of the block structure (multilinear ranks).

Main contributions. This chapter of the thesis proposes a new approach to calculate the rank-$(L_r, L_r, 1)$ block-term decomposition of noise-corrupted data based on the SEmi-algebraic framework for approximate Canonical polyadic decompositions via SImultaneous Matrix Diagonalizations (SECSI) [RH13], which shows enhanced numerical stability even for low SNR scenarios. In contrast to the schemes in [RKG21b; RKG21a; RGK21; GOF+20], our algorithm does not require multiple initializations to ensure convergence. Moreover, compared to the algorithm in [VDS+16], the proposed approach can perform the decomposition even with an unknown rank structure. To this end, we introduce an extension of the LaRGE (LineAr Regression of Global Eigenvalues) [KDHH21] scheme, originally designed for the estimation of the CPD rank of a noise-corrupted low-rank tensor, to estimate the number of L_r terms. We also show

how the L_rs can be estimated through k-means clustering. This chapter explores a general rank-$(L_r, L_r, 1)$ decomposition of a single three-dimensional tensor with possibly unknown block- and multilinear ranks. We provide a detailed description of the proposed approach, including the CPD to BTD transformation, the estimation of ranks, and the calculation of the BTD factors. Moreover, we conduct extensive and thorough synthetic data simulations to validate the proposed algorithm and compare it to the schemes from the literature. We refer to the proposed rank-$(L_r, L_r, 1)$ BTD framework as SECSI-BTD. It should be mentioned that it does not require a known rank structure but can still exploit prior knowledge of the model order when available. Altogether, the SECSI-BTD framework to compute an approximate BTD from noise-corrupted measurements is composed of three main blocks (Figure 7.1):

- The model order estimation block based on the LineAr Regression of Global Eigenvalues (LaRGE) scheme to estimate the number of block-terms and the sum of multilinear ranks (can be skipped if the ranks are known beforehand);
- The computation of the initial estimates using the BTD extension of the SECSI framework and clustering, which includes the calculation of the initial rank-$(L_r, L_r, 1)$ decomposition factors via simultaneous matrix diagonalizations and clustering via the k-means algorithm;
- The refinement procedures that bring the initial estimates to the BTD form and return the final estimates of rank-$(L_r, L_r, 1)$ terms by employing ALS or NLS iterations [DN08; SVBD13].

For notational simplicity, hereafter, by writing "BTD" or "rank-$(L_r, L_r, 1)$ decomposition" we refer to the block-term decomposition in rank-$(L_r, L_r, 1)$ terms. Moreover, we use ΣL_r to denote $\sum_{r=1}^{R} L_r$.

The rest of the chapter is organized as follows. Section 7.2 provides some preliminaries on the BTD. In Section 7.3, we present an algorithm to estimate the model order of the rank-$(L_r, L_r, 1)$ BTD. Then, we introduce a simultaneous matrix diagonalization (SMD)-based technique to calculate the block-term factor matrices in Section 7.4. Section 7.5 presents the refinement procedures to construct the final estimates. The numerical results are shown in Section 7.6, and Section 7.7 is devoted to the conclusions.

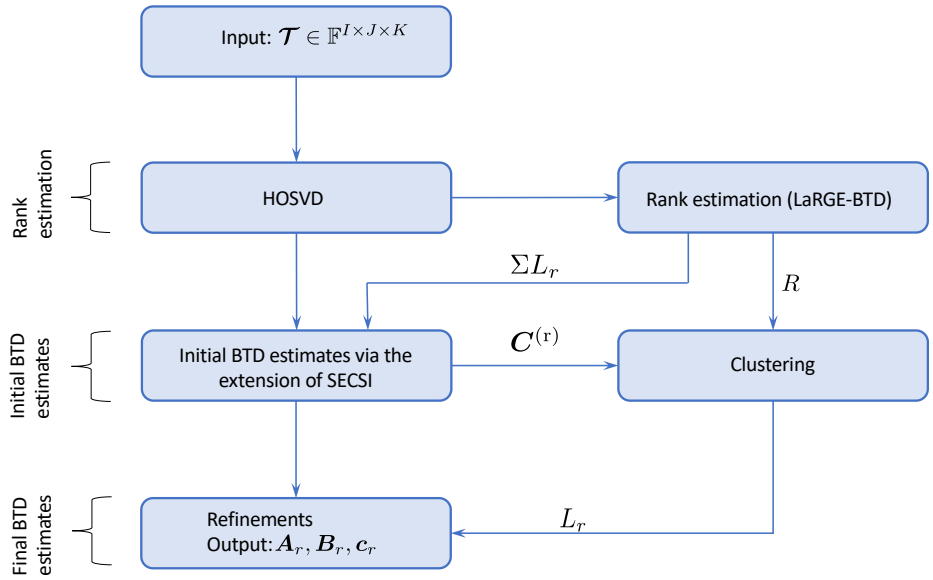


Figure 7.1. Block-diagram of the SECSI-BTD framework.

7.2 Preliminaries: BTD vs. CPD

In this study, we exploit the connection between the CPD and the BTD to estimate the block-terms of a rank-$(L_r, L_r, 1)$ decomposition. Even though their uniqueness properties differ, the BTD and CPD have much in common. The BTD can be seen as a special case of the CPD with collinearity in one of the factors, see Figure 7.2. Then, the rank-$(L_r, L_r, 1)$ BTD of a three-dimensional noise-corrupted tensor $\mathcal{T} \in \mathbb{F}^{I \times J \times K}$ can either be written in a BTD form [VDS+16]

$$\mathcal{T} = \sum_{r=1}^{R} \left(\boldsymbol{A}_r \cdot \boldsymbol{B}_r^{\mathrm{T}} \right) \circ \boldsymbol{c}_r + \mathcal{N}, \tag{7.1}$$

or in a CPD form

$$\mathcal{T} = \sum_{r=1}^{R} \sum_{\ell=1}^{L_r} \boldsymbol{A}_{r,(:,\ell)} \circ \boldsymbol{B}_{r,(:,\ell)} \circ \boldsymbol{c}_r + \mathcal{N} \tag{7.2}$$

$$= \mathcal{I}_{3,\Sigma L_r} \times_1 \boldsymbol{A} \times_2 \boldsymbol{B} \times_3 \boldsymbol{C}^{(\mathrm{r})} + \mathcal{N}, \tag{7.3}$$

where R and L_r are the number of block-terms and the multilinear ranks, respectively. Moreover, $\boldsymbol{A}_r \in \mathbb{F}^{I \times L_r}$ and $\boldsymbol{B}_r \in \mathbb{F}^{J \times L_r}$ are the r-th submatrices of $\boldsymbol{A}$ and $\boldsymbol{B}$ so that

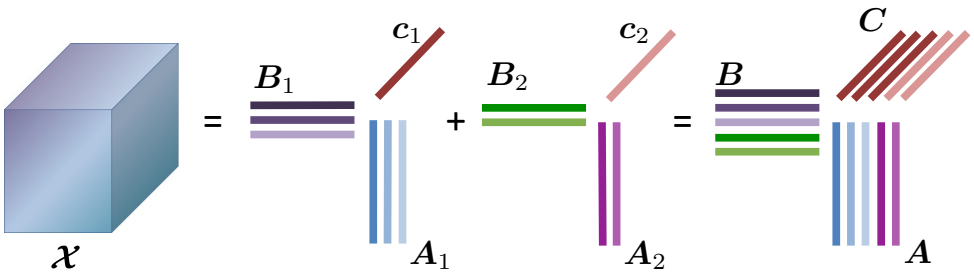


Figure 7.2. Rank-$(L_r, L_r, 1)$ BTD with $L_1 = 3$, $L_2 = 2$, and $R = 2$ as a CPD with repeated columns in the 3-mode (matrix $\boldsymbol{C}$).

$\boldsymbol{A} = [\boldsymbol{A}_1, \boldsymbol{A}_2, ..., \boldsymbol{A}_R] \in \mathbb{F}^{I \times \Sigma L_r}$ and $\boldsymbol{B} = [\boldsymbol{B}_1, \boldsymbol{B}_2, ..., \boldsymbol{B}_R] \in \mathbb{F}^{J \times \Sigma L_r}$. The vectors $\boldsymbol{c}_r \in \mathbb{F}^{K \times 1}$, $r = \{1, ..., R\}$ are stacked in the matrix $\boldsymbol{C}$ so that $\boldsymbol{C} = [\boldsymbol{c}_1, \boldsymbol{c}_2, ..., \boldsymbol{c}_R] \in \mathbb{F}^{K \times R}$. The matrix

$$\boldsymbol{C}^{(\mathrm{r})} = [\boldsymbol{c}_1 \cdot \mathbf{1}_{L_1}^{\mathrm{T}}, \boldsymbol{c}_2 \cdot \mathbf{1}_{L_2}^{\mathrm{T}}, \cdots, \boldsymbol{c}_R \cdot \mathbf{1}_{L_R}^{\mathrm{T}}] \in \mathbb{F}^{K \times \Sigma L_r} \tag{7.4}$$

has repeated (or linearly dependent) columns. Then, the BTD in (7.1) and (7.2) can be viewed as CPD with linear dependencies in the third factor matrix or, similarly, the CPD can be considered as a BTD with all L_rs being equal to one. With repeated or colinear columns in $\boldsymbol{C}^{(\mathrm{r})}$, Kruskal's condition [SS07] (see Section 2.2.4) for uniqueness is evidently not satisfied. On the other hand, the authors in [DeL08a; DD20] point out that the assumption that some L_rs are different from 1 is more realistic for some applications, and multilinear rank decompositions still possess attractive uniqueness properties (we list some uniqueness conditions for the rank-$(L_r, L_r, 1)$ BTD in Appendix A.4). At the same time, while the CPD is said to be essentially unique up to an arbitrary permutation and scaling of its rank-one terms, an additional matrix product ambiguity occurs when considering the essential uniqueness of the multilinear-rank terms in the BTD

$$\boldsymbol{A}_r \boldsymbol{B}_r^{\mathrm{T}} = (\boldsymbol{A}_r \boldsymbol{H}_r) \left(\boldsymbol{H}_r^{-1} \boldsymbol{B}_r^{\mathrm{T}} \right) = \boldsymbol{A}'_r \boldsymbol{B}'^{\mathrm{T}}_r, \tag{7.5}$$

where $\boldsymbol{H}_r$ is an arbitrary nonsingular matrix. Since this submatrix product ambiguity is hard to resolve, the BTD is often written in the following form

$$\mathcal{T} = \sum_{r=1}^{R} \boldsymbol{E}_r \circ \boldsymbol{c}_r + \mathcal{N}, \tag{7.6}$$

where $\boldsymbol{A}_r$ and $\boldsymbol{B}_r$ are merged into $\boldsymbol{E}_r \in \mathbb{F}^{I \times J}$ such that $\boldsymbol{E}_r = \boldsymbol{A}_r \boldsymbol{B}_r^{\mathrm{T}}$ [DeL08a].

In the next sections, we will introduce an approach to determine the number of terms R and the multilinear ranks L_r using the HOSVD and clustering, followed by an estimation of the BTD factors via SECSI [RH13].

7.3 HOSVD-based model order estimation

In the most common applications of the BTD, such as electrocardiography (ECG), or electroencephalography (EEG) and magnetoencephalography (MEG) data processing or source separation, the rank structure is usually not available beforehand. Some papers determine the ranks by trial and error, which is very straightforward but not efficient [CRHH21; dVD+07]. The authors in [RKG21b; RKG21a; RGK21; GOF+20] propose approaches that compute the factors and ranks of the BTD jointly in an iterative way which is computationally not effective in cases when, for example, only the rank structure itself is a subject of interest. In this study, we show how to estimate the model order of noise-corrupted BTD tensors by separating the signal and noise subspaces based on an extension of the linear regression of global eigenvalues (LaRGE) approach in [KDHH21].

To this end, let us consider a noise-corrupted three-dimensional measurement tensor $\boldsymbol{\mathcal{T}}$ given by $\boldsymbol{\mathcal{T}} = \boldsymbol{\mathcal{T}}_0 + \boldsymbol{\mathcal{N}} \in \mathbb{F}^{I\times J\times K}$, where $\boldsymbol{\mathcal{T}}_0 = \sum_{r=1}^{R} \left(\boldsymbol{A}_r \cdot \boldsymbol{B}_r^{\mathrm{T}}\right) \circ \boldsymbol{c}_r \in \mathbb{F}^{I\times J\times K}$ is the noiseless BTD structured data, and $\boldsymbol{\mathcal{N}} \in \mathbb{F}^{I\times J\times K}$ is an additive noise tensor. Whereas the rank of the first two modes of this rank-$(L_r, L_r, 1)$ block-term decomposition is assumed to be equal to ΣL_r, the rank in the 3-mode is assumed to be R. Then, the HOSVD of $\boldsymbol{\mathcal{T}}$ is given by

$$\boldsymbol{\mathcal{T}} = \boldsymbol{\mathcal{S}} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \times_3 \boldsymbol{U}_3, \tag{7.7}$$

where $\boldsymbol{\mathcal{S}} \in \mathbb{F}^{I\times J\times K}$ is the core tensor, and $\boldsymbol{U}_1 \in \mathbb{F}^{I\times I}$, $\boldsymbol{U}_2 \in \mathbb{F}^{J\times J}$, and $\boldsymbol{U}_3 \in \mathbb{F}^{K\times K}$ contain the left singular vectors of the n-mode unfoldings of $\boldsymbol{\mathcal{X}}$ computed from $\left[\boldsymbol{\mathcal{T}}\right]_{(n)} = \boldsymbol{U}_n \boldsymbol{\Sigma}_n \boldsymbol{V}_n^{\mathrm{H}}$, $n \in \{1,2,3\}$, where $\boldsymbol{\Sigma}_n$ contains the n-mode singular values $\sigma_i^{(n)}$ on its main diagonal. The n-mode singular values $\sigma_i^{(n)}$ are related to the n-mode eigenvalues $\lambda_i^{(n)}$ of $\left[\boldsymbol{\mathcal{T}}\right]_{(n)}\left[\boldsymbol{\mathcal{T}}\right]_{(n)}^{\mathrm{H}}$ through $\lambda_i^{(n)} = \left(\sigma_i^{(n)}\right)^2$. Moreover, the eigenvalues can be computed as $\mathrm{diag}\left(\lambda_1^{(n)}, \ldots, \lambda_Q^{(n)}\right) = \left[\boldsymbol{\mathcal{S}}\right]_{(n)} \cdot \left[\boldsymbol{\mathcal{S}}\right]_{(n)}^{\mathrm{H}}$, where $Q \in \{I, J, K\}$ and $\left[\boldsymbol{\mathcal{S}}\right]_{(n)}$ is the n-mode unfolding of the core tensor $\boldsymbol{\mathcal{S}}$ in (7.7).

The main concept of LaRGE is based on the fact that (asymptotically) the noise eigenvalues have an exponential decay and, therefore, unlike the signal eigenvalues, can be approximated by a straight line on a logarithmic scale. Hence, the point where the linear regression fails will indicate the detection of the smallest signal eigenvalue and, accordingly, the rank. In the original LaRGE algorithm [KDHH21] for the estimation of the CPD rank, the authors perform the linear regression on the so-called *global eigenvalues* [CHRDG07]. The ith global eigenvalue is equal to the product of the ith n-mode eigenvalues, where $i = 1, \ldots, M$, $M = \min\left(I, J, K\right)$. The assumption that all factors in the CPD model have the same rank (column dimension) allows taking into account

the eigenvalues from all modes and leads to more reliable estimates. On the other hand, as it follows from the definition of the multilinear rank decompositions, not all the n-mode ranks have to be equal. If this asymmetric rank structure is not taken into account, the global eigenvalues might be heavily affected by the rank deficiencies in the third mode. Therefore, we split the BTD rank structure estimation procedure into two runs to accommodate this. During the first run, the ΣL_r are estimated based on the *semi-global eigenvalues* from the 1-mode and the 2-mode

$$\tilde{\lambda}_i^{(1,2)} = \left(\sigma_i^{(1)}\right)^2 \cdot \left(\sigma_i^{(2)}\right)^2, \tag{7.8}$$

and during the second run, only the 3-mode is used

$$\tilde{\lambda}_i^{(3)} = \left(\sigma_i^{(3)}\right)^2. \tag{7.9}$$

The linear regression scheme can be applied to $\lambda_i^{(1,2)} = \ln \tilde{\lambda}_i^{(1,2)}$ and $\lambda_i^{(3)} = \ln \tilde{\lambda}_i^{(3)}$ separately to estimate ΣL_r and R as follows. Starting from the smallest eigenvalue λ_M, find a prediction $\hat{\lambda}_i$ of the next eigenvalue on a logarithmic scale[1] subject to

$$\underset{a_1,a_2}{\arg\min} \left(\sum_{i=M}^{M-k} \left(\hat{\lambda}_i - \lambda_i\right)^2 \right), \tag{7.10}$$

where

$$\hat{\lambda}_i = a_1 i + a_2 \tag{7.11}$$

is the smallest predicted eigenvalue, $k = \{1, \ldots, M-1\}$ is the step index, and M is equal to $\min\left(I, J\right)$ when estimating ΣL_r and to K when estimating R. For each step k calculate the relative prediction error as follows

$$\delta_{M-k} = \frac{\lambda_{M-k} - \hat{\lambda}_{M-k}}{\left|\hat{\lambda}_{M-k}\right|} = \frac{\Delta_{M-k}}{\left|\hat{\lambda}_{M-k}\right|}. \tag{7.12}$$

Next, calculate the standard deviation of the approximation error as

$$\sigma_{M-k} = \sqrt{\frac{1}{k} \sum_{i=M}^{M-k} \left(\Delta_i - \frac{1}{k} \sum_{i=M}^{M-k} \Delta_i \right)^2}, \tag{7.13}$$

[1] Since the further rank estimation steps are the same for both ΣL_r and R, for notational simplicity, we skip the superscripts (1,2) and (3) in λ_i (λ_i corresponds to $\lambda_i^{(1,2)}$ when estimating ΣL_r, and to $\lambda_i^{(3)}$ when estimating R) and M.

Algorithm 2 LaRGE-BTD: ΣL_r (case 1) and R (case 2)

Require: 3-way tensor $\mathcal{T} \in \mathbb{F}^{I \times J \times K}$

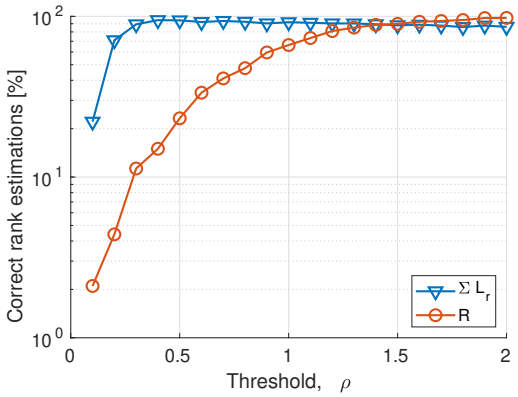
$\sigma_i^{(n)} \leftarrow \text{HOSVD}(\mathcal{T})$
Case 1: $M = \min(I, J)$, Case 2: $M = K$
for $i = 1, \ldots, M$ **do**
    Case 1: $\tilde{\lambda}_i = \left(\sigma_i^{(1)}\right)^2 \cdot \left(\sigma_i^{(2)}\right)^2$; **Case 2:** $\tilde{\lambda}_i = \left(\sigma_i^{(3)}\right)^2$
    $\lambda_i = \ln \tilde{\lambda}_i$
end for
for $k = 1, \ldots, M-1$ **do**
    prediction $\hat{\lambda}_{M-k}$ according to (7.11)
    $\delta_{M-k} = \frac{\lambda_{M-k} - \hat{\lambda}_{M-k}}{|\hat{\lambda}_{M-k}|} = \frac{\Delta_{M-k}}{|\hat{\lambda}_{M-k}|}$
    $\sigma_{M-k} = \sqrt{\frac{1}{k}\sum_{i=M}^{M-k}\left(\Delta_i - \frac{1}{k}\sum_{i=M}^{M-k}\Delta_i\right)^2}$
    $\text{PESDR}_k = \frac{\delta_{M-k}}{\sigma_{M-k-1}}$
    if $(\text{PESDR}_{k-1} < \rho) \wedge (\text{PESDR}_k \geq \rho)$ **then**
        $\text{Rank} = M - k$
        break
    end if
end for

where the second quantity in parentheses denotes the mean value of the absolute prediction error Δ_i. Then, compute the ratio between the relative prediction error and the standard deviation of the approximation errors in the previous steps as

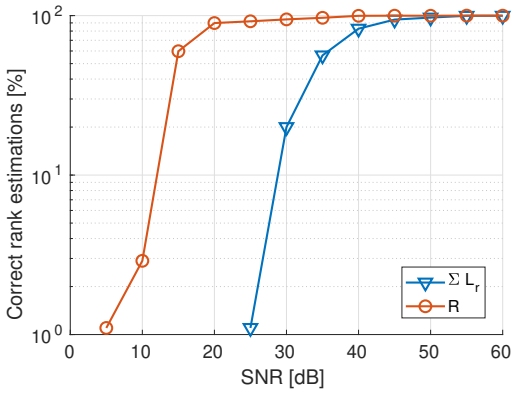
$$\text{PESDR}_k = \frac{\delta_{M-k}}{\sigma_{M-k-1}}. \tag{7.14}$$

The first signal eigenvalue is detected when the Prediction Error to Standard Deviation Ratio (PESDR) exceeds the predefined threshold ρ for the first time. The number of signal eigenvalues corresponds to an estimate of ΣL_r when regression is performed on the 1-mode and 2-mode eigenvalues, and to an estimate of R when the 3-mode eigenvalues are used. The estimate of ΣL_r is further employed for the computation of initial estimates via SECSI, while the estimate of R is used for the clustering of the BTD terms as depicted in Figure 7.2.

We refer to the proposed model order estimation algorithm (for the BTD) as LaRGE-BTD, whose main steps are summarized in Algorithm 2. Moreover, to avoid estimation errors appearing due to a relatively small difference between the rank and the smallest dimension of the tensor, LaRGE with a penalty function (LaRGE PF) can be employed [KDHH22]. The penalty function ensures that the value of σ_{M-k} exceeds a certain threshold ε, which allows reducing the outliers that may lead to wrong estimates. For more details on the LaRGE algorithm, we refer the reader to [KDHH21].



(a) Correct rank estimations vs. Threshold. SNR = 20 dB and SNR = 45 dB for R and L_r simulations, respectively.



(b) Correct rank estimations vs. SNR. Thresholds are $\rho_R = 1.5$ and $\rho_{\Sigma L_r} = 0.5$.

Figure 7.3. Performance of LaRGE-BTD. Tensor of size $40 \times 40 \times 40$ with $R = 3$, $L_r = [3,3,3]$. Results are averaged over 1000 Monte-Carlo trials.

The numerical results of the LaRGE-BTD estimation performance are demonstrated in Figure 7.3. For the simulations, 1000 complex-valued tensors were constructed according to the model in (7.1) with factor matrix entries drawn from a zero mean uncorrelated Gaussian distribution with variance σ_n^2. Accordingly, the SNR is defined as $1/\sigma_n^2$. Figure 7.3(a) depicts the percentage of correct rank estimates as a function of the threshold ρ. As it can be observed, different thresholds should be used for the estimation of the number of block-terms and the sum of the multilinear ranks. According to the simulation results, the recommended threshold ρ for R and ΣL_r are between $1.5-2$ and $0.4-0.5$, respectively. The second simulation in Figure 7.3(b) shows the performance of LaRGE-BTD as a function of the SNR ($\rho_R = 1.5$ and $\rho_{\Sigma L_r} = 0.5$). As it can be seen, the estimation of the number of block-terms R is more reliable. This can be explained by the fact that the difference between ΣL_r and the dimensions of a tensor is smaller than the difference between R and the dimensions of a tensor.

7.4 Extension of the SECSI framework for block-term decomposition

In contrast to previous [DeL06] simultaneous matrix diagonalization (SMD)-based approaches, the SECSI framework exploits the tensor structure of the CPD to construct not only one but the full set of possible SMDs. By solving all SMDs, multiple estimates of the factor matrices can be obtained, and strategies have been presented to choose the best estimate in a subsequent step [RH13].

In this chapter, we introduce an extension of SECSI that enables the computation of initial estimates of the rank-$(L_r, L_r, 1)$ block-term decomposition. The new extension includes the following major enhancements and contributions:

- An introduction of a new heuristic that reduces the number of SMDs to be computed, thereby decreasing the computational load.
- An estimation of the multilinear ranks L_r using clustering.
- A befitting partitioning of the columns of the estimated matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ into multilinear rank submatrices.
- A design of highly reliable initial BTD estimates that guarantees the convergence of the refinement procedures and does not require multiple initializations.

7.4.1 Brief overview of the SECSI framework

The SECSI framework [RH13] has originally been designed to decompose a tensor into the sum of rank-1 terms. Before presenting its extension to a multilinear rank decomposition, let us review its main steps.

Truncated HOSVD

Let $\boldsymbol{\mathcal{T}}_0 \in \mathbb{F}^{I\times J\times K}$ be a noise-free tensor with a given (or estimated) rank d whose CPD is given by

$$\boldsymbol{\mathcal{T}}_0 = \boldsymbol{\mathcal{I}}_{3,d} \times_1 \boldsymbol{F}_1 \times_2 \boldsymbol{F}_2 \times_3 \boldsymbol{F}_3, \tag{7.15}$$

where $\boldsymbol{F}_1$, $\boldsymbol{F}_2$, and $\boldsymbol{F}_3$ are the corresponding factor matrices.

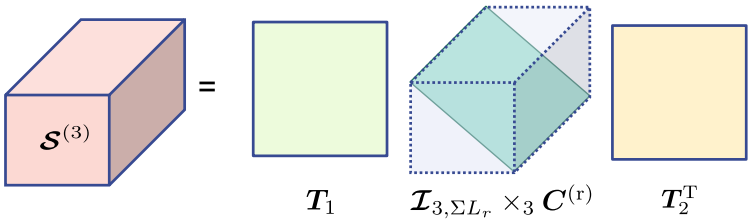


Figure 7.4. Simultaneous matrix diagonalization of the slices of the tensor $\mathcal{S}^{(3)}$.

The first step of SECSI is the computation of a truncated HOSVD[2] of $\mathcal{T}$ to generate a truncated core tensor $\mathcal{S}^{[\mathrm{s}]} \in \mathbb{F}^{d \times d \times d}$ and a set of truncated unitary matrices $\boldsymbol{U}_1^{[\mathrm{s}]} \in \mathbb{F}^{I \times d}$, $\boldsymbol{U}_2^{[\mathrm{s}]} \in \mathbb{F}^{J \times d}$, and $\boldsymbol{U}_3^{[\mathrm{s}]} \in \mathbb{F}^{K \times d}$ that span the column space of the corresponding n-mode unfolding of $\mathcal{T}_0$

$$\mathcal{T}_0 = \mathcal{S}^{[\mathrm{s}]} \times_1 \boldsymbol{U}_1^{[\mathrm{s}]} \times_2 \boldsymbol{U}_2^{[\mathrm{s}]} \times_3 \boldsymbol{U}_3^{[\mathrm{s}]}. \tag{7.16}$$

Only the first d singular values and their corresponding left singular vectors are chosen to take into account the given (or estimated) CPD rank. The columns of each of the unitary matrices span the same vector space as the CPD factor matrices in the same mode. Thus, there exist three non-singular $d \times d$ transform matrices[3] $\boldsymbol{T}_1$, $\boldsymbol{T}_2$, and $\boldsymbol{T}_3$ that diagonalize the core tensor $\mathcal{S}^{[\mathrm{s}]}$ and provide a connection between the two sets of matrices

$$\boldsymbol{F}_1 = \boldsymbol{U}_1^{[\mathrm{s}]} \cdot \boldsymbol{T}_1, \quad \boldsymbol{F}_2 = \boldsymbol{U}_2^{[\mathrm{s}]} \cdot \boldsymbol{T}_2, \quad \boldsymbol{F}_3 = \boldsymbol{U}_3^{[\mathrm{s}]} \cdot \boldsymbol{T}_3. \tag{7.17}$$

In the next step, these transformation matrices are estimated (in several different ways).

Simultaneous matrix diagonalization of the core tensor's slices

It has been shown in [RH13] that the n-mode slices of the core tensor $\mathcal{S}$ have a direct relation to the columns of the n-mode's factor matrix and the transformation matrices of the other two modes from equation (7.17). For example, if $\mathcal{S}^{(3)}_{(:,:,k)}$ is the k-th 3-mode

[2] If the HOSVD has been already calculated during the LaRGE-based rank estimation step (Section 7.3), then it can be further truncated in the first step of SECSI without a need to calculate it again. This will significantly reduce the computational cost.

[3] The assumption of the existence of the three non-singular transform matrices holds when the CPD is not rank-deficient in any of the three modes, i.e., all factor matrices have full column rank. If the CP decomposition is rank-deficient in one mode, the corresponding transform matrix and SMDs do not exist anymore. However, if the remaining two modes are not rank-deficient, we can still construct two SMDs (left-hand and right-hand sides) that do not contain the transform matrix corresponding to the rank-deficient mode. Therefore, the SECSI framework is applicable if at least two of the tensor modes are not rank-deficient.

slice of the modified core tensor $\boldsymbol{\mathcal{S}}^{(3)} = \boldsymbol{\mathcal{S}}^{[\mathrm{s}]} \times_3 \boldsymbol{U}_3^{[\mathrm{s}]} \in \mathbb{F}^{d \times d \times K}$, and $\hat{\boldsymbol{c}}_k$ is the k-th row of $\boldsymbol{F}_3$, then

$$\boldsymbol{\mathcal{S}}^{(3)}_{(:,:,k)} = \boldsymbol{T}_1 \operatorname{diag}\left(\hat{\boldsymbol{c}}_k\right) \boldsymbol{T}_2^{\mathrm{T}}, \tag{7.18}$$

and

$$\operatorname{diag}\left(\hat{\boldsymbol{c}}_k\right) = \boldsymbol{T}_1^{-1} \cdot \boldsymbol{\mathcal{S}}^{(3)}_{(:,:,k)} \cdot (\boldsymbol{T}_2^{\mathrm{T}})^{-1}. \tag{7.19}$$

Equation (7.19) corresponds to a non-symmetric SMD [RH13]. An example illustration of the SMD in (7.18) is given in Figure 7.4. It can be converted into a symmetric one by multiplying (7.19) by the inverse of an arbitrary pivoting slice $p \in \{1, \ldots, K\}$ from either the right-hand side (rhs)

$$\begin{aligned} \boldsymbol{\mathcal{S}}^{(3)\mathrm{rhs}}_{(:,:,k)} &= \boldsymbol{\mathcal{S}}^{(3)}_{(:,:,k)} \cdot (\boldsymbol{\mathcal{S}}^{(3)}_{(:,:,p)})^{-1} && (7.20) \\ &= \boldsymbol{T}_1 \cdot \operatorname{diag}\left(\hat{\boldsymbol{c}}_k\right) \cdot \operatorname{diag}\left(\hat{\boldsymbol{c}}_p\right)^{-1} \cdot \boldsymbol{T}_1^{-1}, && (7.21) \end{aligned}$$

or the left-hand side (lhs)

$$\begin{aligned} \boldsymbol{\mathcal{S}}^{(3)\mathrm{lhs}}_{(:,:,k)} &= ((\boldsymbol{\mathcal{S}}^{(3)}_{(:,:,p)})^{-1} \cdot \boldsymbol{\mathcal{S}}^{(3)}_{(:,:,k)})^{\mathrm{T}} && (7.22) \\ &= \boldsymbol{T}_2 \cdot \operatorname{diag}\left(\hat{\boldsymbol{c}}_k\right) \cdot \operatorname{diag}\left(\hat{\boldsymbol{c}}_p\right)^{-1} \cdot \boldsymbol{T}_2^{-1}. && (7.23) \end{aligned}$$

The slice with the smallest condition number[4] is considered as a clever choice for a pivot. As it can be seen from (7.21) and (7.23), the transform matrices $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$ can be estimated by simultaneous diagonalization of the slices of $\boldsymbol{\mathcal{S}}^{(3)\mathrm{rhs}}$ and $\boldsymbol{\mathcal{S}}^{(3)\mathrm{lhs}}$, respectively. Moreover, from the diagonal elements of the jointly diagonalized matrices, estimates of $\boldsymbol{F}_3$ can be obtained from the right-hand side SMD (7.21) or the left-hand side SMD (7.23). The ambiguity that is brought to each column by $\operatorname{diag}\left(\hat{\boldsymbol{c}}_p\right)^{-1}$ lies within the scaling ambiguity that is inherent for any CPD and can therefore be ignored. The current implementation of SECSI employs the SMD algorithm described in [FG06], which builds up the transformation matrix iteratively out of alternating shear matrices and unitary transform matrices. The acquired $\boldsymbol{T}_1$ ($\boldsymbol{T}_2$) obtained from this SMD can be used to estimate $\boldsymbol{F}_1$ ($\boldsymbol{F}_2$) according to (7.17). After two of the three factor matrix estimates have been found, a least squares solution for the last factor can be computed from one of the following equations

$$\begin{aligned} \boldsymbol{F}_1 &= [\boldsymbol{\mathcal{T}}_0]_{(1)} \cdot (\boldsymbol{F}_2 \diamond \boldsymbol{F}_3)^{\mathrm{T}+}, \\ \boldsymbol{F}_2 &= [\boldsymbol{\mathcal{T}}_0]_{(2)} \cdot (\boldsymbol{F}_3 \diamond \boldsymbol{F}_1)^{\mathrm{T}+}, \\ \boldsymbol{F}_3 &= [\boldsymbol{\mathcal{T}}_0]_{(3)} \cdot (\boldsymbol{F}_1 \diamond \boldsymbol{F}_2)^{\mathrm{T}+}. \end{aligned} \tag{7.24}$$

[4] The condition number of a matrix $\boldsymbol{A}$ is defined as the ratio of its largest singular value to the smallest singular value and quantifies the sensitivity of the $\boldsymbol{Ax} = \boldsymbol{b}$ problem to the changes in $\boldsymbol{A}$ [GVL13].

The similarity between this problem and the generalized eigenvalue decomposition (GEVD) approach that is used to initialize the Tensorlab algorithm should be mentioned here. Whereas both algorithms are essentially pencil-based, the approach in [DD17] uses only two slices of the tensor to compute a generalized eigenvalue decomposition and, subsequently, estimate the factor matrices. In contrast, the SECSI framework uses an SMD-based approach that examines different modes and, instead of considering only one subpencil, takes all the slices of the tensor into account.

The authors in [BBV19] study the performances of the pencil-based algorithms for the CPD and state that "for every pencil-based algorithm, there exists an open set of the rank r tensors in $\mathbb{R}^{n_1 \times n_2 \times n_3}$ for which it is unstable" and the instability is caused by a significant difference between the condition number of a tensor rank decomposition [BV18b] in $\mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\mathbb{R}^{n_1 \times n_2 \times 2}$: the expected condition number of a tensor rank decomposition for $n_1 \times n_2 \times 2$ tensors is much larger on the average than the tensor condition number[5] for $n_1 \times n_2 \times n_3$ tensors. The paper also states that "as n_3 increases, very large condition numbers become increasingly unlikely". Consequently, the fact that we consider the simultaneous matrix diagonalization of tensors of size $d \times d \times K$ (equations (7.20)-(7.22)), according to the aforementioned paper, greatly increases the probability of a low tensor condition number, and consequently, leads to more stable solutions.

Another CPD computation technique that improves the accuracy of the GEVD-based solutions is proposed in [EVD22]. The authors introduce the generalized eigenspace decomposition (GESD) algorithm for computation of CPD that exploits not one but many subpencils of the tensor to find the generalized eigenvectors and eigenspaces that correspond to sufficiently well-separated generalized eigenvalues. Two of the three factor matrices are then obtained by combining information from the different subpencils. The GESD algorithm outperforms the GEVD, especially for the correlated factors case, and therefore its extension to block-term decompositions might be promising. Compared to the GESD, SECSI does not consider the subpencils of the tensor but all the slices jointly. Moreover, in the non-degenerate CPD cases, the algorithm allows computing the SMDs in different modes (not only in the 3-mode) to get the best solution.

[5] Here, by "condition number" and "tensor condition number" we refer to "condition number of tensor rank decomposition" [BV18b; BBV19].

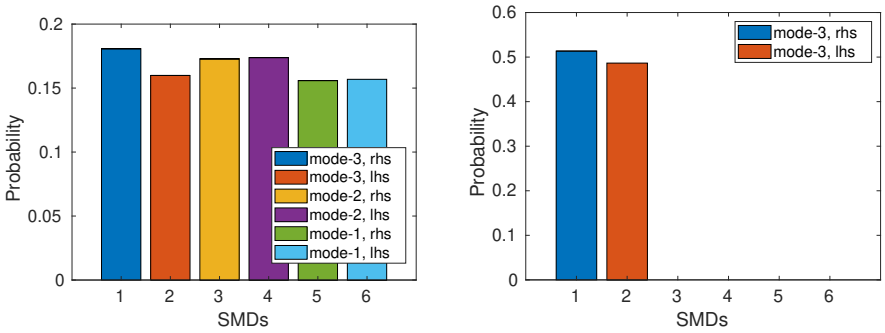
Choice of the Final Estimate

As mentioned above, the three factor matrices can either be retrieved by the left-hand side estimate or the right-hand side estimate, which enables a choice out of two sets of estimated factor matrices for each mode, resulting in 6 sets of factor matrix estimates for a tensor with three dimensions. The best accuracy in terms of the reconstruction error can be achieved by considering all combinations of estimated factor matrices $\boldsymbol{F}_1$, $\boldsymbol{F}_2$, and $\boldsymbol{F}_3$ (from all SMDs) and choosing the combination with the smallest reconstruction error as final estimates ("Best Matching (BM)" approach). Since, in this case, all the combinations, including the factor matrices from different SMDs have to be checked, this solution is computationally quite expensive. On the positive side, the SECSI framework offers several heuristics that allow controlling the trade-off between performance and complexity. Therefore, the final estimates of the framework can be chosen according to predefined heuristics, which are based on different selection criteria such as reconstruction error, condition number, or SMD residuals. The SECSI framework features the following heuristics to choose from [RH13]:

- *REC PS (reconstruction error - paired solutions)*: Instead of enabling factor matrix combinations from different SMDs as in the "BM" approach, only the combinations that originate from the same SMD (*"paired solutions"*) are evaluated. As an accuracy measure, this heuristic uses the *reconstruction error* between the tensor reconstructed from the estimated loading matrices and the original data tensor. The set of matrices that provides the smallest reconstruction error is picked as the final output.

- *RES* (*SMD residuals*): Out of all estimates, the set of factor matrices that originate from the "best" SMD is used, which is the SMD whose transformation matrix $\boldsymbol{T}$ provides the best diagonalization of the n-mode core tensor slices, thus leaving the smallest *residual error*. The residual error is defined as the average Frobenius norm of the off-diagonal elements of the tensor slice pencils after the estimated transformation matrices $\boldsymbol{T}$ and $\boldsymbol{T}^{-1}$ are applied to it.

- *CON PS (condition number - paired solutions)*: Instead of solving all possible SMD problems, only the two SMDs in the mode where the pivot slice has the best *condition number* are computed. The final solution is selected from the SMD (left-hand side or right-hand side) that yields the lower reconstruction error. Subsequently, the two solutions are compared in terms of their reconstruction error, and the one which yields the lower reconstruction error is returned as the final solution. The combinations between the estimates from the two different



(a) Canonical polyadic decomposition. CPD rank is $d = \Sigma L_r = 9$.

(b) Block-term decomposition. The ranks are $R = 3$ and $L_r = [3,3,3]$.

Figure 7.5. Histogram of paired solutions chosen by SECSI ("REC PS") after comparing their reconstruction errors. "lhs" and "rhs" refer to the left-hand side and right-hand side transformation matrix estimate, respectively. $\mathcal{T} \in \mathbb{R}^{20\times30\times40}$, SNR = 50 dB. Results are averaged over 1000 Monte-Carlo trials.

SMDs are not considered ("paired solutions"). This heuristic is the fastest since not all SMDs have to be solved.

These heuristics allow reducing computational complexity by offering an excellent trade-off between the complexity and accuracy of the solution. Since the smallest possible reconstruction error is achieved by the "best matching" approach, we consider it as a benchmark solution that determines the achievable reconstruction error in the SECSI framework.

The proposed BTD extension of SECSI uses new BTD heuristics which exploit the prior knowledge of the block-term structure of the tensor to reduce the computational load of SECSI.

7.4.2 Extension to block-term decomposition

When considering the link between the CP and rank-$(L_r, L_r, 1)$ block-term decompositions, the factor matrices $\boldsymbol{F}_1$, $\boldsymbol{F}_2$, and $\boldsymbol{F}_3$ in (7.15) correspond to the BTD factors $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}^{(\mathrm{r})}$ in (7.3), respectively. Then, the 3-mode matrix $\boldsymbol{F}_3$ correcponding to $\boldsymbol{C}^{(\mathrm{r})}$ will have collinear columns.

The SECSI framework is capable of computing an approximate CPD even in so-called "degenerate" cases, when one factor matrix has proportional (or highly correlated) columns or if one dimension of the tensor is smaller than the CP-rank of the tensor. In these cases, a non-singular transformation matrix for the corresponding mode does

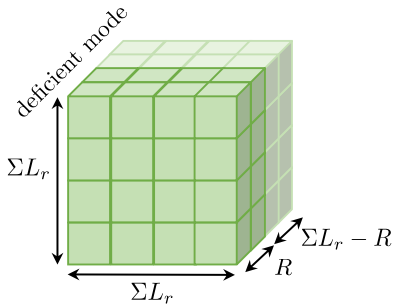


Figure 7.6. The core tensor $\mathcal{S}$ of the HOSVD with the rank-deficiency in the 3-mode.

not exist. However, to be able to generate an estimate for the CPD factor matrices of a three-way array, at least two of the three transformation matrices have to exist and have to be non-singular, so that equations (7.21) and (7.23) can be obtained. If a BTD scenario is viewed as a special case of the CPD, one of the CPD factor matrices must have repeated columns, leading to a degenerate case. A CPD approximation is still possible under these circumstances, although it is only essentially unique (see (7.5) in Section 7.2). Note that the existence of the transformation matrices coincides with one of the sufficient conditions for essential uniqueness defined in [DeL08a].

Reducing the Computational Load of the SMDs

The high numerical stability of the estimates (which makes them favorable) obtained via SECSI comes with a computational cost. If we consider the BTD as a special case of the CPD, we assume that one of the factors has linearly dependent columns. This can be exploited to decrease the computational load of the SECSI framework. For most of the heuristics, all six possible SMD have to be calculated in order to choose the best estimate. This is reasonable in case of the CPD because, apart from the different dimensions of the tensor for each of the modes, the CPD problem is a symmetrical one and the factor matrices often feature similar mathematical properties. In the case of a BTD however, the problem becomes asymmetrical due to the specific rank structure, which is also one of the reasons why two of the factor matrices are only unique up to the product of their submatrices. Moreover, this asymmetry affects the choice of the estimates in the last step of SECSI as follows.

If one of the modes is rank-deficient (assume the third mode in case of the rank-$(L_r, L_r, 1)$ BTD), the corresponding transform matrix and the SMDs where this transform matrix is present do not exist anymore. However, we can still construct the two SMDs (left-hand side and right-hand side, equations (7.21) and (7.23)) in the 3-mode.

Then, the 3-mode factor matrix can be directly obtained from the K diagonal matrices, and the remaining factors are estimated from the transform matrix and the least squares fit. This procedure will result in two sets of estimates for every factor matrix. An example illustration of the HOSVD core tensor $\boldsymbol{\mathcal{S}}$ with rank-deficiency in the 3-mode in shown in Figure 7.6.

The following numerical experiments demonstrate the probabilities with which the final solutions were chosen from particular SMDs when applying SECSI on CPD- and BTD-model tensors. First, we generate three factor matrices with CP rank ΣL_r and random complex Gaussian-distributed entries according to the model in (7.15) to build up a CPD tensor. After that, a noise tensor with a small power is added to the tensor. Then, we decompose the noise-distorted tensor via the SECSI framework with the "REC PS" heuristic and track which of the 6 sets of estimated factor matrices is used as the final output. As the histogram in Figure 7.5(a) shows, there are no preferred modes in the CPD scenario, and the final estimates can originate from all SMDs with a non-zero probability.

The second simulation is conducted in the same way as the one described above, but the tensor is now constructed from block-terms according to (7.1) and with the multilinear rank-$(L_r, L_r, 1)$ structure. Whereas the first and second mode factor matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ have ΣL_r distinct columns, the third mode factor matrix $\boldsymbol{C}^{(\mathrm{r})}$ has repeated columns as in (7.3). The results for the second simulation are shown in Figure 7.5(b). We can see that for the BTD-constructed tensor, in all cases, one of the estimates with the diagonalizations in the 3-mode is used, whereas no such tendency can be seen for the CPD-constructed tensor.

These simulations demonstrate the advantages of the utilization of different mode SMDs in the original SECSI framework for the CPD. In a given application, we might not know beforehand in which mode the rank-deficiency will occur. Most of the GEVD-based algorithms, by default, are applied to the 3-mode slices and fail to obtain an accurate solution when the rank-deficiencies occur in the other modes. SESCI, by contrast, constructs all SMDs, and allows choosing the solution with the best performance (in terms of the chosen selection criterion)

Nevertheless, the computation of the SMDs in all modes increases the computational time. Therefore, we have added new heuristics to the SECSI-framework: "REC PS BT" (*reconstruction error - paired solutions - block-terms*) and "RES BT" (*residual error - block-terms*). Similarly to the CPD-SECSI heuristics described in Section 7.4.1, they choose the final estimates based on the reconstruction error and SMD residual error, respectively. In contrast, to the original heuristics, they only consider the SMDs in

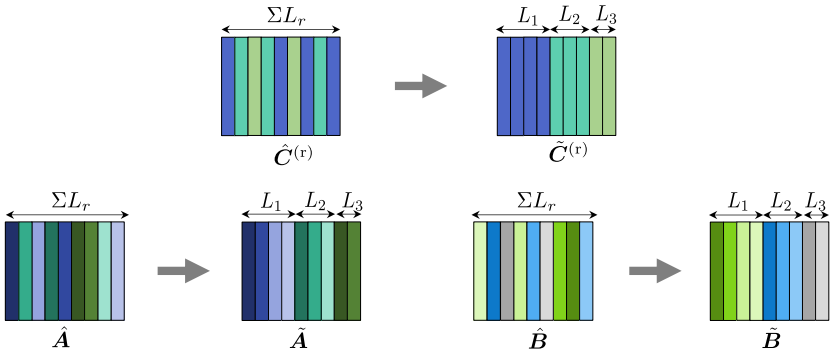


Figure 7.7. Clustering procedure to restore the original sub-matrix structure of the BTD blocks.

the rank-deficient mode. The "REC PS BT" heuristic solves both right-hand and left-hand side SMDs and picks the estimates from the SMD that provides the smallest reconstruction error. The "RES BT" heuristic computes the final estimates only from one SMD that provides the best diagonalization in terms of the residual error. This lowers the computation time of the SMDs up to a factor of three (depending on which heuristic is used as a reference).

Since in the BTD case, we only consider "paired solutions", i.e, the final estimates always originate from the same SMD problem, the scaling and permutation ambiguities of the factor matrices are consistent and do not affect the reconstruction error.

Restoring the submatrix structure

As it can be observed from (7.1) and (7.3), the BTD can be transformed into a CPD by repeating the r-th column of $\boldsymbol{C}$ in (7.1) L_r times. Whereas the transformation of $\boldsymbol{C}$ from the BTD into $\boldsymbol{C}^{(\mathrm{r})}$ in the CPD is therefore rather trivial (if the submatrix structure of $\boldsymbol{A}$ and $\boldsymbol{B}$ is assumed to be known), the reverse operation is a more challenging task. After performing an approximate CPD on the noise-corrupted BTD tensor, the estimated matrix $\hat{\boldsymbol{C}}^{(\mathrm{r})}$ will not perfectly match the structure of $\boldsymbol{C}^{(\mathrm{r})}$ in (7.4). The columns in $\hat{\boldsymbol{C}}^{(\mathrm{r})}$ will be arbitrarily permuted and scaled, and the repeated columns in $\boldsymbol{C}^{(\mathrm{r})}$ will not be explicitly equal in $\hat{\boldsymbol{C}}^{(\mathrm{r})}$. Therefore, one has to find an efficient way to reduce the number of columns of the estimated matrix $\hat{\boldsymbol{C}}^{(\mathrm{r})}$ to R columns to obtain $\hat{\boldsymbol{C}}$ without clipping off valuable information.

First, let us consider the output of the SECSI framework. In the noiseless case, the columns of $\hat{\boldsymbol{C}}^{(\mathrm{r})}$ are repetitions of the columns of $\hat{\boldsymbol{C}}$ in random order. For the computation of $\hat{\boldsymbol{C}}$ from the CPD approximation $\hat{\boldsymbol{C}}^{(\mathrm{r})}$, the submatrix structure of $\boldsymbol{A}$ and $\boldsymbol{B}$ has to be restored. This means that the columns that belong to the same $(L_r, L_r, 1)$-term need

to be adjacent to each other. In the noiseless case, this can be done by rearranging the columns of $\hat{\boldsymbol{C}}^{(\mathrm{r})}$ so that equal columns are adjacent and by rearranging the columns of $\hat{\boldsymbol{A}}$ and $\hat{\boldsymbol{B}}$ in the same way. If the tensor is disturbed by noise, however, the repeated columns of $\hat{\boldsymbol{C}}^{(\mathrm{r})}$ become more distinct from each other, bringing up the need for a more sophisticated approach. The k-means clustering [Llo82], a commonly used technique in data analysis, shows to be quite effective for this task since it only requires the number of clusters to be known beforehand. It is a simple, iterative algorithm that tries to find a set of cluster centers so that the summed squared distance of all elements to their nearest cluster center is minimized. In the BTD context, each cluster belongs to a different $(L_r, L_r, 1)$ block-term, and if there are no clustering errors, the r-th cluster will contain L_r columns.

In our simulations, we use the k-means++ algorithm [AD07] with a distance measure defined as a cosine of an angle between two unit-norm vectors ($\cos\theta = \frac{\mathrm{Re}\{\boldsymbol{c}_i^{\mathrm{H}}\boldsymbol{c}_j\}}{\|\boldsymbol{c}_i\|\|\boldsymbol{c}_j\|}$). After the columns of $\hat{\boldsymbol{C}}^{(\mathrm{r})}$ have been grouped into clusters, a permutation matrix $\boldsymbol{P}$ can be retrieved that rearranges the columns of $\hat{\boldsymbol{C}}^{(\mathrm{r})}$ so that all columns that belong to the same cluster are adjacent to each other. By multiplying $\hat{\boldsymbol{A}}$ and $\hat{\boldsymbol{B}}$ by the same permutation matrix, their submatrix structure can be restored (see Figure 7.7). The k-means clustering algorithm is also used in the GEVD initialization of the Tensorlab BTD algorithm [VDS+16]. It can be observed that in some cases, the k-means fails to generate clusters of the desired size from the GEVD estimates. In that event, the Tensorlab algorithm switches to random initialization. As discussed in Section 7.4.1, these instabilities might be caused by the fact that the expected tensor condition number of the selected two-slices subtensor might be very high on the average, compared to the expected tensor condition number of the more-than-two slices tensor. On the contrary, SECSI exploits the information of all available tensor slices which stabilizes its performance. In cases when the L_rs are not available, we utilize k-means to estimate the multilinear ranks. But if the multilinear ranks are known, we employ the constrained k-means [ZWL10] to guarantee the appropriate cluster sizes.

Another approach to calculate L_r based on symmetric joint block diagonalization is proposed in [DD20]. However, it still requires prior knowledge of R and ΣL_r.

7.5 Refinements

After clustering, the columns of $\hat{\boldsymbol{A}}$, $\hat{\boldsymbol{B}}$, and $\hat{\boldsymbol{C}}^{(\mathrm{r})}$ are sorted in a way that the columns that belong to the same submatrix are adjacent to each other. We denote the permuted matrices as $\tilde{\boldsymbol{A}}$, $\tilde{\boldsymbol{B}}$, and $\tilde{\boldsymbol{C}}^{(\mathrm{r})}$. Next, the following refinement steps can be applied to

Algorithm 3 SECSI-BTD framework for rank-$(L_r, L_r, 1)$ BTD with rank estimation

Require: 3-way tensor $\mathcal{T}$

1: $\hat{R}, \Sigma\hat{L}_r \leftarrow \text{LaRGE}_{\text{BT}}(\mathcal{T})$ ▷ *Ranks est.*
2: $\hat{\boldsymbol{A}}, \hat{\boldsymbol{B}}, \hat{\boldsymbol{C}}^{(\text{r})} \leftarrow \text{SECSI}_{\text{BT}}(\mathcal{T}, \Sigma\hat{L}_r)$ ▷ *Factors in CPD form*
3: $\boldsymbol{P}, \hat{L}_r\text{s} \leftarrow k\text{-means}(\hat{\boldsymbol{C}}, R)$ ▷ *Clustering*
4: $\tilde{\boldsymbol{A}}, \tilde{\boldsymbol{B}}, \tilde{\boldsymbol{C}}^{(\text{r})} \leftarrow \text{reordering}(\hat{\boldsymbol{A}}, \hat{\boldsymbol{B}}, \hat{\boldsymbol{C}}^{(\text{r})}, \hat{L}_r, \boldsymbol{P})$
5: $\tilde{\boldsymbol{A}}_r, \tilde{\boldsymbol{B}}_r, \tilde{\boldsymbol{c}}_r \leftarrow \text{refinement}$ ▷ $\boldsymbol{C}$ *in BTD form, ALS or NLS*

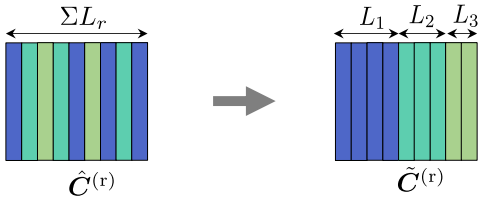


Figure 7.8. Reducing the colinear columns in a 3-mode (rank-deficient) matrix.

generate the final block-term estimates as in (7.1). At this stage, the matrix $\tilde{\boldsymbol{C}}^{(\text{r})}$ still contains the colinear columns ordered according to R clusters and needs to be brought to the BTD form. In other words, the colinear columns have to be reduced (see Figure 7.8). To perform the reduction effectively, i.e., without cutting off any data, we first refine the matrices $\tilde{\boldsymbol{A}}$ and $\tilde{\boldsymbol{B}}$ in an ALS fashion as follows

$$\tilde{\boldsymbol{A}} = \left[\mathcal{T}\right]_{(1)} (\tilde{\boldsymbol{B}} \diamond \tilde{\boldsymbol{C}}^{(\text{r})})^{\text{T}+}, \tag{7.25}$$

$$\tilde{\boldsymbol{B}} = \left[\mathcal{T}\right]_{(2)} (\tilde{\boldsymbol{C}}^{(\text{r})} \diamond \tilde{\boldsymbol{A}})^{\text{T}+}. \tag{7.26}$$

This step incorporates the multiple estimates of the columns of $\tilde{\boldsymbol{C}}^{(\text{r})}$ into $\tilde{\boldsymbol{A}}$ and $\tilde{\boldsymbol{B}}$. Then, an estimate of $\boldsymbol{C} \in \mathbb{F}^{K \times R}$ can be computed using equation for $\tilde{\boldsymbol{C}}$ from the ALS-based scheme for the BTD in [DN08]

$$\tilde{\boldsymbol{C}} = \left[\mathcal{T}\right]_{(3)} [(\tilde{\boldsymbol{A}}_1 \diamond \tilde{\boldsymbol{B}}_1) \cdot \mathbf{1}_{L_1}, \cdots, (\tilde{\boldsymbol{A}}_R \diamond \tilde{\boldsymbol{B}}_R) \cdot \mathbf{1}_{L_R}]^{\text{T}+}. \tag{7.27}$$

Next, to finalize the factor matrix estimation, we perform the refinements either using nonlinear least squares (NLS) or alternating least squares (ALS) [DN08; SVBD13] (We summarize the NLS in Appendix A.5). In the BTD-ALS procedure, the factor matrices $\tilde{\boldsymbol{A}}$ and $\tilde{\boldsymbol{B}}$ are updated as

$$\tilde{\boldsymbol{A}} = \left[\mathcal{T}\right]_{(1)} (\tilde{\boldsymbol{B}} \diamond_{\text{s}} \tilde{\boldsymbol{C}})^{\text{T}+}, \tag{7.28}$$

$$\tilde{\boldsymbol{B}} = \left[\mathcal{T}\right]_{(2)} (\tilde{\boldsymbol{C}} \diamond_{\text{s}} \tilde{\boldsymbol{A}})^{\text{T}+}, \tag{7.29}$$

where $\diamond_{\mathrm{s}}$ denotes the submatrix-wise Khatri-Rao product defined for two matrices $\boldsymbol{X} = [\boldsymbol{X}_1, ..., \boldsymbol{X}_R] \in \mathbb{F}^{I \times \Sigma L_r}$ and $\boldsymbol{Y} = [\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_R] \in \mathbb{F}^{J \times \Sigma L_r}$ as $\boldsymbol{X} \diamond_{\mathrm{s}} \boldsymbol{Y} = [\boldsymbol{X}_1 \otimes \boldsymbol{Y}_1, \ldots, \boldsymbol{X}_R \otimes \boldsymbol{Y}_R] \in \mathbb{C}^{IJ \times \Sigma L_r^2}$ (see Appendix A.1). The matrix $\tilde{\boldsymbol{C}}$ is updated as in (7.27). The NLS or ALS schemes are run until one of the stopping criteria is met (the relative change between two successive iterations is small, the maximum number of iterations is reached, or the change in objective function value relative to the tensor norm is less than a specified tolerance).

The proposed SECSI-BTD framework to compute the rank-$(L_r, L_r, 1)$ BTD with rank estimation is summarized in Algorithm 3. For tensors with a known rank structure, step 1 can be skipped.

7.6 Numerical results

In this section, we conduct simulations with synthetically generated data to assess the performance of the proposed algorithms and compare it to the algorithms from the Tensorlab toolbox [VDS+16; SVBD13] for computing an approximate rank-$(L_r, L_r, 1)$ decomposition.

The tensors for the simulations are constructed according to the rank-$(L_r, L_r, 1)$ BTD model in (7.1) where the factor matrices have been drawn from a zero mean circularly symmetric complex Gaussian (ZMCSCG) distribution with unit variance, and the noise tensor $\mathcal{N}$ have been formed from ZMCSCG entries with variance σ_n^2. Accordingly, the SNR is defined as $1/\sigma_n^2$.

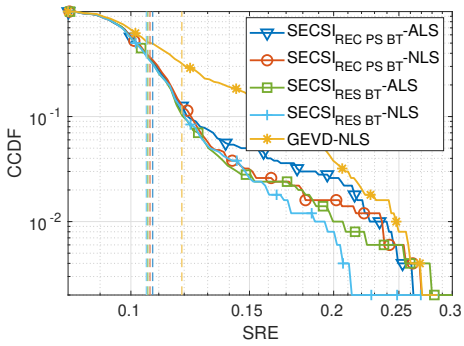
We use two different accuracy measures, a relative squared reconstruction error (SRE) and a relative squared factor error (SFE), to evaluate the accuracy of an estimated BTD. The SRE is defined as

$$\mathrm{SRE} = \frac{\left\| \hat{\mathcal{T}} - \mathcal{T}_0 \right\|_{\mathrm{F}}^2}{\left\| \mathcal{T}_0 \right\|_{\mathrm{F}}^2}, \tag{7.30}$$

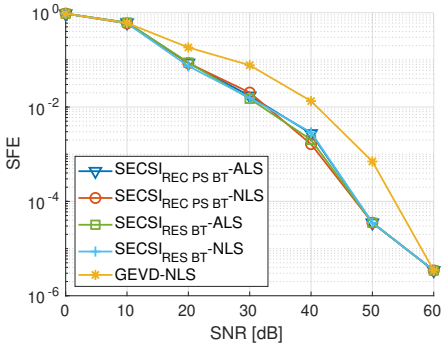
where $\mathcal{T}_0$ is the original noise-free tensor, and $\hat{\mathcal{T}}$ is a tensor reconstructed from the estimated BTD factors as $\hat{\mathcal{T}} = \sum_{r=1}^{R} \left(\tilde{\boldsymbol{A}}_r \cdot \tilde{\boldsymbol{B}}_r^{\mathrm{T}} \right) \circ \tilde{\boldsymbol{c}}_r$. The SFE for the matrix $\boldsymbol{C}$ is defined as

$$\mathrm{SFE} = \frac{\left\| \tilde{\boldsymbol{C}} \cdot \boldsymbol{P}_c - \boldsymbol{C} \right\|_{\mathrm{F}}^2}{\left\| \boldsymbol{C} \right\|_{\mathrm{F}}^2}, \tag{7.31}$$

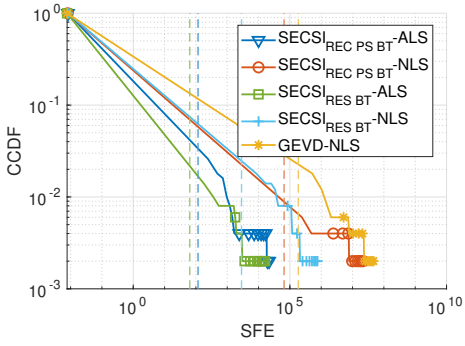
where the matrix $\boldsymbol{P}_c$ corrects the permutation and scaling ambiguity that is inherent in the estimation of the factor matrices. These ambiguities are resolved as follows. First, the columns of both $\tilde{\boldsymbol{C}}$ and $\boldsymbol{C}$ are normalized to unit norm. The permutation is



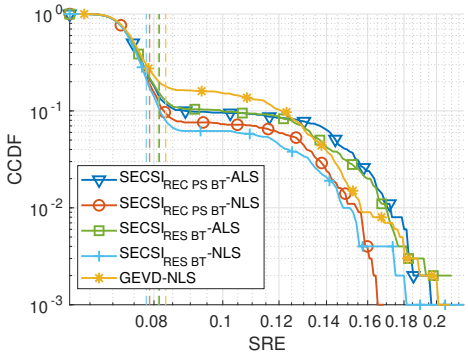
(a) CCDF vs. SRE. Asymmetric scenario. $L_r = [2,3,4]$, SNR = 30 dB.



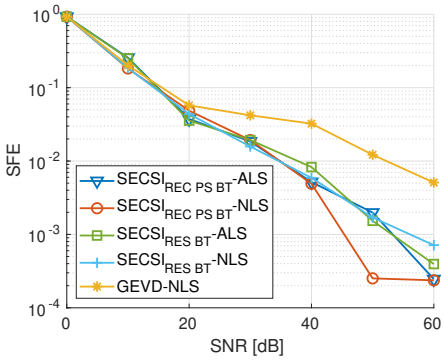
(b) SFE vs. SNR for $\boldsymbol{C}$. Asymmetric scenario. $L_r = [2,3,4]$.



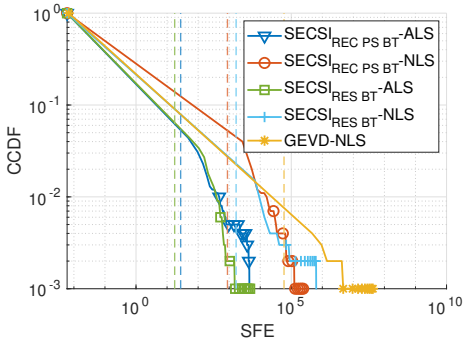
(c) CCDF vs. SFE for $\boldsymbol{A}_r \boldsymbol{B}_r^{\mathrm{T}}$. Asymmetric scenario. $L_r = [2,3,4]$, SNR = 30 dB.



(d) CCDF vs. SRE. Symmetric scenario. $L_r = [5,5,5,5]$, SNR = 30 dB.



(e) SFE vs. SNR for $\boldsymbol{C}$. Symmetric scenario. $L_r = [5,5,5,5]$.



(f) CCDF vs. SFE for $\boldsymbol{A}_r \boldsymbol{B}_r^{\mathrm{T}}$. Symmetric scenario. $L_r = [5,5,5,5]$, SNR = 30 dB.

Figure 7.9. $\mathcal{T} \in \mathbb{C}^{20\times30\times40}$ with asymmetric and symmetric multilinear rank structures. 1000 realizations of a tensor.

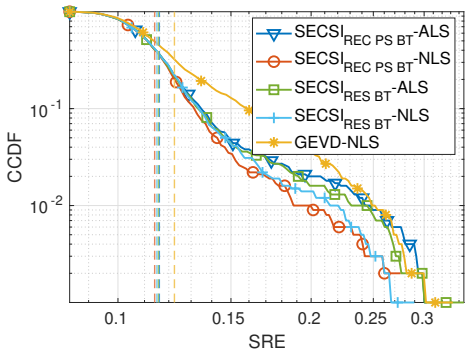
corrected by rearranging the columns in $\tilde{\boldsymbol{C}}$ so that the inner product $\tilde{\boldsymbol{c}}_r^{\mathrm{H}} \cdot \boldsymbol{c}_r$ between the estimated and original factor columns is maximized. Then, the scaling is resolved by multiplying the estimated columns by their scalar projection $\tilde{\boldsymbol{c}}_r^{\mathrm{H}} \cdot \boldsymbol{c}_r$ in the direction of the original vectors $\boldsymbol{c}_r$. Whereas the matrix $\boldsymbol{C}$ can be estimated up to scaling and column permutation ambiguities, the multilinear factor matrices $\boldsymbol{A}_r$ and $\boldsymbol{B}_r$ are only unique up to the products of their submatrices due to the submatrix product ambiguities show in (7.5). Therefore, when evaluating the performance for the matrices $\boldsymbol{A}_r$ and $\boldsymbol{B}_r$, in case of the SFE calculations we consider the products $\boldsymbol{A}_r \boldsymbol{B}_r^{\mathrm{T}}$ as follows

$$\mathrm{SFE}(\boldsymbol{A}\boldsymbol{B}^{\mathrm{T}}) = \frac{1}{R} \sum_{r=1}^{R} \frac{\left\| \tilde{\boldsymbol{A}}_r \tilde{\boldsymbol{B}}_r^{\mathrm{T}} - \boldsymbol{A}_r \boldsymbol{B}_r^{\mathrm{T}} \right\|_{\mathrm{F}}^2}{\left\| \boldsymbol{A}_r \boldsymbol{B}_r^{\mathrm{T}} \right\|_{\mathrm{F}}^2}, \tag{7.32}$$

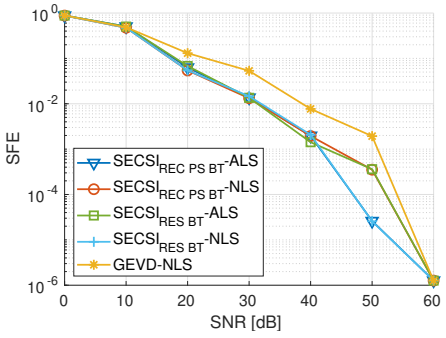
where $\boldsymbol{A}_r$ and $\boldsymbol{B}_r$ are the original factor matrices, and $\tilde{\boldsymbol{A}}_r$ and $\tilde{\boldsymbol{B}}_r$ are the estimated factors after resolving the permutation and scaling ambiguities.

We compare the estimation performances of the proposed SECSI-BTD algorithm with the performance of the NLS algorithm with GEVD initialization from [SVBD13], which has been commonly used in many BTD applications [CQA17; SBG17; dVD+07; CRHH21]. For the proposed SECSI-BTD approach, we examine both the ALS and NLS refinement procedures. ALS stops when the change in the reconstruction error between two successive iterations is less than a specified threshold (10^{-6}) or the maximum number of iterations (30) is reached. For the NLS refinement, we use the nonlinear least squares procedure based on Gauss-Newton with dogleg trust region from Tensorlab [VDS+16] with its default settings. The algorithm stops when one of the following criteria is met: the maximum number of iterations has been reached (200), the change in the objective function value relative to the norm of the tensor is less than a specified tolerance (10^{-12}), or the ratio of the step size relative to the norm of the current iterate is less than a specified tolerance (10^{-6}). To ensure a fair comparison, the rank structure is assumed to be known for all algorithms.

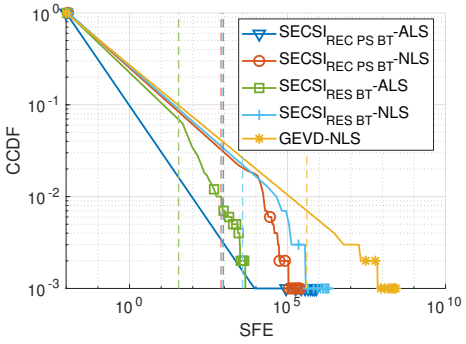
In the simulations, we consider four scenarios with asymmetric and symmetric multilinear rank structures. In the first two scenarios, the tensors are of size $(20 \times 30 \times 40)$ with the rank structures $R = 3$, $L_r = [2,3,4]$ and $R = 4$, $L_r = [5,5,5,5]$. In the other two scenarios, the tensors have a smaller 3-mode dimension and are of size $(30 \times 40 \times 15)$ with $L_r = [2,3,4]$ and $L_r = [2,2,2]$. The SRE and SFE performances for $\boldsymbol{\mathcal{T}} \in \mathbb{C}^{20\times30\times40}$ and $\boldsymbol{\mathcal{T}} \in \mathbb{C}^{30\times40\times15}$ are shown in Figures 7.9 and 7.10, respectively. The blue curves denote the proposed SECSI-BTD approach with the "REC PS BT" heuristic and ALS refinement, red lines correspond to SECSI-BTD with "REC PS BT" and NLS refinement, the green and the light blue curves denote the SECSI-BTD approach with the



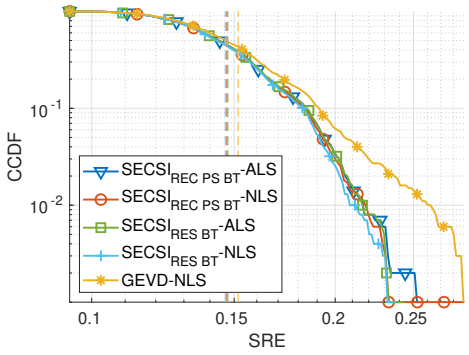
(a) CCDF vs. SRE. Asymmetric scenario. $L_r = [2,3,4]$, SNR = 30 dB.



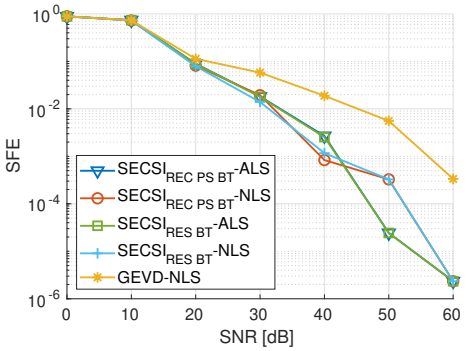
(b) SFE vs. SNR for $\boldsymbol{C}$. Asymmetric scenario. $L_r = [2,3,4]$.



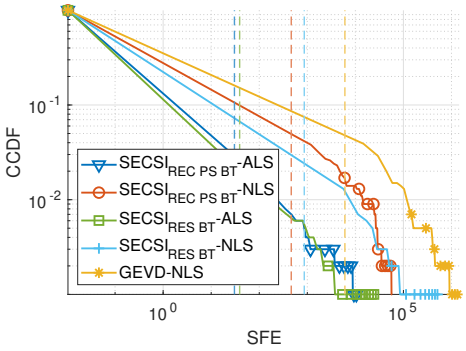
(c) CCDF vs. SFE for $\boldsymbol{A}_r \boldsymbol{B}_r^{\mathrm{T}}$. Asymmetric scenario. $L_r = [2,3,4]$, SNR = 30 dB.



(d) CCDF vs. SRE. Symmetric scenario. $L_r = [2,2,2]$, SNR = 30 dB.



(e) SFE vs. SNR for $\boldsymbol{C}$. Symmetric scenario. $L_r = [2,2,2]$.



(f) CCDF vs. SFE for $\boldsymbol{A}_r \boldsymbol{B}_r^{\mathrm{T}}$. Symmetric scenario. $L_r = [2,2,2]$, SNR = 30 dB.

Figure 7.10. $\boldsymbol{\mathcal{T}} \in \mathbb{C}^{30\times 40\times 15}$ with asymmetric and symmetric multilinear rank structures. 1000 realizations of a tensor.

Scenario	SECSI-BTD-ALS ("REC PS BT")	SECSI-BTD-ALS ("RES BT")	SECSI-BTD-NLS ("REC PS BT")	SECSI-BTD-NLS ("RES BT")	GEVD-NLS
$30 \times 40 \times 15$ $L_r = [2,2,2]$	0.5216	0.5184	1.4900	1.4752	1.3346
$30 \times 40 \times 15$ $L_r = [2,3,4]$	1.1037	1.2001	1.9106	2.2164	1.4910
$20 \times 30 \times 40$ $L_r = [5,5,5,5]$	29.4994	27.9834	32.5071	30.4271	2.0646
$20 \times 30 \times 40$ $L_r = [2,3,4]$	3.2022	3.2068	4.1025	3.9924	1.6720

Table 7.1. Average run time of the algorithms in different scenarios [sec].

"RES" heuristic and ALS or NLS refinements, respectively. The NLS algorithm with GEVD-based initialization from [SVBD13] is denoted with yellow color.

Figures 7.9(a), 7.9(d), 7.10(a) and 7.10(d) show the complementary cumulative distribution functions (CCDF) with respect to the reconstruction errors for all scenarios (for better visual representation without overwhelming the figures, we plot only 10 out of 1000 markers for the bins of the CCDF plots).

The vertical dashed lines represent the mean of the errors for each algorithm. The factor reconstruction errors with respect to SNRs for the matrix $\boldsymbol{C}$ in different scenarios are shown in Figures 7.9(b), 7.9(e), 7.10(b) and 7.10(e). As it can be observed, the proposed algorithms outperform the GEVD-based scheme in a large range of SNRs for all scenarios. In the scenario with $L_r = [5,5,5,5]$ (difficult scenario, since $\Sigma L_r = I = 20$) the SECSI-BTD schemes with the NLS refinement show a better convergence behavior than the other algorithms. The CCDFs with respect to SFE for the matrix products $\boldsymbol{A}_r \boldsymbol{B}_r^{\mathrm{T}}$ are presented in Figures 7.9(c), 7.9(f), 7.10(c) and 7.10(f). As can be seen from the CCDF plots, the GEVD scheme results in more outliers, and the proposed algorithms provide better initialization and, consequently, more stable performance in all scenarios. The explanations of the increased robustness of the SMD-based initialization are discussed in Section 7.4.1.

The average run times of the algorithms for SNR = 20 dB are shown in Table 7.1 (the algorithm performances were evaluated on an Intel Xeon Gold 6342 CPU 2.80GHz machine running Linux CentOS 7, kernel 3.10.0-1160.el7.x86_64 and MATLAB R2020b 64-bit). We can observe that for the scenario with $\mathcal{T} \in \mathbb{C}^{30\times40\times15}$ and $L_r = [2,2,2]$,

the SECSI-BTD with ALS refinement is almost three times faster than the GEVD-NLS and SECSI-NLS algorithms, which show similar time performance. In case of the scenario with $\mathcal{T} \in \mathbb{C}^{30\times40\times15}$ and $L_r = [2,3,4]$, the algorithms show a comparable time performance with the SECSI-BT-ALS being slightly faster on average. For the scenarios with $\mathcal{T} \in \mathbb{C}^{20\times30\times40}$ the proposed algorithms are slower than the GEVD-NLS solutions. However, considering that the SECSI-BTD schemes provide more accurate and reliable error performance results, they can be an appealing solution for applications where accuracy is more important. Moreover, SECSI-BTD is still faster than the GEVD-based approaches for the cases where the dimension in the rank-deficient mode is smaller than in other modes. These differences in the computational time performances are explained by the fact that the rank-deficient mode determines the dimensionality of the SMD problem: it increases the number of matrices to be jointly diagonalized. Similary, the time increase in the third scenario is related to the increased sizes of simultaneously diagonalized matrices ($\Sigma L_r = 20$). However, for these scenarios, the complexity might still be decreased by considering a more efficient diagonalization algorithm than in [FG06].

7.7 Conclusions

In this chapter of the thesis, we have exploited the connection between the Canonical Polyadic and the rank-$(L_r, L_r, 1)$ block-term decompositions and presented the SECSI-BTD framework to compute an approximate rank-$(L_r, L_r, 1)$ BTD. We have divided the proposed algorithm into three main blocks, which include the rank structure estimation, the initial estimation of the BTD factors, and a final refinement procedure. For the rank structure estimation, we have introduced an extension of the LaRGE technique for CPD model order estimation to estimate the number of blocks and the sum of L_rs in the BTD. Moreover, a procedure that uses clustering for the estimation of multilinear ranks has been presented. Furthermore, we have shown how the Semi-algebraic framework for approximate CPD via Simultaneous Matrix Diagonalizations (SECSI) can be employed for the computation of the initial BTD factors. Additionally, new heuristics have been added to the original algorithm to reduce the computational time and make the estimation more efficient. In the last block of the algorithm, we have presented the clustering and refinement procedures that return the final rank-$(L_r, L_r, 1)$ decomposition estimates. The simulation results have shown that the proposed SECSI-BTD algorithm outperforms the state-of-the-art techniques in terms of accuracy and robustness, especially in the context of factor reconstruction errors. The run time simulations show that the time complexity depends on the ranks

and dimensionality of the rank-deficient mode, and the algorithm is faster than the state-of-the-art schemes for some scenarios. Moreover, our algorithm does not require multiple initializations or a known rank structure. However, it can still take advantage of the known ranks when available, which makes it more flexible than the existing techniques in the literature.

Coupled Rank-$(L_r, L_r, 1)$ Block-Term Decomposition

8

Coupled tensor decompositions proved to be a valuable tool for many signal processing applications, including biomedical data analysis, source separation, data fusion, and many others. In this chapter of the thesis, we present an algorithm to calculate the coupled block-term decomposition (BTD) of multiple three-dimensional tensors with a coupled mode and possibly an unknown rank structure. The proposed approach is an extension of the SECSI-BTD algorithm in Chapter 7 to the coupled tensors case and composed of three main parts, the first is an extension of the linear regression of global eigenvalues (LaRGE) technique to estimate the number of blocks and sum of multilinear ranks in the coupled tensors. The second part accounts for the calculation of the preliminary multilinear factors of the coupled BTD by means of the semi-algebraic framework for approximate canonical polyadic decompositions via simultaneous matrix diagonalizations (SECSI). The last part contains the final refinement procedures that return the estimated BTD factors. Moreover, we provide some synthetic data simulations showing that the proposed algorithm demonstrates more stable and accurate results than the schemes from the literature [SKKH22].

8.1 Introduction

The apparent popularity of tensors in all possible signal processing areas is an eloquent indication that they are evidently an effective and promising tool for many applications, from statistics and communications to linguistics and big data analysis. Apart from their mild uniqueness conditions and data structure preserving qualities, tensor decompositions provide effective denoising and feature extraction capabilities [KB09].

The focus of this chapter falls on the block-term decompositions (BTD), to be more specific, on the coupled block-term decomposition in rank-$(L_r, L_r, 1)$ terms. Although it might not be yet as widely used as, for example, the higher-order singular value decomposition (HOSVD) [DDV00a; Tuc66] or the canonical polyadic decomposition

(CPD) (sometimes also referred to as CANDECOMP (canonical decomposition) or PARAFAC (parallel factors)) [CC70; Har70], the BTDs prove to be particularly interesting for applications where the data have a heterogeneous rank structure and possess collinearity is some of the modes.

There are some variations in the definitions of the block-term decomposition based on the ranks of the block-terms. These definitions, uniqueness conditions, and the algebraic and optimization-based algorithms are well described in [DeL08b; DeL08a; DN08; ND09; SVBD13; DeL11]. From the applications point of view, the rank-$(L_r, L_r, 1)$ block-term decomposition appears to be the most attractive one. It has found its applications in communications [DB08], biomedical data processing [OZ19; RAZ16; Zar17; CRHH21], image and graph analysis [GPP20], and many others [YGL+19; HAK+17b; ZFHW19; ZFW+20].

Commonly, when it comes to applications, the question that arises before applying any low-rank approximation concerns the rank of the tensor. Given that it is rarely known beforehand, one should find a way to define it. Some techniques to estimate the factors and the ranks of a single block-term tensor are introduced in [RKG21b; RKG21a; RGK21; GOF+20].

Another direction, in which the BTDs are evolving, is the computation of coupled decompositions and data fusion. Many applications that have multiple related data sets to be analyzed can genuinely benefit from coupled factorizations. They often provide more flexible uniqueness conditions and improved feature extraction capabilities, and thus deeper insights into the data. Several studies on the uniqueness and the algorithms for coupled decompositions, including the coupled BTD, can be found in [GLD+16; SD15; SDD15; SVBD15].

Main contributions. In this chapter, we propose a new approach to estimate the factors of the coupled block-term decomposition of multiple tensors with a common mode based on the semi-algebraic framework for approximate coupled canonical polyadic (CP) decompositions via simultaneous matrix diagonalizations (SECSI) [RH13; MGK+21]. The coupled SECSI-BTD algorithm that we present in this chapter is an extension of the SECSI-BTD algorithm in Chapter 7 to the scenarios with M coupled tensors. Moreover, we show how the multilinear rank structure of the coupled rank-$(L_r, L_r, 1)$ decomposition can be estimated based on the extension of the linear regression of global eigenvalues (LaRGE) and clustering techniques. The block-diagram of the proposed coupled rank-$(L_r, L_r, 1)$ decomposition is presented in Figure 8.1. The simulation results show that compared to the algorithm in [VDS+16; SVBD15], which requires a known rank structure to perform the decomposition, the proposed

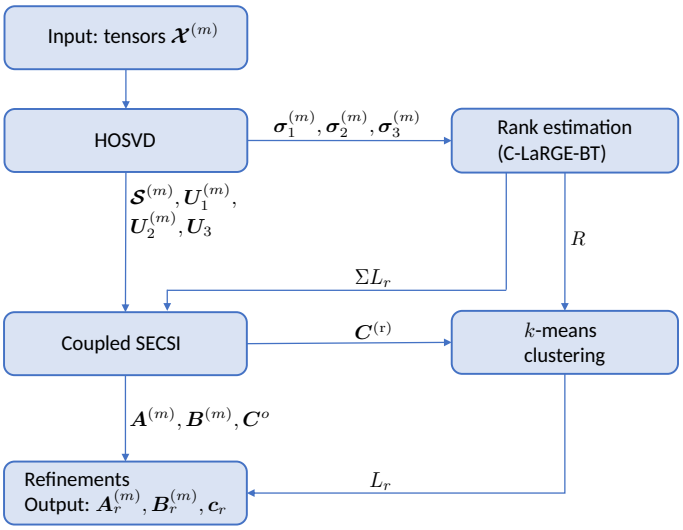


Figure 8.1. The block-diagram of the proposed coupled rank-$(L_r, L_r, 1)$ decomposition.

algorithm is more accurate, numerically stable, and can perform the decomposition even if the ranks are not available beforehand.

As we note in Chapter 7, for notational simplicity, by writing "BTD" or "rank-$(L_r, L_r, 1)$ decomposition" we refer to the block-term decomposition in rank-$(L_r, L_r, 1)$ terms, and we use ΣL_r to denote $\sum_{r=1}^{R} L_r$.

The rest of this chapter is organized as follows. Section 8.2 provides the coupled rank-$(L_r, L_r, 1)$ block-term decomposition model. In Section 8.3, we present an algorithm to estimate the ranks of the coupled rank-$(L_r, L_r, 1)$ decomposition. Then, we introduce the coupled SECSI-BTD technique to estimate the factor matrices of the rank-$(L_r, L_r, 1)$ decomposition in Section 8.4. The numerical results are shown in Section 8.5, while the conclusions are drawn in Section 8.6.

8.2 Data model

Let us consider a set of M noise-corrupted three-dimensional tensors $\boldsymbol{\mathcal{X}}^{(m)} \in \mathbb{C}^{I_1^{(m)} \times I_2^{(m)} \times I_3}$ with the third mode in common[1] (Figure 8.2). Then the coupled rank-$(L_r, L_r, 1)$ BTD of M tensors can be written as follows

$$\boldsymbol{\mathcal{X}}^{(m)} = \boldsymbol{\mathcal{X}}_0^{(m)} + \boldsymbol{\mathcal{N}}^{(m)} = \sum_{r=1}^{R} \left(\boldsymbol{A}_r^{(m)} \cdot \boldsymbol{B}_r^{(m)\mathrm{T}} \right) \circ \boldsymbol{c}_r + \boldsymbol{\mathcal{N}}^{(m)} \tag{8.1}$$

$$= \boldsymbol{\mathcal{T}}_{3,\Sigma L_r} \times_1 \boldsymbol{A}^{(m)} \times_2 \boldsymbol{B}^{(m)} \times_3 \boldsymbol{C}^{(\mathrm{r})} + \boldsymbol{\mathcal{N}}^{(m)}, \tag{8.2}$$

where $m \in \{1, \ldots, M\}$, and R is a number of block-terms with ranks L_r, $r \in \{1, \ldots, R\}$ (for notational simplicity, we assume that the multilinear ranks $L_r^{(m)}$ are equal for all M tensors ($L_r^{(1)} = L_r^{(2)} = \ldots = L_r^{(m)} = L_r$), which is usually the case for the coupled data sets[2]). Moreover, $\boldsymbol{\mathcal{X}}_0^{(m)} \in \mathbb{C}^{I_1^{(m)} \times I_2^{(m)} \times I_3}$ is the noise-free BTD tensor, and $\boldsymbol{\mathcal{N}}^{(m)} \in \mathbb{C}^{I_1^{(m)} \times I_2^{(m)} \times I_3}$ is the additive noise tensor.

The factors $\boldsymbol{A}_r^{(m)} \in \mathbb{C}^{I_1^{(m)} \times L_r}$ can be stacked in a matrix $\boldsymbol{A}^{(m)}$ as $\boldsymbol{A}^{(m)} = [\boldsymbol{A}_1^{(m)}, \boldsymbol{A}_2^{(m)}, ..., \boldsymbol{A}_R^{(m)}] \in \mathbb{C}^{I_1^{(m)} \times \Sigma L_r}$, and the individual factors $\boldsymbol{B}_r^{(m)} \in \mathbb{C}^{I_2^{(m)} \times L_r}$ are represented as a submatrices of $\boldsymbol{B}^{(m)} = [\boldsymbol{B}_1^{(m)}, \boldsymbol{B}_2^{(m)}, ..., \boldsymbol{B}_R^{(m)}] \in \mathbb{C}^{I_2^{(m)} \times \Sigma L_r}$. The common vectors $\boldsymbol{c}_r$ are stacked in the matrix $\boldsymbol{C} = [\boldsymbol{c}_1, \boldsymbol{c}_2, ..., \boldsymbol{c}_R] \in \mathbb{C}^{I_3 \times R}$. Then, it can be seen from (8.2), that the BTD in (8.1) may be viewed as a CP decomposition with CP-rank ΣL_r and with repeated or linearly depended columns in $\boldsymbol{C}^{(\mathrm{r})}$

$$\boldsymbol{C}^{(\mathrm{r})} = \left[\boldsymbol{c}_1 \cdot \mathbf{1}_{L_1}^{\mathrm{T}}, \boldsymbol{c}_2 \cdot \mathbf{1}_{L_2}^{\mathrm{T}}, \cdots, \boldsymbol{c}_R \cdot \mathbf{1}_{L_R}^{\mathrm{T}} \right] \in \mathbb{C}^{I_3 \times \Sigma L_r}. \tag{8.3}$$

In this study, we use this link between the two decompositions to first calculate the initial estimates in the coupled CPD form and subsequently convert them to coupled BTD factors.

[1] In general, any mode (or two modes) can be coupled. For example, in Chapter 9, we apply the coupled BTD to multi-static radar systems and consider a coupled rank-$(L_r, L_r, 1)$ decomposition with the 2-mode in common. The authors in [CRHH21] use the coupled BTD of two tensors with the 3-mode in common for an analysis of electroencephalography (EEG) and magnetoencephalography (MEG) data. Moreover, in Chapter 10, we consider the coupled rank-$(L_r, L_r, 1)$ BTD of four tensors (one EEG and three MEG tensors) with the 3-mode in common.

[2] The ranks $L_r^{(m)}$ can be different across the block-terms r.

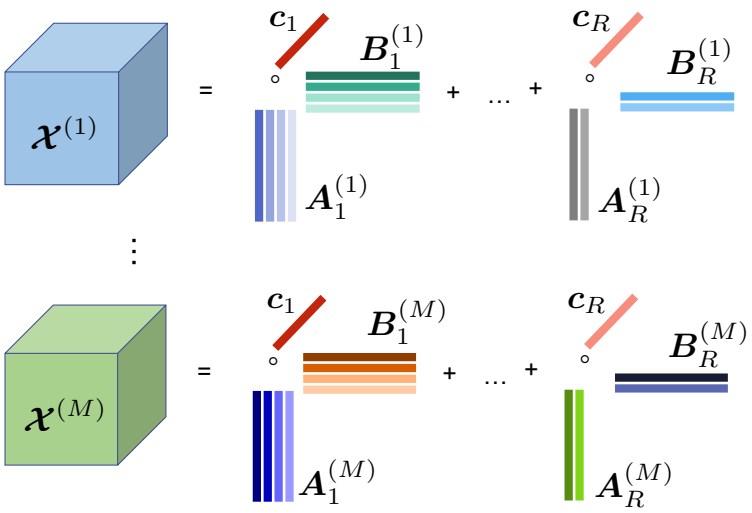


Figure 8.2. Coupled rank-$(L_r, L_r, 1)$ decomposition (noiseless case).

8.3 Rank structure estimation

Since in most of the tensor decomposition applications the rank structure is usually unknown, its correct estimation becomes essential in relation to low-rank tensor decompositions. While a number of algorithms for CPD rank estimation are available in the literature, there are not yet many options for block-term decompositions. Moreover, to the best of our knowledge, there are no techniques devoted to the estimation of ranks in coupled BTD models. Therefore, in this contribution, we present an extension of the LaRGE [KDHH21; AKK+22] and the LaRGE-BTD techniques from Chapter 7 to estimate the rank structure of the coupled rank-$(L_r, L_r, 1)$ decomposition. The LaRGE algorithm estimates the CP-rank of a tensor by performing a linear regression of its noise eigenvalues, which commonly follow an exponential decline, i.e., a linear decline on a logarithmic scale. Since the signal eigenvalues do not possess the same property, the linear approximation will fail when reaching the signal eigenvalues, which will indicate the detection of the rank. In the following, we present the extension of LaRGE for coupled BTD rank estimation.

Given a set of M noise-corrupted three-dimensional tensors $\boldsymbol{\mathcal{X}}^{(m)} \in \mathbb{C}^{I_1^{(m)} \times I_2^{(m)} \times I_3}$ as in (8.1), the coupled HOSVD of $\boldsymbol{\mathcal{X}}^{(m)}$ can be written as

$$\boldsymbol{\mathcal{X}}^{(m)} = \boldsymbol{\mathcal{S}}^{(m)} \times_1 \boldsymbol{U}_1^{(m)} \times_2 \boldsymbol{U}_2^{(m)} \times_3 \boldsymbol{U}_3, \tag{8.4}$$

where $\boldsymbol{\mathcal{S}}^{(m)} \in \mathbb{C}^{I_1^{(m)} \times I_2^{(m)} \times I_3}$ are the core tensors, $\boldsymbol{U}_1^{(m)} \in \mathbb{C}^{I_1^{(m)} \times I_1^{(m)}}$ and $\boldsymbol{U}_2^{(m)} \in \mathbb{C}^{I_2^{(m)} \times I_2^{(m)}}$ are composed of the left singular vectors of the 1- and 2-mode unfoldings of $\boldsymbol{\mathcal{X}}^{(m)}$

computed from $[\boldsymbol{\mathcal{X}}^{(m)}]_{(n)} = \boldsymbol{U}_n^{(m)} \boldsymbol{\Sigma}_n^{(m)} \boldsymbol{V}_n^{(m)\mathrm{H}}$, where $\boldsymbol{\Sigma}_n^{(m)}$ contains the n-mode singular values $\sigma_{(n),i}^{(m)}$, $n \in \{1,2\}$ on its main diagonal. Moreover, $\boldsymbol{U}_3 \in \mathbb{C}^{I_3 \times I_3}$ is the common unitary matrix calculated from the singular value decomposition (SVD) of the concatenated 3-mode unfoldings of $\boldsymbol{\mathcal{X}}^{(m)}$ [MGK+21], and therefore, $\sigma_{(3),i}^{(m)}$ is common for all tensors. For model order estimation, we are interested in the first $J = \min\left(I_3, I_n^{(m)}\right)$ singular values.

Due to the asymmetric rank structure, we divide the rank estimation processes into two separate procedures. In the first run, only the semi-global eigenvalues $\tilde{\lambda}_{(1,2),i}^{(m)}$, $i = \{1, \ldots, J\}$

$$\tilde{\lambda}_{(1,2),i}^{(m)} = \left(\sigma_{(1),i}^{(m)}\right)^2 \cdot \left(\sigma_{(2),i}^{(m)}\right)^2 \tag{8.5}$$

from the 1- and 2-mode are used to estimate ΣL_r. To take into account the information from all coupled tensors, we compute the geometric mean of M semi-global eigenvalues in (8.5) as follows

$$\tilde{\lambda}_{(1,2),i} = \sqrt[M]{\tilde{\lambda}_{(1,2),i}^{(1)} \cdot \tilde{\lambda}_{(1,2),i}^{(2)} \cdot \ldots \cdot \tilde{\lambda}_{(1,2),i}^{(M)}}. \tag{8.6}$$

Since there are rank-deficiencies in the 3-mode, we consider it separately to estimate the number of block-terms R using the 3-mode eigenvalues $\tilde{\lambda}_{(3),i}^{(m)} = \left(\sigma_{(3),i}^{(m)}\right)^2$ averaged along all M tensors as

$$\tilde{\lambda}_{(3),i} = \sqrt[M]{\tilde{\lambda}_{(3),i}^{(1)} \cdot \tilde{\lambda}_{(3),i}^{(2)} \cdot \ldots \cdot \tilde{\lambda}_{(3),i}^{(M)}}. \tag{8.7}$$

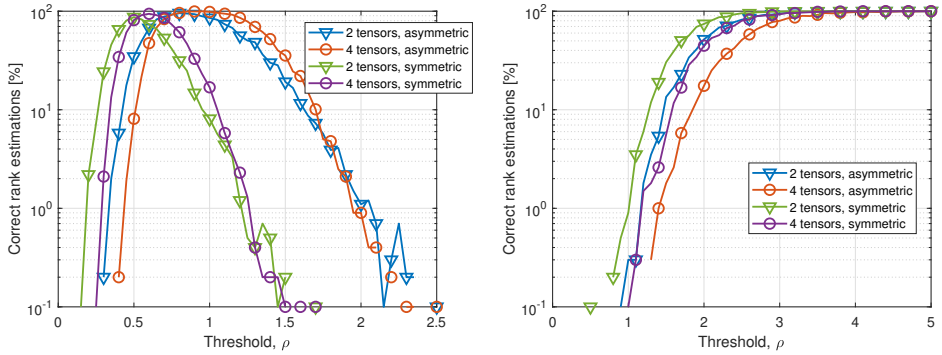
Then, the linear regression is applied separately to $\lambda_{(1,2),i} = \ln\left(\tilde{\lambda}_{(1,2),i}\right)$ and $\lambda_{(3),i} = \ln\left(\tilde{\lambda}_{(3),i}\right)$ to estimate ΣL_r and R, respectively. For notational convenience, in the following, we simply use λ_i which will refer either to $\lambda_{(1,2),i}$ or $\lambda_{(3),i}$, depending on which rank, ΣL_r or R, is to be estimated. The estimation starts from the smallest noise eigenvalue ($i = J$), and subsequently finds a linear prediction $\hat{\lambda}_i$ by moving up the scale. For each step k in the prediction, the relative prediction error is computed as

$$\delta_{J-k} = \frac{\lambda_{J-k} - \hat{\lambda}_{J-k}}{\left|\hat{\lambda}_{J-k}\right|} = \frac{\Delta_{J-k}}{\left|\hat{\lambda}_{J-k}\right|}, \tag{8.8}$$

followed by the standard deviation of the approximation error

$$\sigma_{J-k} = \sqrt{\frac{1}{k} \sum_{i=J}^{J-k} \left(\Delta_i - \frac{1}{k} \sum_{i=J}^{J-k} \Delta_i\right)^2}, \tag{8.9}$$

where Δ_i is the absolute prediction error. Then, by computing the ratio between the relative prediction error and the standard deviation of the approximation errors in the



(a) Multilinear ranks ΣL_r.

(b) Number of blocks R.

Figure 8.3. Correct rank estimations vs. Threshold. Simulation parameters: $\Sigma L_r = 9$, $R = 3$, $L_r = [2,3,4]$ (asymmetric), SNR $= -11$ dB; $\Sigma L_r = 20$, $R = 4$, $L_r = [5,5,5,5]$ (symmetric), SNR $= -5$ dB. 300 Monte-Carlo trials.

previous steps as $\text{PESDR}_k = \frac{\delta_{J-k}}{\sigma_{J-k-1}}$, the algorithm can detect the first signal eigenvalue (corresponding either to ΣL_r or R) if the prediction error to standard deviation ratio (PESDR) exceeds the predefined threshold ρ for the first time. For a more detailed description of the LaRGE algorithm, we refer the reader to [KDHH21; AKK+22]. Figures 8.3(a) and 8.3(b) show the percentage of correct rank estimations with respect to a threshold ρ for the estimation of ΣL_r and R, respectively. For this simulation, we generated two ($\boldsymbol{\mathcal{X}}_0^{(1)} \in \mathbb{C}^{50\times60\times70}$ and $\boldsymbol{\mathcal{X}}_0^{(2)} \in \mathbb{C}^{52\times60\times70}$) and four coupled BTD tensors ($\boldsymbol{\mathcal{X}}_0^{(1)} \in \mathbb{C}^{50\times60\times70}$, $\boldsymbol{\mathcal{X}}_0^{(2)} \in \mathbb{C}^{52\times60\times70}$, $\boldsymbol{\mathcal{X}}_0^{(3)} \in \mathbb{C}^{54\times60\times70}$, and $\boldsymbol{\mathcal{X}}_0^{(4)} \in \mathbb{C}^{55\times60\times70}$) with the 3-mode in common, and added zero mean circularly symmetric complex Gaussian (ZMCSCG) noise. We consider an asymmetric case with multilinear ranks L_r different for every block ($L_r = [2,3,4]$) and a symmetric case, where the multilinear ranks are equal ($L_r = [5,5,5,5]$). The SNR is -11 dB in the asymmetric case and -5 dB in the symmetric case. For higher SNRs, the percent of correct rank estimations tends to 100 for all ρs in the considered range. As it can be observed, the threshold value does not depend on the number of tensors and is approximately equal to 0.75 for ΣL_r and to 3.5 for R.

8.4 CPD-based computation of the coupled rank-$(L_r, L_r, 1)$ BTD

In this section we describe the next steps of the proposed coupled SECSI-BTD algorithm. They include three main parts, an initial estimation step based on the simulta-

neous matrix diagonalizations (SECSI), clustering, and final refinements. These are an extension of the corresponding SECSI-BTD steps in Chapter 7 to the scenarios with M coupled tensors.

Initial coupled BTD estimates via coupled CPD In order to compute the initial coupled BTD estimates, we employ the SECSI framework for the coupled CPD. The coupled decomposition with rank ΣL_r is performed on the BTD tensors as in (8.2). SECSI utilizes the simultaneous matrix diagonalization (SMD) [DeL06] approach that defines several heuristics to enhance the accuracy of the CP decomposition [RH13; MGK+21]. The main steps of SECSI include the computation of the truncated HOSVD and subsequent estimation of the transformation matrices that represent the connection between the CPD and truncated HOSVD. Assuming that the SVD of unfoldings in the uncoupled modes has been previously computed in the rank estimation step in (8.4), they can be used to calculate the truncated HOSVD without repeating the calculations. Assuming the noiseless case for simplicity, the coupled truncated HOSVD of M tensors $\boldsymbol{\mathcal{X}}_0^{(m)}$ with the third mode in common is written as

$$\boldsymbol{\mathcal{X}}_0^{(m)} = \boldsymbol{\mathcal{S}}^{[\mathrm{s}](m)} \times_1 \boldsymbol{U}_1^{[\mathrm{s}](m)} \times_2 \boldsymbol{U}_2^{[\mathrm{s}](m)} \times_3 \boldsymbol{U}_3^{[\mathrm{s}]}, \tag{8.10}$$

where $\boldsymbol{\mathcal{S}}^{[\mathrm{s}](m)} \in \mathbb{C}^{\Sigma L_r \times \Sigma L_r \times \Sigma L_r}$ are the core tensors, and a set of truncated unitary matrices $\boldsymbol{U}_1^{[\mathrm{s}](m)} \in \mathbb{C}^{I_1^{(m)} \times \Sigma L_r}$, $\boldsymbol{U}_2^{[\mathrm{s}](m)} \in \mathbb{C}^{I_2^{(m)} \times \Sigma L_r}$, and $\boldsymbol{U}_3^{[\mathrm{s}]} \in \mathbb{C}^{I_3 \times \Sigma L_r}$ span the column space of the corresponding n-mode unfoldings of $\boldsymbol{\mathcal{X}}_0^{(m)}$. The 3-mode matrix $\boldsymbol{U}_3^{[\mathrm{s}]}$ is shared for all factorizations and can be calculated from the following SVD (in the noiseless case)

$$\left[\left[\boldsymbol{\mathcal{X}}_0^{(1)}\right]_{(3)}, \ldots, \left[\boldsymbol{\mathcal{X}}_0^{(M)}\right]_{(3)} \right] = \boldsymbol{U}_3^{[\mathrm{s}]} \boldsymbol{\Sigma}_3^{[\mathrm{s}]} \boldsymbol{V}_3^{[\mathrm{s}]\mathrm{H}}. \tag{8.11}$$

Given that these unitary matrices share the same column space as the CPD factors in (8.2), the CPD factor matrices can be expressed through the non-singular transform matrices $\boldsymbol{T}_1^{(m)}$, $\boldsymbol{T}_2^{(m)}$, and $\boldsymbol{T}_3$ as follows

$$\begin{aligned} \boldsymbol{A}^{(m)} = \boldsymbol{U}_1^{[\mathrm{s}](m)} \boldsymbol{T}_1^{(m)}, \quad \boldsymbol{B}^{(m)} = \boldsymbol{U}_2^{[\mathrm{s}](m)} \boldsymbol{T}_2^{(m)}, \\ \boldsymbol{C}^{(\mathrm{r})} = \boldsymbol{U}_3^{[\mathrm{s}]} \boldsymbol{T}_3. \end{aligned} \tag{8.12}$$

The inverses of the transform matrices diagonalize the core tensors $\boldsymbol{\mathcal{S}}^{[\mathrm{s}](m)}$. Thus, if we consider, for instance, a tensor $\boldsymbol{\mathcal{S}}_3^{(m)}$ such that $\boldsymbol{\mathcal{S}}_3^{(m)} = \boldsymbol{\mathcal{S}}^{[\mathrm{s}](m)} \times_3 \boldsymbol{U}_3^{[\mathrm{s}]}$, and a tensor $\boldsymbol{\mathcal{D}}_3 = \boldsymbol{\mathcal{I}}_{3,\Sigma L_r} \times_3 \boldsymbol{C}^{(\mathrm{r})}$, then $\boldsymbol{\mathcal{S}}_3^{(m)} = \boldsymbol{\mathcal{D}}_3 \times_1 \boldsymbol{T}_1^{(m)} \times_2 \boldsymbol{T}_2^{(m)}$, where every ith 3-mode slice of

$\mathcal{D}_3$ is a diagonal matrix $\boldsymbol{D}_{3,i}$ with the ith row of $\boldsymbol{C}^{(\mathrm{r})}$ on its main diagonal. The previous equation can be rewritten as

$$\boldsymbol{S}_{3,i}^{(m)} = \boldsymbol{T}_1^{(m)} \boldsymbol{D}_{3,i} \boldsymbol{T}_2^{(m)\mathrm{T}}, \tag{8.13}$$

where $\boldsymbol{S}_{3,i}^{(m)}$ is the ith 3-mode slice of $\boldsymbol{\mathcal{S}}_3^{(m)}$, $i \in \{1,\ldots,I_3\}$. Equation (8.13) corresponds to a non-symmetric SMD [RH13] (an example illustration of the non-symmetric SMD is given in Figure 7.4 on page 106). By multiplying (8.13) from either the right-hand side (rhs) or the left-hand side (lhs) by an arbitrary pivot slice $(\boldsymbol{S}_{3,p}^{(m)})^{-1}$, $p \in \{1,\ldots,I_3\}$, the asymmetric SMD problem in (8.13) can be transformed to a symmetric one

$$\begin{aligned} \boldsymbol{S}_{3,i}^{(m)\mathrm{rhs}} &= \boldsymbol{S}_{3,i}^{(m)} \left(\boldsymbol{S}_{3,p}^{(m)}\right)^{-1} \\ &= \boldsymbol{T}_1^{(m)} \left(\boldsymbol{D}_{3,i} \boldsymbol{D}_{3,p}{}^{-1}\right) \left(\boldsymbol{T}_1^{(m)}\right)^{-1}, \end{aligned} \tag{8.14}$$

$$\begin{aligned} \boldsymbol{S}_{3,i}^{(m)\mathrm{lhs}} &= \left(\left(\boldsymbol{S}_{3,p}^{(m)}\right)^{-1} \boldsymbol{S}_{3,i}^{(m)}\right)^{\mathrm{T}} \\ &= \boldsymbol{T}_2^{(m)} \left(\boldsymbol{D}_{3,i} \boldsymbol{D}_{3,p}{}^{-1}\right) \left(\boldsymbol{T}_2^{(m)}\right)^{-1}. \end{aligned} \tag{8.15}$$

Consequently, the transform matrices $\boldsymbol{T}_1^{(m)}$ and $\boldsymbol{T}_2^{(m)}$ can be estimated by means of SMD [FG06]. The matrix $\boldsymbol{C}^{(\mathrm{r})}$ can be then retrieved from the jointly diagonalized matrices $\boldsymbol{D}_{3,i}$ up to one scaling ambiguity per column, and for each SMD problem (rhs or lhs), the factors $\boldsymbol{A}^{(m)}$ and $\boldsymbol{B}^{(m)}$ are obtained either from (8.12) or from the least squares (LS) fit as $\boldsymbol{A}^{(m)} = [\boldsymbol{\mathcal{X}}_0^{(m)}]_{(1)} \cdot (\boldsymbol{B}^{(m)} \diamond \boldsymbol{C}^{(\mathrm{r})})^{\mathrm{T}+}$ or $\boldsymbol{B}^{(m)} = [\boldsymbol{\mathcal{X}}_0^{(m)}]_{(2)} \cdot (\boldsymbol{C}^{(\mathrm{r})} \diamond \boldsymbol{A}^{(m)})^{\mathrm{T}+}$. Similar SMD problems can be constructed by multiplying the core tensors $\boldsymbol{\mathcal{S}}^{[\mathrm{s}](m)}$ by $\boldsymbol{U}_1^{[\mathrm{s}](m)}$ or $\boldsymbol{U}_2^{[\mathrm{s}](m)}$ in the 1-mode or the 2-mode, respectively. This, in the CPD case, will lead in total to $2+4M$ SMD problems, and thus $4+2M$ estimates of the third factor matrix and six sets of estimates of non-coupled factor matrices [MGK+21]. Then the final best set of factor estimates can be chosen according to a predefined heuristic [RH13]. However, as we have discussed in Section 7.4.2 of the previous chapter, in contrast to the CPD model data, the rank-$(L_r, L_r, 1)$ BTD tensors always lead to the final estimates from the 3-mode SMDs. Therefore, to estimate the BTD factors by means of SECSI, we only solve the SMDs in (8.13). For more details on the SECSI and the coupled SECSI frameworks, we refer the reader to [RH13; MGK+21].

Clustering The solution of SMDs in (8.14) or (8.15) leads to initial estimates $\hat{\boldsymbol{A}}^{(m)}$, $\hat{\boldsymbol{B}}^{(m)}$, and $\hat{\boldsymbol{C}}^{(\mathrm{r}),(m)}$ in the CPD form. Therefore, in the next step, the matrix $\hat{\boldsymbol{C}}^{(\mathrm{r}),(m)}$ with collinear columns has to be brought to the proper BTD form. This can be done by k-

means clustering [Llo82] (in our simulations, we use the k-means++ algorithm [AD07]). By applying the clustering to $\hat{\boldsymbol{C}}^{(\mathrm{r}),(m)}$, we retrieve permutation matrices $\boldsymbol{P}^{(m)}$ that assign each column in $\hat{\boldsymbol{C}}^{(\mathrm{r}),(m)}$ to one of the R clusters. Additionally, the k-means finds the optimal number of elements in each cluster (L_r). Therefore, it can also be employed in cases when the multilinear ranks are not known. For the cases with a known rank structure, clustering with size constraints can be used [ZWL10].

By multiplying $\hat{\boldsymbol{A}}^{(m)}$, $\hat{\boldsymbol{B}}^{(m)}$, and $\hat{\boldsymbol{C}}^{(\mathrm{r}),(m)}$ by the permutation matrices $\boldsymbol{P}^{(m)}$, we rearrange their columns in a way that the columns corresponding to rth block are grouped together and follow the correct order as in the description of (8.1). These permuted factors are denoted as $\tilde{\boldsymbol{A}}^{(m)}$, $\tilde{\boldsymbol{B}}^{(m)}$, and $\tilde{\boldsymbol{C}}^{(\mathrm{r}),(m)}$.

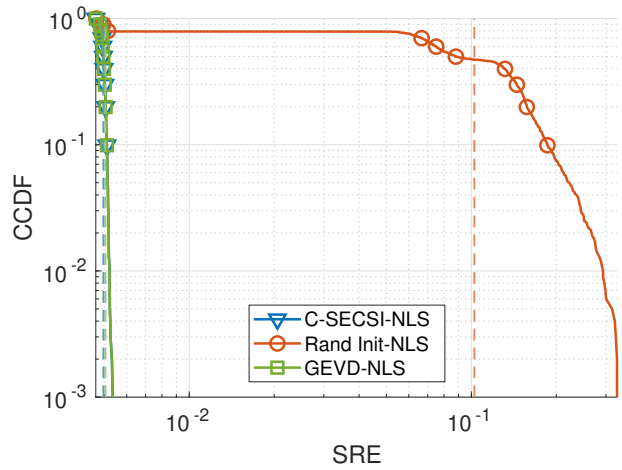
Final refinements To return the final BTD estimates of $\boldsymbol{A}_r^{(m)}$, $\boldsymbol{B}_r^{(m)}$, and $\boldsymbol{c}_r$, the collinear columns in $\tilde{\boldsymbol{C}}^{(\mathrm{r}),(m)}$ have to be reduced. We perform this by using one ALS iteration as follows. First, recalculate $\tilde{\boldsymbol{A}}^{(m)}$ and $\tilde{\boldsymbol{B}}^{(m)}$ with a current $\tilde{\boldsymbol{C}}^{(\mathrm{r}),(m)}$ as $\tilde{\boldsymbol{A}}^{(m)} = [\boldsymbol{\mathcal{X}}^{(m)}]_{(1)} \cdot (\tilde{\boldsymbol{B}}^{(m)} \diamond \tilde{\boldsymbol{C}}^{(\mathrm{r}),(m)})^{\mathrm{T}+}$, and $\tilde{\boldsymbol{B}}^{(m)} = [\boldsymbol{\mathcal{X}}^{(m)}]_{(2)} \cdot (\tilde{\boldsymbol{C}}^{(\mathrm{r}),(m)} \diamond \tilde{\boldsymbol{A}}^{(m)})^{\mathrm{T}+}$. Then the submatrices in $\tilde{\boldsymbol{C}}^{(\mathrm{r}),(m)}$ are reduced to vectors as follows [DN08]

$$\tilde{\boldsymbol{C}}^{(m)} = [\boldsymbol{\mathcal{X}}^{(m)}]_{(3)} \cdot \left[\left(\tilde{\boldsymbol{A}}_1^{(m)} \diamond \tilde{\boldsymbol{B}}_1^{(m)} \right) \mathbf{1}_{L_1}, \cdots, \left(\tilde{\boldsymbol{A}}_R^{(m)} \diamond \tilde{\boldsymbol{B}}_R^{(m)} \right) \mathbf{1}_{L_R} \right]^{\mathrm{T}+}. \tag{8.16}$$

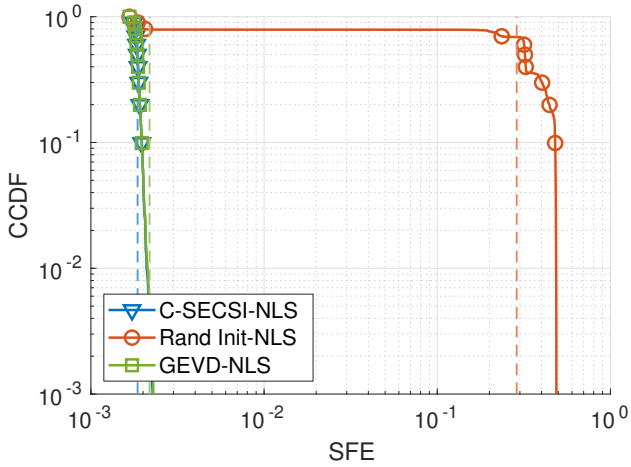
Since there are M similar estimates of $\boldsymbol{C}$, we choose the one with the smallest reconstruction error, and use the obtained $\tilde{\boldsymbol{A}}^{(m)}$, $\tilde{\boldsymbol{B}}^{(m)}$, and $\tilde{\boldsymbol{C}}$ as an initialization of the nonlinear least squares (NLS) algorithm in [SVBD15].

8.5 Numerical results

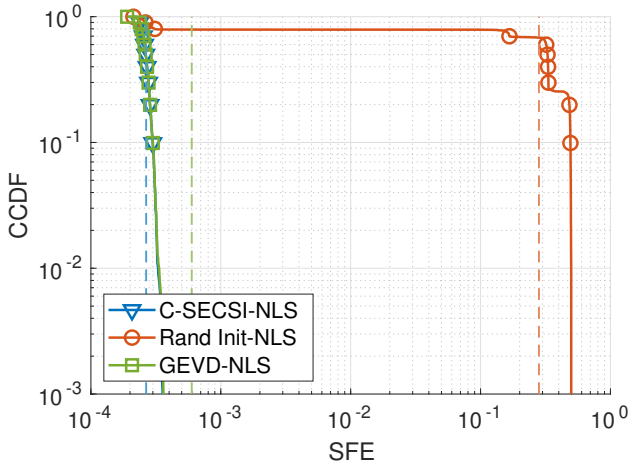
In this section, we compare the performances of the proposed coupled BTD algorithm with the algorithm from the Tensorlab toolbox which is based on the structured data fusion via nonlinear least squares (SDF-NLS) in [SVBD15]. For the simulations, we have generated two coupled BTD tensors $\boldsymbol{\mathcal{X}}_0^{(1)} \in \mathbb{C}^{50\times 60\times 70}$ and $\boldsymbol{\mathcal{X}}_0^{(2)} \in \mathbb{C}^{54\times 60\times 70}$ according to the model in (8.1), and added the noise tensors $\boldsymbol{\mathcal{N}}^{(m)}$ containing ZMCSCG entries with variance σ_n^2. Accordingly, the SNR is defined as $1/\sigma_n^2$. The 3-mode is in common.



(a) CCDF vs. SRE.



(b) CCDF vs. SFE for $\boldsymbol{A}_r^{(m)}\boldsymbol{B}_r^{(m)\mathrm{T}}$.



(c) CCDF vs. SFE for $\boldsymbol{C}$.

Figure 8.4. CCDF vs. Reconstruction errors. SNR = 0 dB. Blue lines denote the proposed algorithm, and the red and green lines denote the Tensorlab [SVBD15; VDS+16] algorithms. Simulation parameters: $R = 3$, $L_r = [2,3,4]$ (asymmetric); 500 Monte-Carlo trials.

We use two metrics to compare the performances, squared reconstruction error (SRE) and squared factor reconstruction error (SFE)

$$\text{SRE} = \frac{1}{M}\sum_{m=1}^{M}\frac{\left\|\hat{\boldsymbol{\mathcal{X}}}_0^{(m)} - \boldsymbol{\mathcal{X}}_0^{(m)}\right\|_{\text{F}}^2}{\left\|\boldsymbol{\mathcal{X}}_0^{(m)}\right\|_{\text{F}}^2}, \tag{8.17}$$

$$\text{SFE}(\boldsymbol{C}) = \frac{\left\|\tilde{\boldsymbol{C}}\cdot\boldsymbol{P}_c - \boldsymbol{C}\right\|_{\text{F}}^2}{\|\boldsymbol{C}\|_{\text{F}}^2}, \tag{8.18}$$

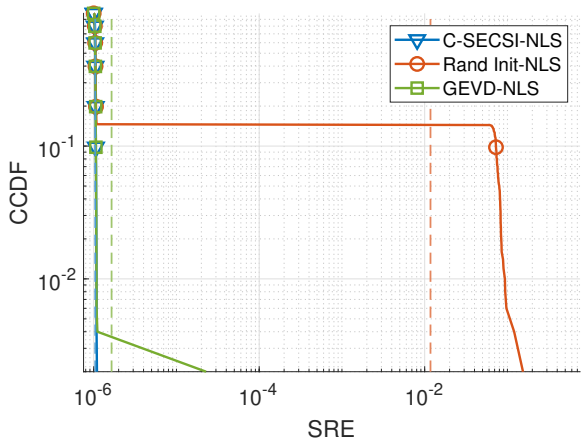
$$\text{SFE}(\boldsymbol{A}\boldsymbol{B}^{\text{T}}) = \frac{1}{RM}\sum_{r=1}^{R}\sum_{m=1}^{M}\frac{\left\|\tilde{\boldsymbol{A}}_r^{(m)}\tilde{\boldsymbol{B}}_r^{(m)\text{T}} - \boldsymbol{A}_r^{(m)}\boldsymbol{B}_r^{(m)\text{T}}\right\|_{\text{F}}^2}{\left\|\boldsymbol{A}_r^{(m)}\boldsymbol{B}_r^{(m)\text{T}}\right\|_{\text{F}}^2}, \tag{8.19}$$

where $\hat{\boldsymbol{\mathcal{X}}}_0$ is a tensor reconstructed from the estimated BTD factors, $(\boldsymbol{A}_r^{(m)}, \boldsymbol{B}_r^{(m)})$, and $\boldsymbol{C}$ are the original and $\tilde{\boldsymbol{A}}_r^{(m)}$, $\tilde{\boldsymbol{B}}_r^{(m)}$, and $\tilde{\boldsymbol{C}}$ are the estimated factors, respectively. The matrix $\boldsymbol{P}_c$ resolves the permutation and scaling. The errors for $\boldsymbol{A}^{(m)}$ and $\boldsymbol{B}^{(m)}$ are shown together due to inherent matrix product ambiguities [DeL08a]. Similarly to the simulations in Section 8.3, we consider an asymmetric case with multilinear ranks L_r different for every block ($L_r = [2,3,4]$) and a symmetric case, where the multilinear ranks are equal ($L_r = [5,5,5,5]$).

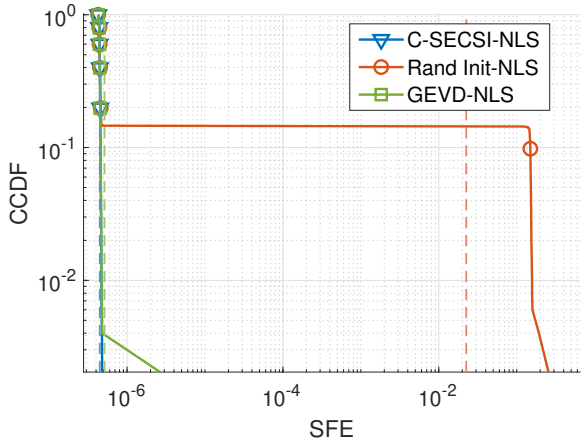
The complementary cumulative distribution functions (CCDFs) of the SRE and the SFE for asymmetric and symmetric scenarios are presented in Figures 8.4 and 8.5, respectively. The errors for $(\boldsymbol{A}_r^{(m)}\boldsymbol{B}_r^{(m)\text{T}})$ are averaged across all values of r. C-SECSI-NLS corresponds to the proposed coupled SECSI-BTD approach with NLS refinement, and GEVD-NLS denotes the generalized eigenvalue decomposition (GEVD)-based initialization and NLS refinement. "Rand Init-NLS" denotes the NLS with random initialization. As we can see, the proposed coupled SECSI-BTD algorithm outperforms both the GEVD-NLS and randomly initialized NLS algorithms. It can be explained by the fact that in contrast to the GEVD in [VDS+16], the coupled SECSI framework uses an SMD-based approach that takes all the slices of a tensor into account and chooses the best estimate instead of considering only one pencil in GEVD, which makes its performance more stable.

8.6 Conclusions

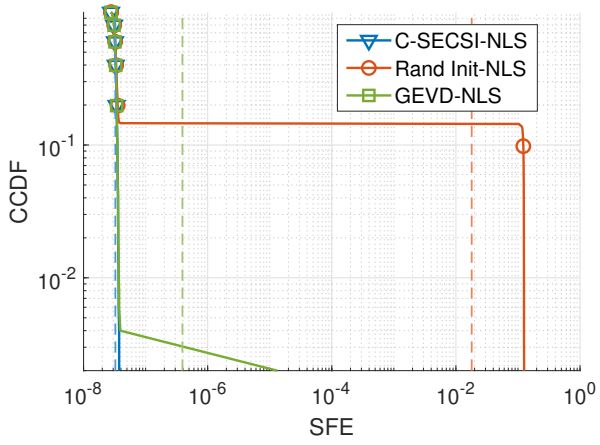
In this chapter of the thesis, we have introduced the coupled SECSI-BTD algorithm to estimate the factors of the coupled rank-$(L_r, L_r, 1)$ BTD of multiple tensors with a



(a) CCDF vs. SRE.



(b) CCDF vs. SFE for $\boldsymbol{A}_r^{(m)}\boldsymbol{B}_r^{(m)\mathrm{T}}$.



(c) CCDF vs. SFE for $\boldsymbol{C}$.

Figure 8.5. CCDF vs. Reconstruction errors. SNR = 40 dB. Blue lines denote the proposed algorithm, and the red and green lines denote the Tensorlab [SVBD15; VDS+16] algorithms. Simulation parameters: $R = 4$, $L_r = [5,5,5,5]$ (symmetric). 1000 Monte-Carlo trials.

common mode based on the coupled extension of the semi-algebraic framework for approximate CP decompositions via simultaneous matrix diagonalizations (SECSI). The approach is an expansion of the algorithm proposed in Chapter 7 to factorize a set of tensors with common mode(s). We have exploited the joint HOSVD of multiple tensors and the link between the HOSVD, the CPD and the BTD to estimate the common and individual factors of the coupled rank-$(L_r, L_r, 1)$ decomposition. Additionally, we have shown how the multilinear rank structure of the coupled rank-$(L_r, L_r, 1)$ decomposition can be estimated based on the extension of the LaRGE scheme and clustering techniques. The numerical results demonstrate that compared to the GEVD-based approach in [VDS+16; SVBD15], the proposed SECSI-based initialization guarantees convergence, and therefore is more accurate and numerically stable. Moreover, in contrast to [VDS+16; SVBD15], our approach allows performing the decomposition even if the ranks are not available beforehand. Therefore it can be further used in real data applications with unknown rank structure, for instance, in EEG-MEG data processing, since these signals are usually recorded simultaneously and thus exhibit coupling.

Tensor-based Near-field Localization in MIMO Radar Systems

9

This chapter presents a high-resolution coupled rank-$(L_r, L_r, 1)$ block-term decomposition-based near-field localization scheme for multi-static MIMO radar systems. The proposed COBRAS (COupled Block-term decomposition for multi-static RAdar Systems) algorithm uses the exact wavefront model to estimate the target location parameters in 3D space and can be applied for arbitrary array geometries. Compared to the far-field models, the exact near-field wavefront model allows exploiting the distance information for high-accuracy positioning. Moreover, we consider a system with massive antenna arrays, which increases the Fresnel region and expands the range of the near-field assumption. The COBRAS algorithm includes the initial tensor decomposition of the data and further post-processing steps that allow extracting the location parameters. Additionally, we compare the performance of different rank-$(L_r, L_r, 1)$ decomposition algorithms (including the algorithms proposed in Chapters 7 and 8) and demonstrate how the employment of coupling improves the localization performance compared to the non-coupled solutions.

9.1 Introduction

The near-field region provides new opportunities and challenges for communication and sensing applications. Compared to the far-field design, the near-field allows exploiting the distance information. Therefore, near-field beamforming can yield a higher spatial multiplexing gain by transmitting different data to users located at different locations in the same direction. Moreover, the near-field is highly beneficial in radar systems: in contrast to far-field systems, it allows a more accurate localization with not only angle but also distance parameters. Until recently, most applications of wireless systems have focused on the far-field assumption. However, operations in the mmWave and THz bands, the use of large antenna arrays, and the deployment of reconfigurable intelligent surfaces make the exploitation of the near-field essential and inevitable [CWL+23]: the near-field region increases, and the wavefronts cannot be considered as planar anymore.

Commonly, publications that focus on near-field signal processing consider the Fresnel approximation of the spherical wavefronts [HCS96; EP12; SWC16; SWC17a; ZJQ15; ZXO+19], which uses a Taylor expansion of the true phase distribution of the wavefront across antennas. Alternatively, some authors [SWC17b; SWC16; PHT+18; PSK+21] avoid using the Fresnel approximation and utilize the exact spherical wavefront model, which avoids systematic errors introduced by the Fresnel approximation and improves the estimation accuracy. Other tools that demonstrate promising results in the area of multiple-input multiple-output (MIMO) radar signal processing include tensor decompositions and multidimensional signal processing. For example, in [PSK+21], we propose a canonical polyadic (CP) decomposition-based localization and channel estimation algorithm for massive antenna arrays. The studies in [SWC17b; SWC16], and our paper [PHT+18] employ the CP decomposition for parameter estimation in bistatic MIMO radar systems. The multi-static MIMO radar scenario is considered in [YGL+19; CZYG19]. The authors propose to use the coupled CP and coupled rank-$(L_r, L_r, 1)$ decompositions to localize the targets in the far-field. However, the algorithm suggested in [YGL+19] is limited to 2D space localization when considering a uniform rectangular array (URA) antenna configuration due to the chosen wavefront model. Tensor-based techniques are also widely employed for direction-of-arrival estimation using a planar wavefront model [SBG00; SDD18; LLN+20].

Main contributions. This chapter of the thesis focuses on multi-static MIMO radar systems with multiple receive and transmit massive antenna arrays. We consider the parameter estimation of multiple targets in 3D space using the exact spherical wavefront model. Moreover, we use the signal model from [YGL+19] to arrange the received data as a low-rank tensor and employ a robust coupled block-term decomposition (BTD) algorithm that we presented in [SKKH22] and in Chapter 8 to estimate the steering matrices. We also develop appropriate post-processing steps, such as phase unwrapping, the solution of the system of linear equations, and the parameter extraction to obtain the final parameter estimates. We refer to the proposed algorithm as "COBRAS" (COupled Block-term decomposition for multi-static RAdar Systems). In contrast to the algorithm in [YGL+19], our approach is designed for near-field systems and uses the exact spherical wavefront model. Additionally, we compare the performance of the different BTD algorithms in terms of their impact on localization accuracy.

The remainder of this chapter is organized as follows: Section 9.2 provides a system model. Section 9.3 outlines the main steps of the proposed coupled rank-$(L_r, L_r, 1)$ decomposition-based near-field localization algorithm. Section 9.4 presents our experimental setup, simulation results, and a comparison of the impact of different BTD

algorithms on localization performance. Finally, Section 9.5 concludes the study by summarizing the key findings.

9.2 System model

We consider a multi-static multi-pulse MIMO radar system that uses M_{T} transmit and M_{R} receive arrays to localize R targets. Each transmitter emits temporally orthogonal signals that impinge on the targets, and their reflections are intercepted by receiving arrays. At the receiver side, the signals are matched-filtered using the known waveforms from the transmitter side. The targets are assumed to be in the near-field region[1] of both arrays, such that the wavefronts of the impinging and reflecting waves can be considered spherical, and the target radar cross section (RCS) is assumed to be uncorrelated from pulse to pulse (Swerling 2 model [RHS10]). In the sequel, we focus on uniform rectangular arrays and describe a high-resolution algorithm to estimate the range, azimuth, and elevation parameters of the dominant wavefronts in 3D space. However, the COBRAS algorithm is also applicable to arrays of arbitrary (known) geometries.

9.2.1 Scenario

An example scenario is shown in Figure 9.1. The antennas in the transmit and receive arrays are depicted with red and blue dots, respectively. The black dots denote targets. Each transmit and receive array consists of $N_{\mathrm{T}}^{[m_{\mathrm{T}}]}$, $m_{\mathrm{T}} \in \{1,\dots,M_{\mathrm{T}}\}$ and $N_{\mathrm{R}}^{[m_{\mathrm{R}}]}$, $m_{\mathrm{R}} \in \{1,\dots,M_{\mathrm{R}}\}$ antennas, respectively. In the following, to ease the notation, we skip the superscripts $[m_{\mathrm{T}}]$ and $[m_{\mathrm{R}}]$ in the notations of N_{T} and N_{R}, and assume an equal number of antennas in all transmit and all receive arrays. However, the algorithm supports scenarios with different numbers of elements in the antenna arrays. We denote the Cartesian coordinates of the transmit antennas as $\left\{x_{\mathrm{T},n_T}^{[m_{\mathrm{T}}]}, y_{\mathrm{T},n_T}^{[m_{\mathrm{T}}]}, z_{\mathrm{T},n_T}^{[m_{\mathrm{T}}]}\right\}$, $n_{\mathrm{T}} \in \{1,\dots,N_{\mathrm{T}}\}$ and the Cartesian coordinates of the receive antennas as $\left\{x_{\mathrm{R},n_R}^{[m_{\mathrm{R}}]}, y_{\mathrm{R},n_R}^{[m_{\mathrm{R}}]}, z_{\mathrm{R},n_R}^{[m_{\mathrm{R}}]}\right\}$, $n_{\mathrm{R}} \in \{1,\dots,N_{\mathrm{R}}\}$.

[1] It is also refered to as the Fresnel region r_F, which can be expressed as [BEO16] $0.62\left(\frac{d^3}{\lambda}\right)^{\frac{1}{2}} < r_F < \frac{2d^2}{\lambda}$, where d is the maximum diameter of the antenna aperture, and λ is a wavelength. As an example, for the antenna array of size 10×10 with inter-element spacing $\lambda/2$ the Fresnel region is approximately 10λ to 81λ; and for the 25×25 antenna array, it is approximately 43λ to 576λ.

Every rth target in 3D space (see Appendix A.6) is characterized by a set of parameters

$$\boldsymbol{\Theta}_{\mathrm{T},r}^{[m_\mathrm{T}]} = \left\{\rho_{\mathrm{T},r}^{[m_\mathrm{T}]}, \phi_{\mathrm{T},r}^{[m_\mathrm{T}]}, \theta_{\mathrm{T},r}^{[m_\mathrm{T}]}\right\} \tag{9.1}$$

and

$$\boldsymbol{\Theta}_{\mathrm{R},r}^{[m_\mathrm{R}]} = \left\{\rho_{\mathrm{R},r}^{[m_\mathrm{R}]}, \phi_{\mathrm{R},r}^{[m_\mathrm{R}]}, \theta_{\mathrm{R},r}^{[m_\mathrm{R}]}\right\}, \tag{9.2}$$

where $\rho_{\mathrm{T},r}^{[m_\mathrm{T}]}$ and $\rho_{\mathrm{R},r}^{[m_\mathrm{R}]}$ are the distances from the rth target to the reference transmit and receive antennas of the transmit and receive arrays, respectively, and $r \in \{1, \ldots, R\}$. Moreover, $\phi_{\mathrm{T},r}^{[m_\mathrm{T}]}$ and $\phi_{\mathrm{R},r}^{[m_\mathrm{R}]}$ are the azimuth, and $\theta_{\mathrm{T},r}^{[m_\mathrm{T}]}$ and $\theta_{\mathrm{R},r}^{[m_\mathrm{R}]}$ are the elevation angles, defined with respect to the transmit and the receive systems of coordinates. We assume that the reference antennas of the transmit and receive arrays are located at the origin of the "local" system of coordinates (at every array). Furthermore, we introduce a "global" system of coordinates with the origin at the reference antenna of the first transmit array. Then the coordinates $\{x_{\mathrm{T},m_\mathrm{T}}, y_{\mathrm{T},m_\mathrm{T}}, z_{\mathrm{T},m_\mathrm{T}}\}$ and $\{x_{\mathrm{R},m_\mathrm{R}}, y_{\mathrm{R},m_\mathrm{R}}, z_{\mathrm{R},m_\mathrm{R}}\}$ denote the locations of the m_Tth and m_Rth reference antennas with respect to the global origin. Additionally, we introduce a set of Cartesian coordinates of the rth target as $\boldsymbol{\Theta}_{\mathrm{T},r}^{[m_\mathrm{T}]} = \left\{x_{\mathrm{T},r}^{[m_\mathrm{T}]}, y_{\mathrm{T},r}^{[m_\mathrm{T}]}, z_{\mathrm{T},r}^{[m_\mathrm{T}]}\right\}$ and $\boldsymbol{\Theta}_{\mathrm{R},r}^{[m_\mathrm{R}]} = \left\{x_{\mathrm{R},r}^{[m_\mathrm{R}]}, y_{\mathrm{R},r}^{[m_\mathrm{R}]}, z_{\mathrm{R},r}^{[m_\mathrm{R}]}\right\}$ defined with respect to the reference antennas at the transmit and receive arrays, respectively.

9.2.2 Signal model

The received signal $\boldsymbol{Y}_p^{[m_\mathrm{R}]} \in \mathbb{C}^{N_\mathrm{R} \times T}$ at the m_Rth receiving array in the pth pulse can be written as [YGL+19]

$$\boldsymbol{Y}_p^{[m_\mathrm{R}]} = \sum_{m_\mathrm{T}=1}^{M_\mathrm{T}} \sum_{r=1}^{R} \alpha_r^{[m_\mathrm{R}]} \varrho_{r,p}^{[m_R, m_T]} \boldsymbol{b}_r^{[m_\mathrm{R}]} \boldsymbol{a}_r^{[m_\mathrm{T}]\mathrm{T}} \boldsymbol{S}^{[m_\mathrm{T}]\mathrm{T}} + \boldsymbol{N}_p^{[m_\mathrm{R}]}, \tag{9.3}$$

where m_T, m_R, r, and p are the transmit array, the receive array, the target, and the pulse indices. The complex gain parameters $\alpha_r^{[m_\mathrm{R}]}$ represent such effects as attenuation and random phase shifts. The complex quantity $\varrho_{r,p}^{[m_R, m_T]}$ is the reflection coefficient of the rth target relative to the m_Rth receive and m_Tth transmit array in the pth pulse. The matrix $\boldsymbol{S}^{[m_\mathrm{T}]} \in \mathbb{C}^{T \times N_\mathrm{T}}$ contains the transmitted signal during one pulse period after matched-filtering at the receiver, where T is the number of time snapshots in each pulse. The matrix $\boldsymbol{N}_p^{[m_\mathrm{R}]} \in \mathbb{C}^{N_\mathrm{R} \times T}$ represents the independently and identically distributed (i.i.d.) zero mean spatially and temporally white additive noise

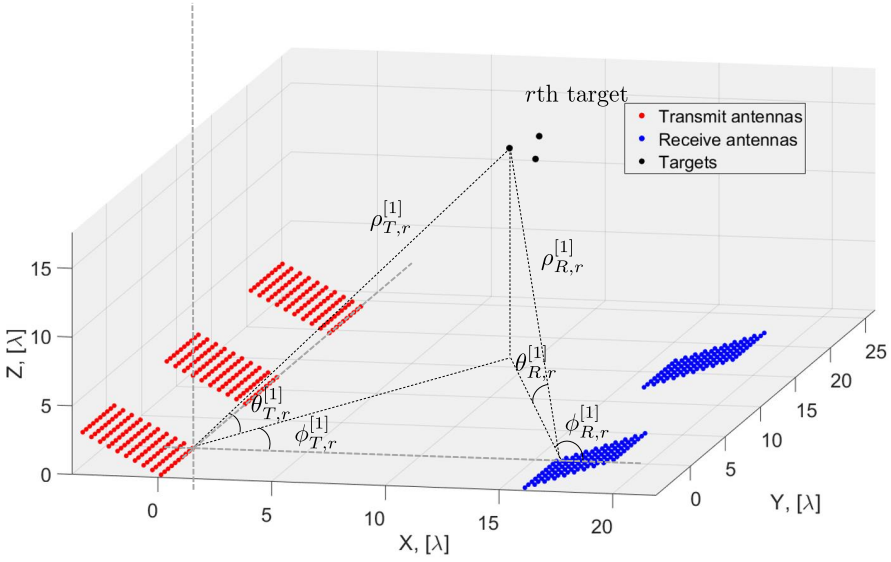


Figure 9.1. Scenario geometry with $R = 3$ targets, $M_{\mathrm{T}} = 3$ transmit URAs, and $M_{\mathrm{R}} = 2$ receive URAs. The parameters of one of the targets are given with respect to transmit and receive reference antennas of the first transmit and the first receive arrays.

with variance σ_n^2. Moreover, the transmit and receive array steering vectors for the rth target are defined in terms of the path differences as

$$\boldsymbol{a}_r^{[m_{\mathrm{T}}]} = \left[\ 1,\quad \mathrm{e}^{-j\frac{2\pi}{\lambda}\delta^{[m_{\mathrm{T}}]}_{\mathrm{T},r,2}},\quad \ldots,\quad \mathrm{e}^{-j\frac{2\pi}{\lambda}\delta^{[m_{\mathrm{T}}]}_{\mathrm{T},r,N_{\mathrm{T}}}}\ \right]^{\mathrm{T}} \in \mathbb{C}^{N_{\mathrm{T}}} \tag{9.4}$$

and

$$\boldsymbol{b}_r^{[m_{\mathrm{R}}]} = \left[\ 1,\quad ,\mathrm{e}^{-j\frac{2\pi}{\lambda}\delta^{[m_{\mathrm{R}}]}_{\mathrm{R},r,2}},\quad \ldots,\quad \mathrm{e}^{-j\frac{2\pi}{\lambda}\delta^{[m_{\mathrm{R}}]}_{\mathrm{R},r,N_{\mathrm{R}}}}\ \right]^{\mathrm{T}} \in \mathbb{C}^{N_{\mathrm{R}}}, \tag{9.5}$$

respectively.

The exact expression for the geometric path differences $\delta^{[m_{\mathrm{T}}]}_{\mathrm{T},r,n_{\mathrm{T}}}$ and $\delta^{[m_{\mathrm{R}}]}_{\mathrm{R},r,n_{\mathrm{R}}}$ between the reference antenna and the n_{T}th (n_{R}th) antenna in the array depends on the chosen wavefront model, which can be planar if the sources are in the far-field [SN89], approximated spherical (Fresnel approximation [HCS96]) or exactly spherical [PHT+18; SWC16; PSK+21; KPH21]. In this work, we adopt the exact spherical wavefront model given as

$$\delta^{[m_{\mathrm{T}}]}_{\mathrm{T},r,n_{\mathrm{T}}} = \sqrt{\left(x^{[m_{\mathrm{T}}]}_{\mathrm{T},r} - x^{[m_{\mathrm{T}}]}_{\mathrm{T},n_T}\right)^2 + \left(y^{[m_{\mathrm{T}}]}_{\mathrm{T},r} - y^{[m_{\mathrm{T}}]}_{\mathrm{T},n_T}\right)^2 + \left(z^{[m_{\mathrm{T}}]}_{\mathrm{T},r} - z^{[m_{\mathrm{T}}]}_{\mathrm{T},n_T}\right)^2} - \rho^{[m_{\mathrm{T}}]}_{\mathrm{T},r},\ \ \forall n_{\mathrm{T}}, \forall m_{\mathrm{T}}, \tag{9.6}$$

$$\delta^{[m_{\mathrm{R}}]}_{\mathrm{R},r,n_{\mathrm{R}}} = \sqrt{\left(x^{[m_{\mathrm{R}}]}_{\mathrm{R},r} - x^{[m_{\mathrm{R}}]}_{\mathrm{R},n_R}\right)^2 + \left(y^{[m_{\mathrm{R}}]}_{\mathrm{R},r} - y^{[m_{\mathrm{R}}]}_{\mathrm{R},n_R}\right)^2 + \left(z^{[m_{\mathrm{R}}]}_{\mathrm{R},r} - z^{[m_{\mathrm{R}}]}_{\mathrm{R},n_R}\right)^2} - \rho^{[m_{\mathrm{R}}]}_{\mathrm{R},r},\ \ \forall n_{\mathrm{R}}, \forall m_{\mathrm{R}}. \tag{9.7}$$

By stacking the matrices $\boldsymbol{Y}_p^{[m_\mathrm{R}]}$ with fixed m_R and varying p along the 3-mode, the data received at the m_Rth array for P pulses can be represented as a three-dimensional tensor $\boldsymbol{\mathcal{Y}}'^{[m_\mathrm{R}]}_{(:,:,p)} = \boldsymbol{Y}_p^{[m_\mathrm{R}]} \in \mathbb{C}^{N_\mathrm{R}\times T}$. After permuting the first and the third indices in $\boldsymbol{\mathcal{Y}}'^{[m_\mathrm{R}]} \in \mathbb{C}^{N_\mathrm{R}\times T\times P}$, we can rewrite it as the tensor $\boldsymbol{\mathcal{Y}}^{[m_\mathrm{R}]} \in \mathbb{C}^{P\times T\times N_\mathrm{R}}$ (we refer the reader to Appendix A.7 and to [YGL+19] for detailed derivations) in the following way

$$\boldsymbol{\mathcal{Y}}^{[m_\mathrm{R}]} = \sum_{r=1}^{R} \alpha_r^{[m_\mathrm{R}]} \left(\boldsymbol{F}_r^{[m_\mathrm{R}]} \boldsymbol{C}_r^\mathrm{T}\right) \circ \boldsymbol{b}_r^{[m_\mathrm{R}]} + \boldsymbol{\mathcal{N}}^{[m_\mathrm{R}]} \in \mathbb{C}^{P\times T\times N_\mathrm{R}}, \tag{9.8}$$

where the columns of the matrix of reflection coefficients

$$\boldsymbol{F}_r^{[m_\mathrm{R}]} = \left[\begin{array}{ccc} \boldsymbol{f}_r^{[m_R,1]}, & \ldots, & \boldsymbol{f}_r^{[m_R,M_T]} \end{array}\right] \in \mathbb{C}^{P\times M_\mathrm{T}} \tag{9.9}$$

are expressed as

$$\boldsymbol{f}_r^{[m_R,m_T]} = \left[\begin{array}{ccc} \varrho_{r,1}^{[m_R,m_T]}, & \ldots, & \varrho_{r,P}^{[m_R,m_T]} \end{array}\right]^\mathrm{T} \in \mathbb{C}^P. \tag{9.10}$$

Moreover, the matrix

$$\boldsymbol{C}_r = \left[\begin{array}{ccc} \boldsymbol{c}_r^{[1]}, & \ldots, & \boldsymbol{c}_r^{[M_\mathrm{T}]} \end{array}\right] \in \mathbb{C}^{T\times M_\mathrm{T}} \tag{9.11}$$

is constructed from the vectors $\boldsymbol{c}_r^{[m_\mathrm{T}]} = \boldsymbol{S}^{[m_\mathrm{T}]} \boldsymbol{a}_r^{[m_\mathrm{T}]} \in \mathbb{C}^T$ that combine the transmitted signal and the transmit steering vectors. The representation of the data in (9.8) corresponds to the noise-corrupted coupled rank-$(L_r, L_r, 1)$ block-term decomposition [YGL+19; SD15; SKKH22] with the M_R tensors coupled in the 2-mode since the arrays receive the same transmitted data. We assume that the matrices $\boldsymbol{F}_r^{[m_\mathrm{R}]} \in \mathbb{C}^{P\times M_\mathrm{T}}$ and $\boldsymbol{C}_r \in \mathbb{C}^{T\times M_\mathrm{T}}$ in (9.8) have a full column rank equal to M_T, i.e., the number of pulses and snapshots is greater than M_T. In this case, the rank-$(L_r, L_r, 1)$ BTD is identifiable [DeL08a] and the matrices in (9.8) can be estimated through the coupled rank-$(L_r, L_r, 1)$ decomposition. Then, the final target parameter estimates are obtained via the solution of a system of linear equations and further processing, which we cover in the following.

9.3 Coupled BTD-based target localization

This section describes the main steps of the proposed COBRAS algorithm that utilizes the coupled rank-$(L_r, L_r, 1)$ decomposition for target localization in multi-static near-field radar systems. It includes the estimation of the initial decomposition of the

received data, phase unwrapping, the solution of the system of linear equations, and the parameter extraction to calculate the final estimates.

9.3.1 Tensor decomposition and phase unwrapping

The first step of the algorithm is an estimation of the matrices $\boldsymbol{F}_r^{[m_\mathrm{R}]}$, $\boldsymbol{C}_r$, and vectors $\boldsymbol{b}_r^{[m_\mathrm{R}]}$ which can be performed via an approximate coupled rank-$(L_r, L_r, 1)$ decomposition of the tensors $\boldsymbol{\mathcal{Y}}^{[m_\mathrm{R}]}$ with the block-rank R and the L_r-ranks equal to M_T. Several authors have proposed algorithms to compute the coupled rank-$(L_r, L_r, 1)$ decomposition, for example, the coupled SECSI-BTD algorithm that we propose in [SKKH22] and in Chapter 8, structured data fusion algorithms in [SVBD15], or simultaneous diagonalization (SD) methods in [SD15; SDD15; SVBD15]. In this study, we focus on the solution from Chapter 8 and compare its performance to the other schemes.

At this point, we should mention the additional matrix product ambiguity that occurs in the rank-$(L_r, L_r, 1)$ decompositions when estimating the product $\left(\boldsymbol{F}_r^{[m_\mathrm{R}]}\boldsymbol{C}_r^\mathrm{T}\right)$ as shown in [DeL08a] (see also (7.5) in Section 7.2). Since the ambiguities in $\left(\boldsymbol{F}_r^{[m_\mathrm{R}]}\boldsymbol{C}_r^\mathrm{T}\right)$ are difficult to resolve, in this chapter, we focus on the estimation of the steering vectors $\boldsymbol{b}_r^{[m_\mathrm{R}]}$ at the receive arrays, which are essentially unique (up to an arbitrary permutation and scaling) [SD15; DeL08a] and can be used for the localization of the targets. We denote the estimated vectors $\boldsymbol{b}_r^{[m_\mathrm{R}]}$ as $\hat{\boldsymbol{b}}_r^{[m_\mathrm{R}]}$ and use them as columns in the matrices $\hat{\boldsymbol{B}}^{[m_\mathrm{R}]} = [\hat{\boldsymbol{b}}_1^{[m_\mathrm{R}]}, \ldots, \hat{\boldsymbol{b}}_R^{[m_\mathrm{R}]}] \in \mathbb{C}^{N_\mathrm{R} \times R}$. Based on the knowledge that at the reference (first) antenna, the path difference is equal to zero, the BTD scaling ambiguity can be corrected by dividing the vectors $\hat{\boldsymbol{b}}_r^{[m_\mathrm{R}]}$ by their first element.

As described in [PHT+18], to acquire the correct phases of the steering vectors $\hat{\boldsymbol{b}}_r^{[m_\mathrm{R}]}$, the unwrapping procedure has to be performed on the extracted phase angles as

$$\hat{\boldsymbol{\delta}}_r^{[m_\mathrm{R}]} = \frac{\lambda}{2\pi} \cdot \mathcal{U}\left\{\angle \hat{\boldsymbol{b}}_r^{[m_\mathrm{R}]}\right\}, \forall r, \tag{9.12}$$

where $\hat{\boldsymbol{\delta}}_r^{[m_\mathrm{R}]}$ is the vector of path difference estimates and $\mathcal{U}\{\bullet\}$ denotes the unwrapping algorithm. For uniformly distributed geometries, such as Uniform Linear Arrays (ULAs) or URAs, the correct phase unwrapping is possible if the element spacing is not larger than $\lambda/2$ with a simple 1-dimensional unwrapping algorithm [Ito82] (in case of a URA it should be applied row- and column-wise). For other types of array geometries, 2- (or 3-) dimensional unwrapping algorithms should be applied [PHT+18; GP98].

9.3.2 Solving the system of linear equations

In order to calculate the target location parameters, the following post-processing steps are applied independently for every $\hat{\boldsymbol{\delta}}_r^{[m_\mathrm{R}]}$, $\forall r$, $\forall m_\mathrm{R}$. Since the processing steps are the same for every receiving array, for notational simplicity, we drop the superscript $[m_\mathrm{R}]$ and the subscript R, denoting the array number and the receive side, respectively. Substituting the estimated path difference $\hat{\delta}_{r,n_\mathrm{R}}$ into (9.7), moving ρ_r to the left side, and taking the square results in

$$\left(\hat{\delta}_{r,n_\mathrm{R}} + \rho_r\right)^2 = \left(x_r - x_{n_\mathrm{R}}\right)^2 + \left(y_r - y_{n_\mathrm{R}}\right)^2 + \left(z_r - z_{n_\mathrm{R}}\right)^2. \tag{9.13}$$

By expanding (9.13) further, we get

$$\begin{aligned}\hat{\delta}_{r,n_\mathrm{R}}^2 + 2\rho_r\hat{\delta}_{r,n_\mathrm{R}} + \rho_r^2 = \underbrace{x_r^2 + y_r^2 + z_r^2}_{\rho_r^2} + \underbrace{x_{n_\mathrm{R}}^2 + y_{n_\mathrm{R}}^2 + z_{n_\mathrm{R}}^2}_{d_{n_\mathrm{R}}^2} \\ -2\left(x_r x_{n_\mathrm{R}} + y_r y_{n_\mathrm{R}} + z_r z_{n_\mathrm{R}}\right).\end{aligned} \tag{9.14}$$

which results in a set of $N_\mathrm{R} - 1$ linear equations (since the reference antennas are located at the origin of the system of coordinates, $x_1 = y_1 = z_1 = 0$, and $\hat{\delta}_{r,1} = 0$, $\forall r$) [PHT+18]

$$2\underbrace{\begin{bmatrix} x_2 & y_2 & z_2 & \hat{\delta}_{r,2} \\ x_3 & y_3 & z_3 & \hat{\delta}_{r,3} \\ \vdots & \vdots & \vdots & \vdots \\ x_{N_\mathrm{R}-1} & y_{N_\mathrm{R}-1} & z_{N_\mathrm{R}-1} & \hat{\delta}_{r,N_\mathrm{R}-1} \end{bmatrix}}_{\boldsymbol{X}_r \in \mathbb{R}^{(N_\mathrm{R}-1)\times 4}} \boldsymbol{p}_r = \underbrace{\begin{bmatrix} d_2^2 - \hat{\delta}_{r,2}^2 \\ d_3^2 - \hat{\delta}_{r,3}^2 \\ \vdots \\ d_{N_\mathrm{R}-1}^2 - \hat{\delta}_{r,N_\mathrm{R}-1}^2 \end{bmatrix}}_{\boldsymbol{y}_r \in \mathbb{R}^{(N_\mathrm{R}-1)\times 1}} \tag{9.15}$$

where $d_{n_\mathrm{R}}^2 = x_{n_\mathrm{R}}^2 + y_{n_\mathrm{R}}^2 + z_{n_\mathrm{R}}^2$ is the distance from the reference antenna (located at the origin) to the n_Rth antenna, and

$$\boldsymbol{p}_r = \begin{bmatrix} x_r \\ y_r \\ z_r \\ \sqrt{x_r^2 + y_r^2 + z_r^2} \end{bmatrix} = \begin{bmatrix} \rho_r \cos(\phi_r)\cos(\theta_r) \\ \rho_r \sin(\phi_r)\cos(\theta_r) \\ \rho_r \sin(\theta_r) \\ \rho_r \end{bmatrix}. \tag{9.16}$$

The set of equations in (9.15) allows estimating the location of targets in the full 3D space around the receive array if the antenna array geometry exhibits diversity in all spatial directions (i.e., the arrays are not planar). In the case of the planar antenna arrays, for example, URAs, the matrix $\boldsymbol{X}_r$ in (9.15) will not have full column rank.

In such a situation, we choose a local system of coordinates with the origin at the reference antenna such that all antennas lie in the x-y-plane. Then, $z_{n_\mathrm{R}} = 0$, $\forall n_\mathrm{R}$, and equations (9.15) and (9.16) reduce to

$$2 \underbrace{\begin{bmatrix} x_2 & y_2 & \hat{\delta}_{r,2} \\ x_3 & y_3 & \hat{\delta}_{r,3} \\ \vdots & \vdots & \vdots \\ x_{N_\mathrm{R}-1} & y_{N_\mathrm{R}-1} & \hat{\delta}_{r,N_\mathrm{R}-1} \end{bmatrix}}_{\boldsymbol{X}_r \in \mathbb{R}^{(N_\mathrm{R}-1)\times 3}} \boldsymbol{p}_r = \underbrace{\begin{bmatrix} d_2^2 - \hat{\delta}_{r,2}^2 \\ d_3^2 - \hat{\delta}_{r,3}^2 \\ \vdots \\ d_{N_\mathrm{R}-1}^2 - \hat{\delta}_{r,N_\mathrm{R}-1}^2 \end{bmatrix}}_{\boldsymbol{y}_r \in \mathbb{R}^{(N_\mathrm{R}-1)\times 1}} \tag{9.17}$$

and

$$\boldsymbol{p}_r = \begin{bmatrix} x_r \\ y_r \\ \sqrt{x_r^2 + y_r^2 + z_r^2} \end{bmatrix} = \begin{bmatrix} \rho_r \cos(\phi_r)\cos(\theta_r) \\ \rho_r \sin(\phi_r)\cos(\theta_r) \\ \rho_r \end{bmatrix}, \tag{9.18}$$

respectively. As can be seen from (9.18), in such a case, the parameter estimation is only possible in the upper half of the 3D space where $z_r > 0$ (or $0 \leq \theta_r \leq \pi/2$).

The solution vector can, for example, be found by means of least squares (LS) or total least squares (TLS). In the latter case, the solution vector $\boldsymbol{p}_r$ is found as

$$\hat{\boldsymbol{p}}_r = \frac{(\boldsymbol{v}_r)_{(1:q-1)}}{(\boldsymbol{v}_r)_{(q)}} \tag{9.19}$$

where q is the number of elements in $\boldsymbol{v}_r$. We use $(\boldsymbol{a})_{(i)}$ to denote the ith element of $\boldsymbol{a}$. The vector $\boldsymbol{v}_r$ is the right-singular vector of the augmented matrix $\begin{bmatrix} \boldsymbol{X}_r & \boldsymbol{y}_r \end{bmatrix}$ that corresponds to the smallest singular value [SWC17b; GVL13]. However, according to our simulations, the LS solution performs similarly to the TLS solution.

9.3.3 Parameter extraction

In the general 3D case, the final range, azimuth, and elevation estimates can be computed as

$$\rho_r = \left(\hat{\boldsymbol{p}}_r\right)_{(4)}, \tag{9.20a}$$

$$\phi_r = \operatorname{atan2}\left(\frac{\left(\hat{\boldsymbol{p}}_r\right)_{(2)}}{\rho_r}, \frac{\left(\hat{\boldsymbol{p}}_r\right)_{(1)}}{\rho_r}\right), \tag{9.20b}$$

and

$$\theta_r = \operatorname{atan2}\left(\frac{\left(\hat{\boldsymbol{p}}_r\right)_{(3)}}{\rho_r}, \frac{\sqrt{\left(\hat{\boldsymbol{p}}_r\right)_{(1)}^2 + \left(\hat{\boldsymbol{p}}_r\right)_{(2)}^2}}{\rho_r} \right), \tag{9.20c}$$

where $\operatorname{atan2}(y, x)$ denotes the four quadrant arctangent function (see Appendix A.6).

If all the array antennas are in the same plane (e.g., URA case), the range is $\rho_r = \left(\hat{\boldsymbol{p}}_r\right)_{(3)}$ and the elevation is computed as

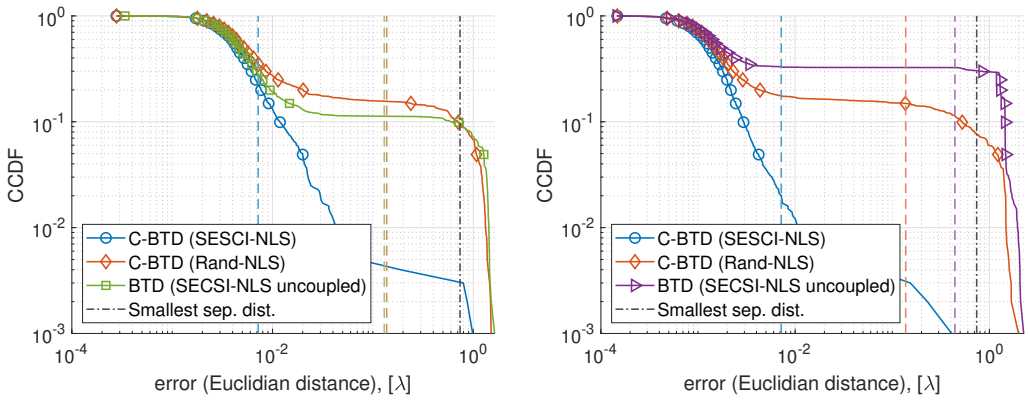
$$\theta_r = \cos^{-1}\left(\frac{\sqrt{\left(\hat{\boldsymbol{p}}_r\right)_{(1)}^2 + \left(\hat{\boldsymbol{p}}_r\right)_{(2)}^2}}{\rho_r} \right). \tag{9.21}$$

The equation (9.21) additionally provides a simple reliability test that can be applied in this case. If the value of $\sqrt{\left(\hat{\boldsymbol{p}}_h\right)_{(1)}^2 + \left(\hat{\boldsymbol{p}}_h\right)_{(2)}^2} / \rho_h$ is larger than one, then the test has failed, and the location estimates should be considered as not reliable.

9.4 Simulation results

This section presents the empirical validation of the COBRAS algorithm, demonstrating its performance and the impact of the different rank-$(L_r, L_r, 1)$ decomposition algorithms on the localization accuracy.

For the simulations, we consider a multi-static MIMO radar system with $M_{\mathrm{T}} = 3$ transmitting arrays and $M_{\mathrm{R}} = 2$ receiving arrays. The carrier frequency is $f_c = 50$ GHz which results in a wavelength of $\lambda = c/f_c \approx 0.6$ cm. As the transmitting and receiving arrays, we use URAs of 10×10 elements, resulting in $N_{\mathrm{T}} = N_{\mathrm{R}} = 100$. The antenna element spacing is set to $\lambda/2$. The Cartesian coordinates of the reference transmit antennas are $\{0, 0, 0\}$, $\{0, 12\lambda, 0\}$, and $\{0, 24\lambda, 0\}$, and the coordinates of the receiving reference antennas are $\{16\lambda, 0, 0\}$, $\{16\lambda, 17\lambda, 0\}$. The antenna arrays are rotated about the y-axis by the angles $\gamma_{\mathrm{T}} = 140^\circ$ and $\gamma_{\mathrm{R}} = 30^\circ$, for transmit and receive sides, respectively. The number of time snapshots is set to $T = 200$, and the number of pulses is $P = 100$. The size of the received data tensors $\boldsymbol{\mathcal{Y}}^{[1]}$ and $\boldsymbol{\mathcal{Y}}^{[2]}$ for the given parameters is $100 \times 200 \times 100$. The range, azimuth, and elevation parameters of the $R = 3$ closely spaced targets with respect to the reference antennas of two receive arrays are $\boldsymbol{\Theta}_{R,1}^{[1]} = \{22\lambda, 116.6^\circ, 44.7^\circ\}$, $\boldsymbol{\Theta}_{R,2}^{[1]} = \{21\lambda, 112.9^\circ, 45.7^\circ\}$, $\boldsymbol{\Theta}_{R,3}^{[1]} = \{23.4\lambda, 110.6^\circ, 43.1^\circ\}$, $\boldsymbol{\Theta}_{R,1}^{[2]} = \{17.9\lambda, -156.8^\circ, 63.8^\circ\}$, $\boldsymbol{\Theta}_{R,2}^{[2]} = \{16.4\lambda, -148.4^\circ, 66^\circ\}$, and $\boldsymbol{\Theta}_{R,3}^{[2]} = \{17.1\lambda, -170.5^\circ, 69.1^\circ\}$, respectively. The example scenario used for the simulations is illustrated in Figure 9.1. The reflection coefficients and the transmitted signal after matched-filtering at the receiver are drawn



(a) First receiving array. **(b)** Second receiving array.

Figure 9.2. CCDF vs. errors in terms of the Euclidian distance between true and estimated locations. Results are averaged over all $R = 3$ targets. SNR = 30 dB. 1000 Monte-Carlo trials.

from a zero mean complex Gaussian distribution with unit variance. The additive noise tensor $\boldsymbol{\mathcal{N}}^{[m_{\mathrm{R}}]}$ is modeled as an independently and identically distributed (i.i.d.) zero mean spatially and temporally white additive noise with variance σ_n^2. The simulations were performed for $K = 1000$ trials.

We evaluate the accuracy of the target localization, defining the errors in terms of the Euclidean distance between the true ($\boldsymbol{u}$) and the estimated ($\hat{\boldsymbol{u}}_k$) location of the target as

$$\mathrm{err} = \sqrt{\eta(\hat{\boldsymbol{u}}_{r,k}) \left\| \boldsymbol{u}_r - \hat{\boldsymbol{u}}_{k,r} \right\|^2} \quad \forall r, \forall k, \tag{9.22}$$

where $k \in 1, \ldots, K$ is a trial index and the

$$\eta(\hat{\boldsymbol{u}}) = \begin{cases} 1, & \text{if } \hat{\boldsymbol{u}} \text{ passed the reliability test} \\ 0, & \text{otherwise} \end{cases} \tag{9.23}$$

represents the indicator function of the reliability test described at the end of Section 9.3.3. Therefore, only the trials in which the reliability test did not fail are taken into account.

Moreover, we measure the estimation accuracy of the location parameter $x \in \{\rho, \phi, \theta\}$ as

$$\mathrm{err}(x) = \sqrt{\eta(\hat{\boldsymbol{u}}_{r,k})(x_r - \hat{x}_{r,k})^2} \tag{9.24}$$

where x_r is the true value of the estimated quantity for the rth target and the $\eta(\hat{\boldsymbol{u}}_{r,k})$ is defined in (9.23), i.e. we again only consider the trials in which the reliability test has been passed.

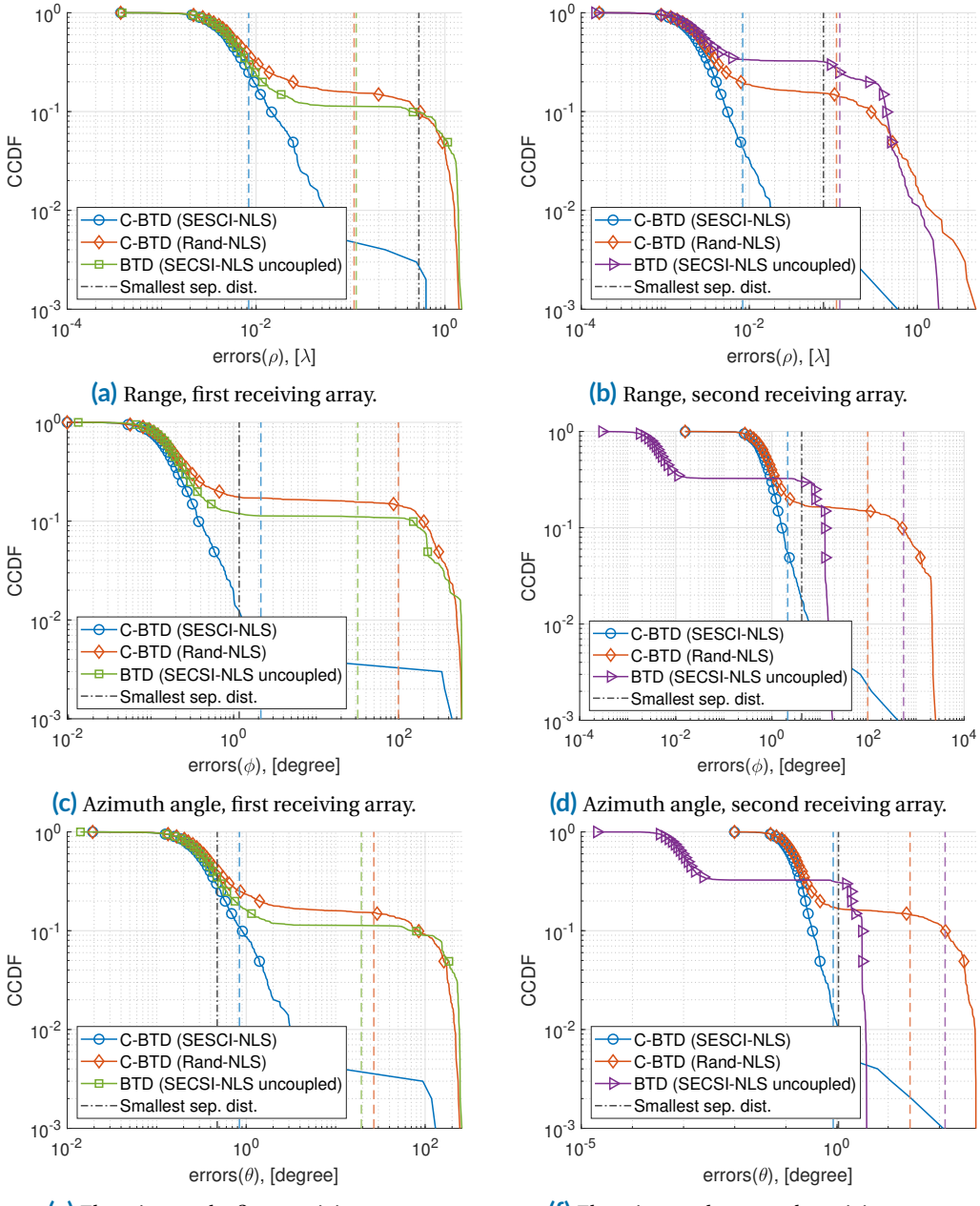
Additionally, as in [PSK+21], we use the minimum separation measure between the values of the parameter x of all targets as

$$s(x) = \min_{r_1, r_2} \frac{\left| x_{r_1} - x_{r_2} \right|}{2} \tag{9.25}$$

where $r_1, r_2 \in \{1, \ldots, R\}$, to ease the representation of the results.

The complementary cumulative distribution function (CCDF) of the errors in terms of the Euclidian distance between the true and the estimated locations of the targets for the SNR = 30 dB is shown in Figure 9.2. The CCDFs in terms of range, azimuth, and elevation parameters of targets are depicted in Figure 9.3. The colored vertical lines in the plots represent the mean values for each curve. We compare three BTD algorithms for the estimation of the steering matrices (the other steps are performed according to COBRAS): (i) coupled rank-$(L_r, L_r, 1)$ BTD algorithm from Chapter 8 and [SKKH22] (nonlinear least squares (NLS) with SECSI-based initialization, blue curve); (ii) NLS solution for coupled rank-$(L_r, L_r, 1)$ BTD from [VDS+16] (random initialization, orange curve); and (iii) non-coupled solution from Chapter 7 and [KSP+23] (green and purple curves). In the third case, only the data from one given array is processed. Therefore, there is no joint processing, and the coupling is not considered. The dash-dotted line denotes the separation distance described in (9.25). In our simulation scenario, only 0.03% of estimates among all trials, targets, and arrays are not reliable for the first and second algorithms. The third algorithm provided zero unreliable estimates. As can be observed, the SECSI-NLS scheme outperforms the considered alternatives, especially the non-coupled approaches. This can be explained by the more reliable initialization provided by the SECSI algorithm, which reduces the number of outliers. Conversely, the NLS solution with random initialization does not always converge. The inferior performance of the single BTD demonstrates the benefits of the coupling: the localization is more accurate and reliable when data from both arrays is processed jointly.

The estimation performances can be further improved by, for example, constructing the system in (9.15) using the estimates from both arrays and using a weighted least squares solution. Another solution is to consider only the estimates from the array that passed the reliability test in (9.23) or take into account only the estimates from the array that is located closer to the target.



(a) Range, first receiving array.

(b) Range, second receiving array.

(c) Azimuth angle, first receiving array.

(d) Azimuth angle, second receiving array.

(e) Elevation angle, first receiving array.

(f) Elevation angle, second receiving array.

Figure 9.3. CCDF vs. errors in terms of the range, azimuth, and elevation parameters of the targets. Results are averaged over all $R = 3$ targets. SNR = 30 dB. 1000 Monte-Carlo trials.

9.5 Conclusions

This chapter presents a high-resolution coupled rank-$(L_r, L_r, 1)$ decomposition-based near-field localization algorithm for multi-static MIMO radar systems. The COBRAS algorithm performs the target location parameters estimation in 3D space based on the exact wavefront model and is applicable to arbitrary array geometries. Compared to the far-field planar wavefront, the use of the exact wavefront model enables the estimation of not only the direction-of-arrival parameters but also the range and, consequently, the position of the target. Moreover, the use of massive antenna arrays increases the Fresnel region and makes the near-field assumption applicable in practical scenarios. Furthermore, the algorithm utilizes a reliability measure, which allows discarding unreliable parameter estimates. The simulation results show that the employment of coupling and joint processing of the data from multiple arrays improves the localization performance compared to the non-coupled solutions. Moreover, we have shown that the coupled BTD algorithm proposed in Chapter 8 outperforms its alternatives from the literature.

Rank-$(L_r, L_r, 1)$ Block-Term Decomposition for Joint Processing of Magnetic Fields and Electrical Potentials

10

This chapter of the thesis is devoted to one of the common and promising applications of tensor decompositions, electroencephalography (EEG) and magnetoencephalography (MEG) data processing. In particular, we show how the coupled rank-$(L_r, L_r, 1)$ decomposition can be applied to the EEG-MEG recordings of somatosensory evoked electrical potentials (SEPs) and somatosensory evoked magnetic fields (SEFs) to separate the signal components related to 200 Hz band activity. The studies show that explorations into oscillatory phenomena in this frequency range have yielded significant insights into identifying and interpreting diverse brain activities. In our analysis, we use the SEPs and SEFs recordings, evoked through electrical median nerve stimulation conducted on eight healthy volunteers. By employing time-frequency analysis, we produce the time-dependent spectra of the signals and organize them into three-dimensional EEG, MEG magnetometer (MAG), MEG gradiometer (GRAD)-1, and MEG gradiometer (GRAD)-2 data tensors. The approach that addresses the challenge of separating signal components associated with the 200 Hz band activity in SEPs and SEFs utilizing the coupled multilinear rank-$(L_r, L_r, 1)$ block-term decomposition has already been proposed in [CRHH21]. The authors assessed and confirmed the advantages of the joint processing of simultaneous EEG and MEG recordings in seven of eight subjects. However, the analysis in [CRHH21] was limited to the joint analysis of only EEG and MEG-MAG tensors. This study focuses on the complete data set, including the gradiometer measurements. In essence, this chapter validates the application of the coupled block-term decomposition (BTD) algorithm proposed in Chapter 8 on the analysis of neural oscillations, which, in turn, holds the potential to provide enhanced insights into brain function and dynamics.

10.1 Introduction

This study focuses on the analysis of somatosensory evoked electrical potentials (SEPs) and somatosensory evoked magnetic fields (SEFs) from EEG (Electroencephalogram) and MEG (Magnetoencephalogram) recordings. These recordings capture signals

from the human somatosensory cortex, originating from peripheral nerve stimulations [HSM+01]. Previous research has identified activity within various frequency bands, notably spectral components around 600 Hz [CMB+94; CMB+97] and around 200 Hz [HSM+01; PCP+13]. These components at approximately 600 Hz are often referred to as high-frequency oscillations (HFOs). The term HFOs also encompasses normal or pathological EEG and MEG signals within the 80 Hz to 250 Hz range [MRJ+17]. Spontaneous non-pathological HFOs occuring during sleep, linked to sleep-specific transient signals [BDLP+16; MFG+18]. Pathological HFOs are emerging as potential epilepsy biomarkers, more indicative of epileptogenic tissue than spikes [Gvv+17]. They appear to possess a more focal nature than spikes and might be involved in initiating epileptic activity, particularly in childhood absence epilepsy [Gvv+17]. MEGs can detect HFOs during the periictal state [JNN+17].

Similar to [CRHH21], in this chapter of the thesis, we exploit tensor decompositions to separate signal components related to 200 Hz band oscillations in SEPs and SEFs. As in many other signal processing fields, the application of tensors for MEG and EEG data analysis is receiving an increased interest [CLK+15; BAC+14; NKHH17; KB09]. EEG and MEG data are one of the examples, for which tensors are perfect representations: EEGs naturally include information in different modes of trials, time, frequency, channel, etc. Thus, there are several works in the literature that utilize tensor decompositions for EEG data processing to perform feature extraction, classification, and source separation [KSAD15; ZGK+21; CMD+15]. For instance, the authors in [GPBS22] propose a method to detect a seizure before its actual manifestation based on the canonical polyadic decomposition (CPD).

As the quality of the acquired EEG signals is often very poor, corrupted by the noise, spikes, and artifacts (due to the sensor limitations, the background environmental noise, or the human factors such as the winks), their suppression plays an important role in the medical data analysis. Thus, for many years, the primary focus of biomedical signal processing has been signal filtering and reduction of the noise. Organisms are complex systems whose subsystems interact, so the measured signals of one subsystem usually contain the signals of other subsystems. Removing unwanted signal components can then support subsequent medical discoveries. A fundamental method for noise cancellation analyzes the signal spectra and suppresses undesired frequency components.

Other efficient methods to suppress the noise and detect the artifacts are based on the independent component analysis and tensor decompositions [NLK+20; DCVP+05]. The factorization of the multiway data into the individual components or factors allows not only isolating the essential signal subspace but also visualizing and removing the

spikes and artifacts. The canonical polyadic (CP) decomposition is probably the one that is most commonly utilized for these tasks. Moreover, the coupled extension of the CPD is considered a good alternative to the single tensor CPD in cases when several data sets have common modes and can be decomposed jointly. A coupled variant of the PARAFAC2 supports simultaneous SEPs and SEFs processing [CNH+18]. While these multilinear processing-based studies offer insights into SEPs' and SEFs' multiplicity within the initial cortical response latency, they do not offer improved descriptions of the 200 Hz band activity as compared to [HSM+01].

To address this, the authors in [CRHH21] suggest utilizing the multilinear rank-$(L_r, L_r, 1)$ block-term decomposition [DeL11; SD15] to extract 200 Hz band activity-related signal components in SEPs and SEFs. Prior applications of the multilinear rank-$(L_r, L_r, 1)$ decomposition in blind source separation and analysis have focused on atrial fibrillation electrocardiogram (ECG) recordings [RAZ16; OZ19]. Limited results in EEG data analysis relied on modeling EEG signals as a sum of exponentially damped sinusoids [HCS+14; DCVP+05].

Main contributions. As in [CRHH21], in this study, we employ the smoothed pseudo Wigner-Ville distribution-based (SPWV) method [Coh95] for the time-frequency analysis of SEPs and SEFs. We arrange time-dependent spectra obtained after the time-frequency analysis into three-dimensional tensors encompassing frequency, space (channels), and time modes. In contrast to [CRHH21], we consider the joint processing of not two but four data tensors. More precisely, we apply the coupled rank-$(L_r, L_r, 1)$ block-term decomposition to EEG, MEG magnetometer (MAG), and MEG gradiometer (GRAD-1 and GRAD-2) tensors. Then we use multilinear-rank factors of the decomposition to visualize channel-dependent spectral signatures, enabling extraction of the 200 Hz signal component.

The remainder of this chapter is organized as follows. In Section 10.2, we describe the analyzed data, the preprocessing steps, and the tensor construction. Section 10.3 is devoted to the coupled rank-$(L_r, L_r, 1)$ decomposition-based analysis of the EEG-MEG data. Section 10.4 presents the results, and Section 6.5 concludes the chapter.

10.2 Data description

The analysis in this study involved SEPs (Somatosensory Evoked Potentials) and SEFs (Somatosensory Evoked Fields). These were recorded with a 60-channel EEG cap and

a 306-channel helmet-shaped MEG system [GHWH14] at the Biomagnetic Center of the University Hospital in Jena, Germany.

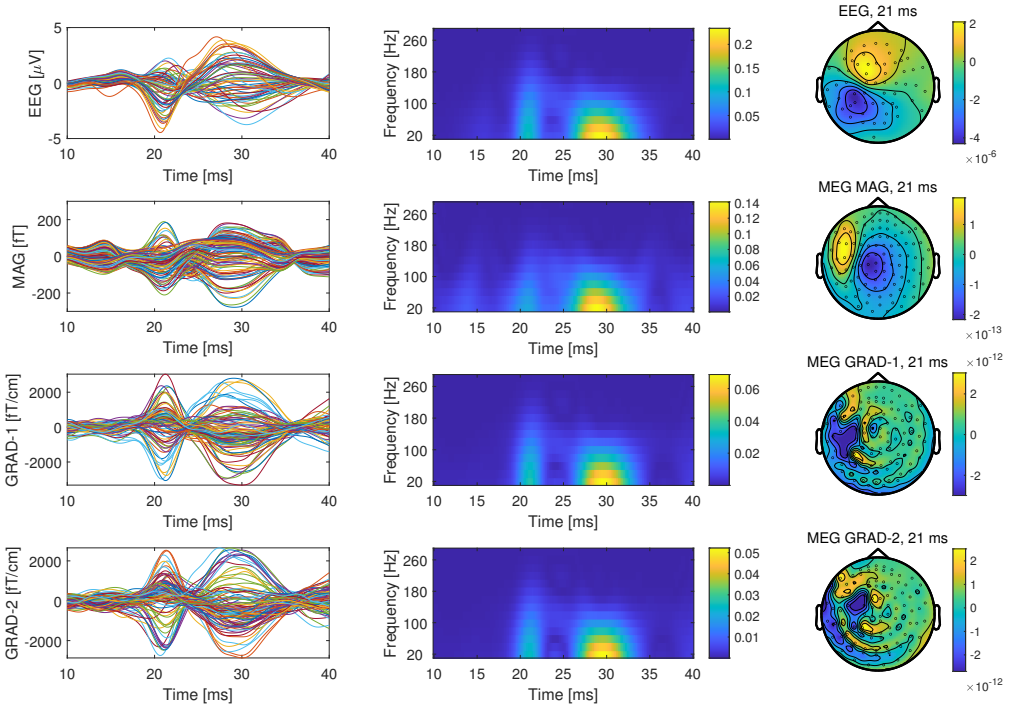
To stimulate the right median nerve, monophasic square wave constant current pulses with a 200 µs duration were employed. These pulses were generated by a clinical constant current stimulator (DS7A, Digitimer Ltd., Welwyn Garden City, United Kingdom). Electrodes were placed on the right hand's wrist for each volunteer. The current amplitude was tailored individually based on the International Federation of Clinical Neurophysiology's motor plus sensory threshold guidelines [NAD+94]. Stimulation occurred at a repetition rate of 2 Hz (consisting of 6000 trials) with intermittent pauses introduced approximately every 15 minutes to ensure sustained participant attention. These pauses, lasting 2 seconds, were to be counted by the participants.

For the determination of electrode positions, anatomical locations, and MEG localization coil sets, a 3D Digitizer (3SPACE FASTRAK; Polhemus Inc., Colchester, VT) was used. For further detailed information about the experimental setup, we refer the reader to [GHWH14]. The study primarily concentrates on data collected from eight participants. The data were sampled at 5 kHz. The initial raw MEG data underwent filtering using Maxfilter Version 2.0.21 (Elekta Neuromag Oy, Helsinki, Finland) with a time-domain extension [TH09]. A constant interpolation technique was applied immediately post-stimulation to minimize artifacts.

We used the FieldTrip toolbox [OFMS11], to segment the data into trials in accordance with the trigger information. The time window of interest spanned from 80 ms before stimulation to 100 ms after the stimulation. The signals underwent a band-pass filter (fourth-order Butterworth) ranging from 10 to 300 Hz. Taking subject 1 as an example, two bad EEG and five bad GRAD-1 and GRAD-2 channels were identified, removed, and the analysis proceeded with the remaining 58 EEG, 97 MEG-GRAD-1, 97 MEG-GRAD-2, and 102 MEG-MAG channels. The time behaviors averaged across all trials, are presented in Figure 10.1(a). The spatial distribution of EEG, MEG-MAG, MEG-GRAD-1, and MEG-GRAD-2 at 21 ms after stimulation is depicted in Figure 10.1(c).

Recognizing the time-varying characteristics of the SEPs and SEFs, the smoothed pseudo Wigner-Ville distribution (SPWV) [Coh95] was employed to obtain the time-dependent spectra from these signals, with a frequency resolution of 20 Hz. The outcomes are shown in Figure 10.1(b). An increase in power around 20 ms post-stimulation is noticeable across various frequency bands.

For further analysis, the data were structured into three-dimensional tensors. Each channel's time-frequency representation formed one lateral slice of the corresponding



(a) Butterfly plots of all channels.

(b) Time-frequency representation of the signals averaged over all channels.

(c) Spatial distributions at 21 ms after the stimulation.

Figure 10.1. Different representations of EEG, MEG-MAG, MEG-GRAD-1, and MEG-GRAD-2 signals.

EEG, MEG-MAG, MEG-GRAD-1, and MEG-GRAD-2 tensor. These tensors had dimensions of $N_F \times N_C \times N_T$, where N_F refers to the number of frequency bins, N_C denotes the number of channels, and N_T denotes the number of time samples. The tensors were constructed for the time window from 10 ms to 40 ms after stimulation, resulting in 151 samples. For subject 1, the dimensions of the EEG, MEG-MAG, MEG-GRAD-1, and MEG-GRAD-2 data tensors, which were later analysed by the to coupled multilinear rank-$(L_r, L_r, 1)$ block-term decomposition, were $14 \times 58 \times 151$, $14 \times 102 \times 151$, $14 \times 97 \times 151$, and $14 \times 97 \times 151$, respectively.

10.3 Coupled decomposition of EEG and MEG tensors

Based on the results obtained [CRHH21], in this study, we employ the coupled multilinear rank-$(L_r, L_r, 1)$ decomposition to jointly process the simultaneously recorded SEPs and SEFs. Leveraging the observation that temporal patterns in SEPs and SEFs exhibit comparability, we couple the three-mode (temporal mode) of EEG, MEG-MAG, MEG-GRAD-1, and MEG-GRAD-2 data tensors denoted as $\boldsymbol{\mathcal{X}}_{\text{EEG}} \in \mathbb{R}^{N_F \times N_C^E \times N_T}$, $\boldsymbol{\mathcal{X}}_{\text{MAG}} \in \mathbb{R}^{N_F \times N_C^M \times N_T}$, $\boldsymbol{\mathcal{X}}_{\text{GRAD1}} \in \mathbb{R}^{N_F \times N_C^{G1} \times N_T}$, and $\boldsymbol{\mathcal{X}}_{\text{GRAD2}} \in \mathbb{R}^{N_F \times N_C^{G1} \times N_T}$, respectively. Here, N_C^E, N_C^M, N_C^{G1}, and N_C^{G2} denote the corresponding number of channels. The coupled rank-$(L_r, L_r, 1)$ decomposition of the four tensors is expressed as follows

$$\boldsymbol{\mathcal{X}}_{\text{EEG}} = \sum_{r=1}^{R} \left(\boldsymbol{A}_{\text{EEG},r} \cdot \boldsymbol{B}_{\text{EEG},r}^{\mathrm{T}} \right) \circ \boldsymbol{c}_r + \boldsymbol{\mathcal{N}}_{\text{EEG}}, \tag{10.1a}$$

$$\boldsymbol{\mathcal{X}}_{\text{MAG}} = \sum_{r=1}^{R} \left(\boldsymbol{A}_{\text{MAG},r} \cdot \boldsymbol{B}_{\text{MAG},r}^{\mathrm{T}} \right) \circ \boldsymbol{c}_r + \boldsymbol{\mathcal{N}}_{\text{MAG}}, \tag{10.1b}$$

$$\boldsymbol{\mathcal{X}}_{\text{GRAD1}} = \sum_{r=1}^{R} \left(\boldsymbol{A}_{\text{GRAD1},r} \cdot \boldsymbol{B}_{\text{GRAD1},r}^{\mathrm{T}} \right) \circ \boldsymbol{c}_r + \boldsymbol{\mathcal{N}}_{\text{GRAD1}}, \tag{10.1c}$$

$$\boldsymbol{\mathcal{X}}_{\text{GRAD2}} = \sum_{r=1}^{R} \left(\boldsymbol{A}_{\text{GRAD2},r} \cdot \boldsymbol{B}_{\text{GRAD2},r}^{\mathrm{T}} \right) \circ \boldsymbol{c}_r + \boldsymbol{\mathcal{N}}_{\text{GRAD2}}. \tag{10.1d}$$

The tensors share the same three-mode factor matrix $\boldsymbol{C} = [\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_R] \in \mathbb{R}^{N_T \times R}$. where R denotes the number of block-terms, or, equally, the number of temporal signatures. The multilinear ranks of $\boldsymbol{\mathcal{X}}_{\text{EEG}}$, $\boldsymbol{\mathcal{X}}_{\text{MAG}}$, $\boldsymbol{\mathcal{X}}_{\text{GRAD1}}$, and $\boldsymbol{\mathcal{X}}_{\text{GRAD2}}$ are denoted as L_r^E, L_r^M, L_r^{G1}, and L_r^{G2}, respectively. Since we perform the coupling in the 3-mode, the multilinear ranks of the coupled tensors do not have to be equal. The corresponding factor matrices for $\boldsymbol{\mathcal{X}}_{\text{EEG}}$ are $\boldsymbol{A}_{\text{EEG},r} \in \mathbb{R}^{N_F \times L_r^E}$ and $\boldsymbol{B}_{\text{EEG},r} \in \mathbb{R}^{N_C^E \times L_r^E}$. For the MEG tensors $\boldsymbol{\mathcal{X}}_{\text{MAG}}$, $\boldsymbol{\mathcal{X}}_{\text{GRAD1}}$, and $\boldsymbol{\mathcal{X}}_{\text{GRAD2}}$, the factor matrices are $\boldsymbol{A}_{\text{MAG},r} \in \mathbb{R}^{N_F \times L_r^M}$ and $\boldsymbol{B}_{\text{MAG},r} \in \mathbb{R}^{N_C^M \times L_r^M}$, $\boldsymbol{A}_{\text{GRAD1},r} \in \mathbb{R}^{N_F \times L_r^{G1}}$ and $\boldsymbol{B}_{\text{GRAD1},r} \in \mathbb{R}^{N_C^{G1} \times L_r^{G1}}$, and $\boldsymbol{A}_{\text{GRAD2},r} \in \mathbb{R}^{N_F \times L_r^{G2}}$ and $\boldsymbol{B}_{\text{GRAD2},r} \in \mathbb{R}^{N_C^{G2} \times L_r^{G2}}$, respectively. The four tensors $\boldsymbol{\mathcal{N}}_{\text{EEG}}, \boldsymbol{\mathcal{N}}_{\text{MAG}}, \boldsymbol{\mathcal{N}}_{\text{GRAD1}}$, and $\boldsymbol{\mathcal{N}}_{\text{GRAD2}}$ collect the residuals.

Due to the uniqueness properties of the rank-$(L_r, L_r, 1)$ block-term decomposition (see Section 2.2.6 for details), we use the following representation of the coupled BTD decomposition of the tensors in (10.1a), (10.1b), (10.1c), and (10.1d)

$$\boldsymbol{\mathcal{X}}_{\text{EEG}} = \sum_{r=1}^{R} \boldsymbol{G}_{\text{EEG},r} \circ \boldsymbol{c}_r + \boldsymbol{\mathcal{N}}_{\text{EEG}}, \tag{10.2a}$$

$$\boldsymbol{\mathcal{X}}_{\text{MAG}} = \sum_{r=1}^{R} \boldsymbol{G}_{\text{MAG},r} \circ \boldsymbol{c}_r + \boldsymbol{\mathcal{N}}_{\text{MAG}}, \tag{10.2b}$$

$$\boldsymbol{\mathcal{X}}_{\text{GRAD1}} = \sum_{r=1}^{R} \boldsymbol{G}_{\text{GRAG1},r} \circ \boldsymbol{c}_r + \boldsymbol{\mathcal{N}}_{\text{GRAD1}}, \tag{10.2c}$$

$$\boldsymbol{\mathcal{X}}_{\text{GRAD2}} = \sum_{r=1}^{R} \boldsymbol{G}_{\text{GRAG2},r} \circ \boldsymbol{c}_r + \boldsymbol{\mathcal{N}}_{\text{GRAD2}}, \tag{10.2d}$$

where

$$\boldsymbol{G}_{\text{EEG},r} = \boldsymbol{A}_{\text{EEG},r} \cdot \boldsymbol{B}_{\text{EEG},r}^{\text{T}} \in \mathbb{R}^{N_F \times N_C^E}, \tag{10.3a}$$

$$\boldsymbol{G}_{\text{MAG},r} = \boldsymbol{A}_{\text{MAG},r} \cdot \boldsymbol{B}_{\text{MAG},r}^{\text{T}} \in \mathbb{R}^{N_F \times N_C^M}, \tag{10.3b}$$

$$\boldsymbol{G}_{\text{GARD1},r} = \boldsymbol{A}_{\text{GARD1},r} \cdot \boldsymbol{B}_{\text{GARD1},r}^{\text{T}} \in \mathbb{R}^{N_F \times N_C^{G1}}, \tag{10.3c}$$

$$\boldsymbol{G}_{\text{GARD2},r} = \boldsymbol{A}_{\text{GARD2},r} \cdot \boldsymbol{B}_{\text{GARD2},r}^{\text{T}} \in \mathbb{R}^{N_F \times N_C^{G2}} \tag{10.3d}$$

represent the channel-dependent frequency signatures for the EEG, MEG-MAG, MEG-GRAD-1, and MEG-GRAD-2 tensors, respectively. In this setup, the ith columns, $i \in \{1,\ldots,N_C\}$, of $\boldsymbol{G}_{\text{EEG},r}$, $\boldsymbol{G}_{\text{MAG},r}$, $\boldsymbol{G}_{\text{GRAD1},r}$, and $\boldsymbol{G}_{\text{GRAD2},r}$ represent the spectral signatures of the ith channel of the rth component, $r \in \{1,\ldots,R\}$. The coupled BTD according to the equations (10.2a), (10.2b), (10.2c), and (10.2d) is illustrated in Figure 10.2 (assuming zero residual for simplicity of the representation). The jth rows, $j \in \{1,\ldots,N_F\}$, of $\boldsymbol{G}_{\text{EEG},r}$, $\boldsymbol{G}_{\text{MAG},r}$, $\boldsymbol{G}_{\text{GRAD1},r}$, and $\boldsymbol{G}_{\text{GRAD2},r}$ are identified as the spatial signature of the rth component, $r \in \{1,\ldots,R\}$ with respect to the jth frequency bin. The temporal signature of the rth component is represented by the rth column, $r \in \{1,\ldots,R\}$, of the factor matrix $\boldsymbol{C} \in \mathbb{R}^{N_T \times R}$ of the time mode.

As has been mentioned before, the block-term decomposition is unique up to scaling and permutation ambiguities [DeL08a]. Given the insignificance of resulting signature scaling, normalization has been applied to ensure comparability of the rth signal components signatures in plots. As one of the advantages of coupled BTD-based multi-way component analysis can be considered its capability to extract channel-dependent frequency signatures, an interesting feature compared to the other works from the literature. We want to highlight that the PARAFAC2 decomposition [Har72] of data tensors can be formulated in a manner similar to (10.2a)-(10.2d) [CHGH18]. However, the imposition of the Harshman constraint [Har72] is essential to ensure PARAFAC2's uniqueness, resulting in a full rank assumption for the $\boldsymbol{G}_r$s in PARAFAC2. These characteristics constrain the full exploitation of inherent variability within SEPs and SEFs. By contrast, the multilinear rank-$(L_r, L_r, 1)$ block-term decomposition offers greater flexibility. Fine-tuning of the parameters such as the number of components R and the multilinear ranks L_r contributes to the alignment between the data and

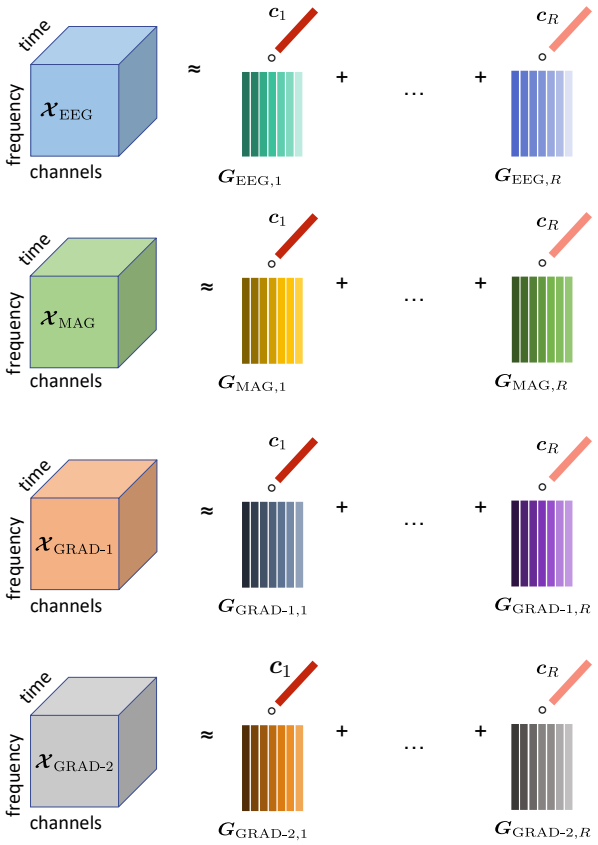


Figure 10.2. Coupled rank-$(L_r, L_r, 1)$ decomposition of four tensors (EEG, MEG-MAG, MEG-GRAD-1, and MEG-GRAD-2) with 3-mode in common. Multilinear ranks are equal to 7.

the decomposition model [CRHH21], leading to the extraction of significant signal characteristics.

10.4 Results

In this section, we present the signal signatures obtained via the coupled rank-$(L_r, L_r, 1)$ BTD of the data tensors defined in the previous section. For the decompositions, we set the block-rank R to 2 and the multilinear ranks L_r to 7, as the authors in [CRHH21] showed that these are suitable settings for the given data.

The common temporal signatures and separate spectral signatures of EEG, MEG-MAG, MEG-GRAD-1, and MEG-GRAD-2 tensors of subject 1 obtained via the coupled rank-$(L_r, L_r, 1)$ decomposition are shown in Figures 10.3(a) to 10.3(d), respectively

(left column - temporal signatures, middle and right columns - spectral signatures corresponding to components one and two). The different curves on the plots of the spectral signatures correspond to different channels. Figures 10.4(a) to 10.4(d) show the corresponding spatial signatures at selected frequencies. The spectral and temporal signatures reveal a clear difference between the spectral signature of component 1 and component 2. The ascending slope from approximately 100 to 200 Hz is exclusive to the spectral component 1, which, according to [CRHH21], indicates a separation of the 200 Hz component. Consequently, the first component is the signature related to the 200 Hz band activity.

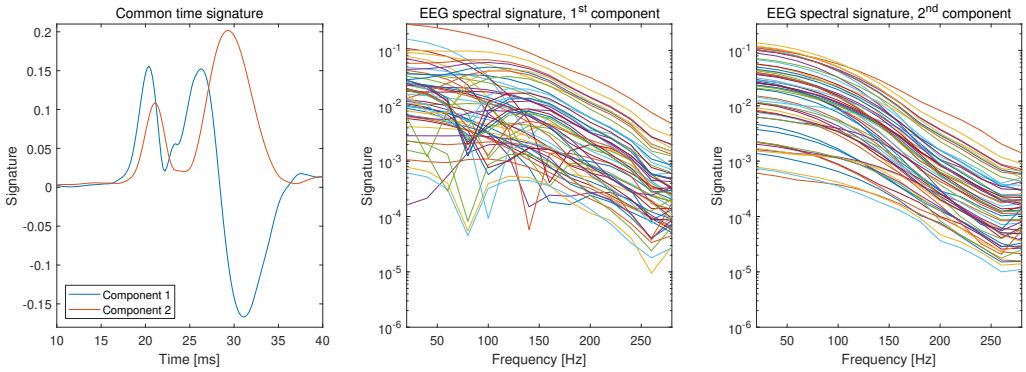
The examples of extracted spatial signatures are presented in Figures 10.4(a) to 10.4(d). We can notice that the higher frequency components (at 200 and 260 Hz) are more focal than the ones at lower frequencies. The spatial signature at 140 Hz marks a transition, while the spatial signatures at 200 Hz and 260 Hz show similarities.

In contrast to the approach in [CRHH21], in this study, we utilize the coupled SECSI-based algorithm for computing the joint rank-$(L_r, L_r, 1)$ decomposition proposed in Chapter 8 and in [SKKH22]. It provides decent results in a single run without the need for multiple random initializations recommended for the Tensorlab algorithm [SDD15]. Therefore, Figures 10.3(a) to 10.3(d) show that the coupled SECSI-BTD algorithm proposed in Chapter 8 has an advantage over the conventional coupled rank-$(L_r, L_r, 1)$ block-term decomposition algorithm employed in [CRHH21].

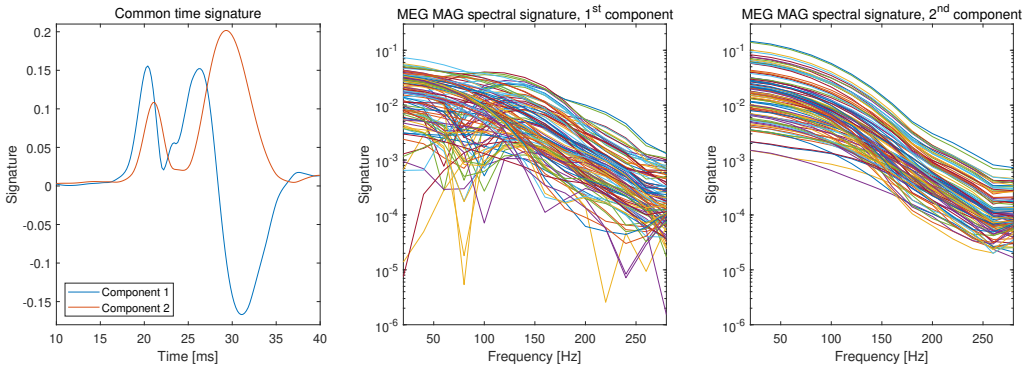
In this study, we primarily focused on the extraction of high-frequency components from the noise-corrupted EGG-MEG measurements. Some more findings on the evaluation of the results can be found in [CRHH21]. Generally, low-rank tensor decompositions, such as CP or block-term decompositions, allow the extraction and separation of distinct components from the data. These distinct components may as well correspond to artifacts. Therefore, tensor techniques have also been successfully employed for the removal of artifacts [MGK+21; GHH14; GAT+16].

10.5 Conclusions

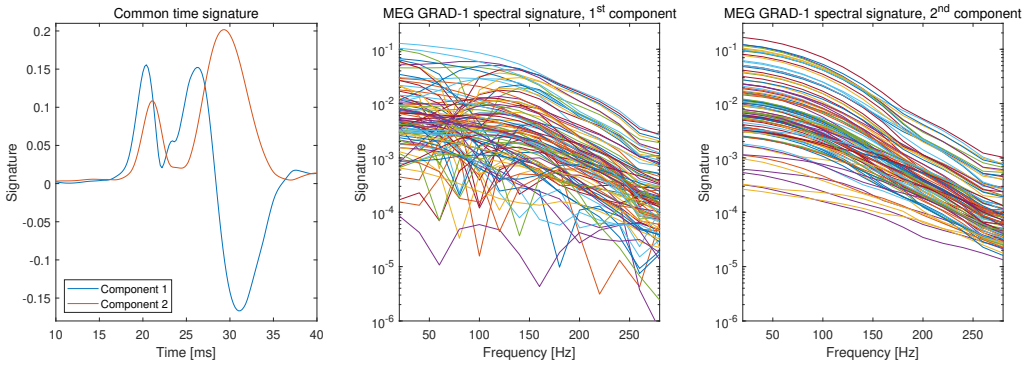
In this chapter, we have utilized the coupled rank-$(L_r, L_r, 1)$ block-term decomposition for the joint processing of SEP and SEF recordings in EEG, MEG-MAG, MEG-GRAD-1, and MEG-GRAD-2 data tensors. The results have shown that the coupled SECSI-BTD algorithm we have proposed in Chapter 8 effectively extracts the channel-dependent



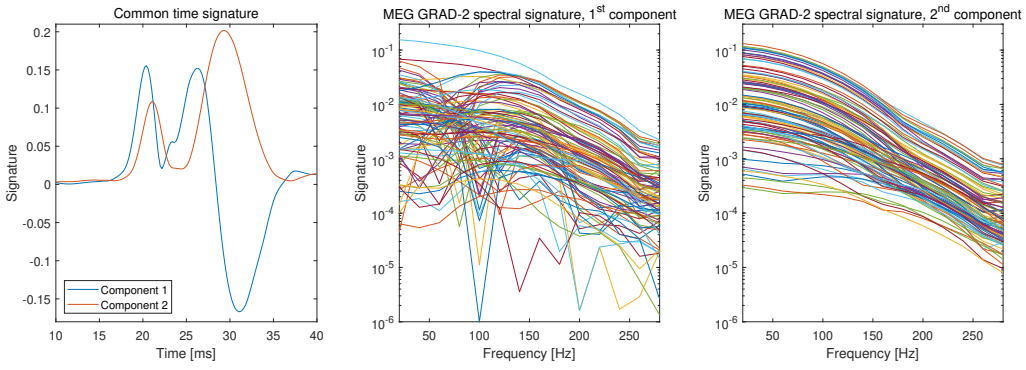
(a) EEG.



(b) MEG-MAG.

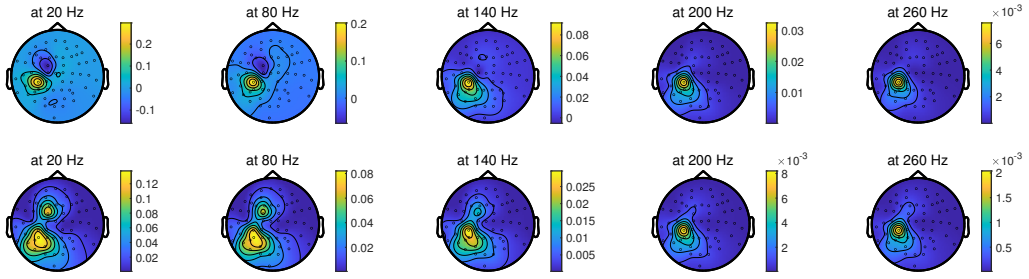


(c) MEG-GRAD-1.

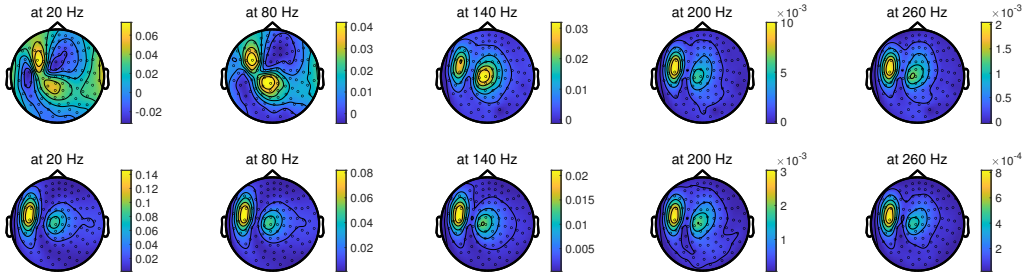


(d) MEG-GRAD-2.

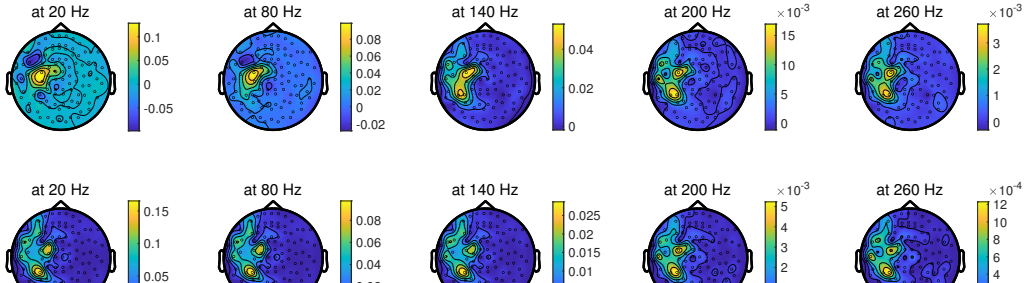
Figure 10.3. Common temporal (first column) and respective separate spectral signatures (second and third columns) of jointly decomposed EEG, MEG-MAG, MEG-GRAD-1, and MEG-GRAD-2 tensors.



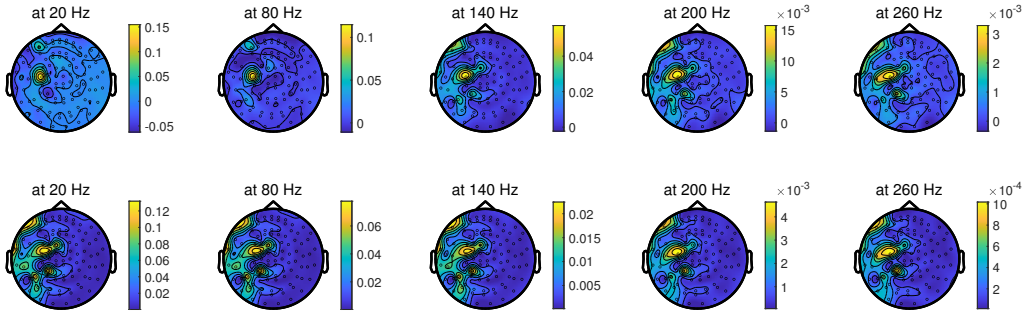
(a) EEG.



(b) MEG-MAG.



(c) MEG-GRAD-1.



(d) MEG-GRAD-2.

Figure 10.4. Spatial signatures extracted via the coupled rank-$(L_r, L_r, 1)$ decomposition of EEG, MEG-MAG, MEG-GRAD-1, and MEG-GRAD-2 tensors. For every data set, the top and bottom rows represent the first and the second components, respectively.

spectral signatures and, therefore, achieves the separation of the 200 Hz band activity-related signal components in SEPs and SEFs. Compared to the previous studies on these data, we performed the data fusion of four measurement tensors, including not only the conventional EEG and MEG-MAG data but also the gradiometer recordings. The decomposition resulted in the effective extraction of temporal, spectral, and spatial signatures that can potentially help neuroscientists to gain better insights into the brain functions. The approach used in this study can also be applied to other combined evoked potential and field recordings and can be utilized for artifact removal.

Part IV.

Conclusions & Future Directions

Conclusions & Future Directions

11

In this final chapter, we summarize the key results from our study on the multilinear generalized singular value decomposition (ML-GSVD) and the multilinear rank-$(L_r, L_r, 1)$ block-term decomposition (BTD) and their applications. Additionally, we outline possible research directions for future work.

11.1 Summary

Tensor decompositions are a powerful mathematical tool for processing, analyzing, and handling multidimensional data. Unlike traditional matrix-based techniques, tensor factorizations possess better uniqueness properties and can capture complex relationships and patterns across multiple dimensions. Therefore, they find applications in diverse fields, including image and signal processing, machine learning, neuroscience, communications, and more. There are many different tensor decompositions in the literature, and each finds use in various signal processing fields. The core content of the thesis is divided into two main parts based on two types of tensor decompositions: multilinear generalized singular value decomposition (Part II) and rank-$(L_r, L_r, 1)$ block-term decompositions (Part III). More specifically, in Part II, we introduce the ML-GSVD, an algorithm to compute it, and present its application in multi-user MIMO communication systems. Moreover, in Part III of the thesis, we explore the rank-$(L_r, L_r, 1)$ and coupled rank-$(L_r, L_r, 1)$ block-term decompositions, propose new algorithms to calculate them, and demonstrate their application to radar systems and joint electroencephalography (EEG) and magnetoencephalography (MEG) data processing. In the following, we give a more specific summary of the thesis chapters.

Chapter 2 is devoted to the basics of tensor algebra. We present the notation used in the thesis and review the fundamental concepts and tools of multilinear algebra. The presented material includes essential definitions and concepts of tensor algebra, such as n-mode products, ranks, unfoldings, and vectorization. Moreover, in this chapter,

we give a brief overview of the basic tensor and matrix decompositions that we then exploit in the subsequent chapters. The discussed factorizations include the generalized singular value decomposition (GSVD) of two matrices, which extension to multiple matrices we present in Chapter 3. Additionally, we briefly look at the PARAFAC2 decomposition, also utilized in Chapter 3. Moreover, we review the higher-order singular value decomposition (HOSVD) and the canonical polyadic (CP) decomposition, the most popular tensor decompositions, and discuss some of the characteristics of the tensor factorizations, such as uniqueness, ambiguities, and matrix representations. Furthermore, we review some of their computational algorithms and applications. At the end of the chapter, we cover the block-term decompositions and their different representations that we later use in Part III of the thesis.

In Chapter 3, we present a new Multilinear Generalized Singular Value Decomposition (ML-GSVD) as an extension of the matrix-based GSVD to jointly factorize a set of K matrices with one common dimension. In comparison with existing GSVD generalizations, our ML-GSVD preserves the properties of the original GSVD, such as the orthogonality of the 2-mode factor matrices. We consider three different cases of the ML-GSVD depending on the dimensions and the rank structure of the decomposed matrices. Moreover, we show that every case provides a certain decomposition structure with private and/or common subspaces. Furthermore, by exploiting the connection to the PARAFAC2 decomposition, we develop an alternating least squares (ALS)-based algorithm to compute the proposed ML-GSVD. Since the ML-GSVD extends the matrix GSVD, it can be employed further in several different communication and biomedical applications, such as coordinated beamforming, MIMO relaying, physical layer security, and genomic signal processing. The ML-GSVD allows increasing the number of jointly factorized matrices and, therefore, can be used in more complex systems.

In Chapter 4, we introduce an application of the ML-GSVD to multi-user MIMO systems with joint unicast and multicast transmissions. We show that in such a system scenario, the factor matrices and subspace structure of the ML-GSVD can be utilized to design precoding and decoding matrices. Moreover, we demonstrate that the three cases of the ML-GSVD can 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.

Chapter 5 of this thesis presents ML-GSVD-based non-orthogonal multiple access (NOMA) transmission technique that can be applied in power-domain MIMO-NOMA downlink communication systems with multiple users. It utilizes the ML-GSVD to

design the precoding and decoding matrices that jointly diagonalize the channels between the BS and the users. Compared to the GSVD-based techniques proposed in the literature, the ML-GSVD supports more than two users on one frequency resource, and the common subspace of the ML-GSVD can be employed to transmit the combined signals of multiple users. Additionally, we present a simple power allocation technique based on the generalized singular values of the ML-GSVD that outperforms the conventional fixed power allocation. We evaluate the performance of the proposed scheme in terms of achievable rates and compare it to traditional orthogonal multiple access (OMA) and state-of-the-art NOMA techniques.

In Chapter 6 of the thesis, we introduce another promising application of the ML-GSVD presented in Chapter 3: rate-splitting multiple access (RSMA) systems. More specifically, this chapter proposes a precoder design for MIMO broadcast systems with rate splitting at the transmitter. We show how the GSVD and the ML-GSVD can be employed to define the number of common and private streams as well as to adjust the message split. The proposed scheme applies to both underloaded and overloaded communication systems and supports the transmission of multiple common and private streams. Moreover, for the cases where the total number of streams does not exceed the number of transmit antennas, we introduce transmit precoding and receive combining designs based on the ML-GSVD, which allow simultaneous transmission of common and private streams but do not require SIC at the receivers. The use of the ML-GSVD overcomes the two-user limitation of the GSVD, allowing its application to systems with more than two users where the common streams can also be transmitted to selected groups of users (multi-layer hierarchical or generalized RS). Simulation results show that both proposed rate splitting schemes outperform the conventional space division multiple access (SDMA) and NOMA.

Chapter 7 is devoted to the rank-$(L_r, L_r, 1)$ block-term decomposition that represents a tensor as a linear combination of low multilinear rank terms and can be explicitly related to the canonical polyadic decomposition. In this chapter, we introduce the SECSI-BTD framework, which exploits the aforementioned connection between the two decompositions to estimate the block-terms of the rank-$(L_r, L_r, 1)$ BTD. The proposed SECSI-BTD algorithm includes the initial calculation of the factor estimates using the SEmi-algebraic framework for approximate Canonical polyadic decompositions via SImultaneous Matrix Diagonalizations (SECSI), followed by clustering and refinement procedures that return the appropriate rank-$(L_r, L_r, 1)$ BTD terms. Moreover, we introduce a new approach to estimate the multilinear rank structure of the tensor based on the HOSVD and k-means clustering. Since the proposed SECSI-BTD algorithm does not require a known rank structure but can still take advantage of

the known ranks when available, it is more flexible than the existing techniques in the literature. Additionally, our algorithm does not require multiple initializations, and the simulation results show that it provides more accurate results and a better convergence behavior for an extensive range of SNRs.

In Chapter 8 of the thesis, we introduce a new algorithm to estimate the factors of the coupled rank-$(L_r, L_r, 1)$ BTD of multiple tensors with a common mode based on the coupled extension of the semi-algebraic framework for approximate CP decompositions via simultaneous matrix diagonalizations (SECSI). The chapter content also includes the technique to estimate the multilinear rank structure of the coupled decomposition. The proposed framework extends the SECSI-BTD algorithm in Chapter 7 to the coupled tensors case. The numerical results demonstrate that compared to the approaches in [VDS+16; SVBD15], the proposed coupled SECSI-based framework provides a much more stable initialization resulting in better convergence and accuracy performances. Moreover, in contrast to [VDS+16; SVBD15], our approach allows performing the decomposition even if the ranks are not available beforehand. Therefore, it can be further used in real data applications with unknown rank structures, for instance, for EEG-MEG recordings, since these signals are usually recorded simultaneously and thus exhibit coupling.

Chapter 9 presents a new coupled rank-$(L_r, L_r, 1)$ block-term decomposition-based near-field localization algorithm for multi-static MIMO radar systems. The proposed COBRAS (COupled Block-term decomposition for multi-static RAdar Systems) algorithm performs the target location parameters estimation in 3D space based on the exact spherical wavefront model and is applicable to arbitrary array geometries. Compared to the far-field planar wavefront, the use of the exact wavefront model enables the estimation of not only the direction-of-arrival parameters but also the range and, consequently, the position of the target. Additionally, the algorithm utilizes a reliability measure, which allows removing the unreliable parameter estimates. The simulation results demonstrate that the employment of coupling and joint processing of the data from multiple arrays improves the localization performance compared to non-coupled solutions. Moreover, our simulations show that the proposed coupled BTD algorithm outperforms its alternatives from the literature.

In Chapter 10, we utilize the coupled rank-$(L_r, L_r, 1)$ block-term decomposition for the joint processing of evoked electrical potentials (SEPs) and somatosensory evoked magnetic fields (SEFs) recordings in EEG, MEG magnetometer (MAG), MEG-GRAD-1, and MEG-GRAD-2 data tensors. Similar to the state-of-the-art publications on this topic, we aim to extract the 200 Hz band activity-related signal components using tensor decompositions. Compared to the previous studies on these data, we jointly

process the complete data set, including not only the conventional EEG and MEG-MAG data but also the gradiometer measurements. The results show that the coupled SECSI-BTD algorithm we have proposed in Chapter 8 effectively extracts the channel-dependent spectral signatures and, therefore, achieves the separation of the 200 Hz band activity-related signal components in SEPs and SEFs. The decomposition results in the effective extraction of temporal, spectral, and spatial signatures that can be used for further processing and potentially help neuroscientists to gain better insights into brain functions.

11.2 Future Work

Tensor-based signal processing, and in particular a newly proposed ML-GSVD and BTD algorithms discussed in this thesis, reveal a broad landscape of possibilities for future research and exploration. We have already outlined earlier that different tensor decompositions find their use in various applications. In this section, we want to identify some open aspects and future directions in the field of tensor decompositions and their application to communication systems, sensing, and biomedical signal processing.

As shown in Part II of the thesis, the ML-GSVD is an extension of GSVD to jointly factorize the set of two or more matrices with one common dimension. In comparison with other extensions of the GSVD, it preserves the original GSVD subspace structure and the orthogonality of the individual factor matrices. However, in comparison to the GSVD, the ML-GSVD is an approximation in the least squares sense. Therefore, there is still a place for improvement of its computational algorithm, for example, in terms of the reconstruction error performance or the computational cost. Moreover, a performance analysis can be a not very straightforward but interesting topic. Furthermore, we have discussed in Chapter 3, that the number of common and private subspaces of the ML-GSVD depends on the ranks and the dimensions of the tensor slices, but can also vary depending on the realization of a tensor. Therefore, we consider deriving the analytical results on the subspace structure of the ML-GSVD as an appealing task.

As we have seen from Chapters 4 to 6, the ML-GSVD, with its subspace structure, is a promising tool for the design of precoders in multi-user MIMO systems. In comparison to the schemes from the literature, the ML-GSVD-based technique is more flexible and intuitive, especially when considering joint multicast and unicast transmissions of the rate-splitting multiple access. However, there are still many open issues in this area of research. Thinking about a related direction, in our RSMA and unicast-multicast

MIMO studies, we have focused primarily on design of the precoders. Therefore, the channel estimation or the precoder optimization under the assumption of only partial channel knowledge can be a very significant future contribution.

Considering more global open issues in the RSMA studies, the important research directions include general studies on the RSMA framework. Many publications often focus only on one aspect of the system and do not consider its interaction with the adjacent fields or consider simplified settings, for example, single antenna users, only two users, and one-layer RSMA. Therefore, investigations in more general settings are important. Moreover, the complexity of a system increases greatly with more layers and users. Then, the problem of optimizing the power allocation between private and common streams becomes very complex and computationally expensive. Thus, this is an essential issue to consider. Furthermore, many aspects of the RSMA system, including its complexity, make the researchers question its efficient practical implementation. Therefore, extensive studies with realistic modeling that will prove them right or wrong are crucial. More than that, there are also other open issues in the area, for which we refer the reader to [MDC+22].

Thinking about more applications of the ML-GSVD, as an interesting perspective, we see a physical layer security application [KWWE07], where the ML-GSVD can be used to extend the existing results to more than two users. Moreover, we discussed in Chapter 3, that the authors in [PSVLA11; ABB03] employ the GSVD and one of its extensions, the higher order generalized singular value decomposition (HO GSVD), for genomic data processing. Therefore, we think it might be interesting to investigate the applicability of the ML-GSVD for similar problems and compare it to the state-of-the-art results.

Moving to the Part III of the thesis, where we considered the multilinear block-term decomposition (BTD), its computation, and applications, also has several future research directions to offer.

As we discussed in Chapter 7, the rank-$(L_r, L_r, 1)$ block-term decomposition is explicitly related to the CP decomposition. We exploited this link to design an efficient algorithm to calculate the rank-$(L_r, L_r, 1)$ decomposition. However, we only considered one algorithm, but there are many more algorithms for the CP decomposition that can potentially inspire researchers to design new fast and accurate schemes for the rank-$(L_r, L_r, 1)$ and coupled rank-$(L_r, L_r, 1)$ block-term decompositions. For example, the generalized eigenspace decomposition (GESD)-based algorithm in [EVD22] offers a good complexity-accuracy performance for the CP decomposition, and its extension to block-term decompositions might be promising. Moreover, as we have already

mentioned in Chapter 7, an interesting direction for further research in the scope of the proposed SECSI-BTD algorithm would be the implementation of more efficient algorithms for the join diagonalization or better clustering approaches to improve the computational-time performance of the proposed algorithm.

In this thesis, we have investigated the rank-$(L_r, L_r, 1)$ block-term decomposition and its applications. However, as we also discussed in Chapter 2, there are other types of block-term decompositions, such as rank-(L, M, N) and rank-$(L, M, \cdot)$ BTDs. In contrast to the rank-$(L_r, L_r, 1)$ BTD, these can be related more to the Tucker decomposition. To the best of our knowledge, the rank-(L, M, N) and the rank-$(L, M, \cdot)$ BTDs have not been studied very extensively in the literature. This opens up a huge space for exploration, including but not limited to efficient computational algorithms, applications, study of the uniqueness properties, and performance analysis.

As has been mentioned in Chapter 9, with the technology moving towards higher frequency bands and extremely large aperture arrays, the electromagnetic diffraction field moves from the far-field region to the near-field. Since the far-field assumptions do not apply anymore, this opens a huge space for exploring novel near-field signal processing techniques for wireless communications and sensing. Compared to the far-field design, the near-field allows exploiting the distance information. Therefore, near-field beamforming can yield higher spatial multiplexing by the transmission of different data to users located at different locations in the same direction. In the near-field you can even transmit multiple streams to the same user in a line-of-sight scenario. Moreover, the near-field is highly beneficial in radar systems: in contrast to far-field systems, it allows a more accurate localization with not only angle but also distance parameters. Therefore, we consider tensor-based signal processing for near-field communications and sensing as a highly attractive and promising topic.

In Chapter 10, we have considered the application of the BTD to EEG-MEG data analysis. Processing these data is often challenging due to their high dimensionality and non-stationary nature. This, therefore, reveals the following research directions in the area: design of computationally effective algorithms that can handle big data tensors, exploration of the tensor decompositions for artifact removal, feature extraction, and noise removal. In general, due to its multidimensional nature, biomedical signal processing is an excellent application for tensor decompositions.

Lastly, we obviously should mention the application of tensors in deep learning as an inevitable future research direction [SDF+17]. Tensors enable the efficient and scalable computation of complex mathematical operations such as convolutions or matrix multiplications [FBH+22]. Additionally, deep learning models often involve large-scale

and high-dimensional data, which can be efficiently represented and processed using tensors. We have not touched this field in this thesis but find it a very attractive future research direction.

We hope this chapter can also invite fellow researchers to further advance our findings by investigating these unexplored research directions.

Part V.

Appendix

Appendices

A

A.1 Matrix products

In this appendix, we have summarized some of the matrix product definitions used in the thesis.

Kronecker product. Given two matrices $\boldsymbol{A} \in \mathbb{F}^{M \times N}$ and $\boldsymbol{B} \in \mathbb{F}^{P \times Q}$, the Kronecker product $\boldsymbol{A} \otimes \boldsymbol{B}$ is defined as

$$\boldsymbol{A} \otimes \boldsymbol{B} = \begin{bmatrix} a_{1,1}\boldsymbol{B} & a_{1,2}\boldsymbol{B} & \dots & a_{1,N}\boldsymbol{B} \\ a_{2,1}\boldsymbol{B} & a_{2,2}\boldsymbol{B} & \dots & a_{2,N}\boldsymbol{B} \\ \vdots & \vdots & \vdots & \vdots \\ a_{M,1}\boldsymbol{B} & a_{M,2}\boldsymbol{B} & \dots & a_{M,N}\boldsymbol{B} \end{bmatrix} \in \mathbb{F}^{PM \times QN}. \tag{A.1}$$

Column-wise Khatri-Rao product. Given two matrices $\boldsymbol{A} = \begin{bmatrix} \boldsymbol{a}_1 & \dots & \boldsymbol{a}_N \end{bmatrix} \in \mathbb{F}^{M \times N}$ and $\boldsymbol{B} = \begin{bmatrix} \boldsymbol{b}_1 & \dots & \boldsymbol{b}_N \end{bmatrix} \in \mathbb{F}^{P \times N}$, the column-wise Khatri-Rao (column-wise Kronecker) product $\boldsymbol{A} \diamond \boldsymbol{B}$ is defined as

$$\boldsymbol{A} \diamond \boldsymbol{B} = \begin{bmatrix} \boldsymbol{a}_1 \otimes \boldsymbol{b}_1 & \boldsymbol{a}_2 \otimes \boldsymbol{b}_2 & \dots & \boldsymbol{a}_N \otimes \boldsymbol{b}_N \end{bmatrix} \in \mathbb{F}^{MP \times N}. \tag{A.2}$$

Partition-wise Khatri-Rao product. Given two partitioned matrices $\boldsymbol{A} = [\boldsymbol{A}_1, \dots, \boldsymbol{A}_R] \in \mathbb{F}^{M \times \sum_{i=r}^{R} N_r}$ and $\boldsymbol{B} = [\boldsymbol{B}_1, \dots, \boldsymbol{B}_R] \in \mathbb{F}^{P \times \sum_{i=r}^{R} L_r}$ composed of R submatrices $\boldsymbol{A}_r \in \mathbb{F}^{M \times N_r}$ and R submatrices $\boldsymbol{B}_r \in \mathbb{F}^{P \times L_r}$, a partition-wise Kronecker product $\boldsymbol{A} \diamond_{\mathrm{s}} \boldsymbol{B}$ is expressed as

$$\boldsymbol{A} \diamond_{\mathrm{s}} \boldsymbol{B} = [\boldsymbol{A}_1 \otimes \boldsymbol{B}_1, \dots, \boldsymbol{A}_R \otimes \boldsymbol{B}_R] \in \mathbb{F}^{MP \times \sum_{i=r}^{R} N_r L_r}. \tag{A.3}$$

When R is equal to the number of columns in $\boldsymbol{A}$ and $\boldsymbol{B}$, the partition-wise Kronecker product is equal to the column-wise Khatri-Rao product.

Hadamard product. Hadamard (element-wise) product between two matrices $\boldsymbol{A} \in \mathbb{F}^{M\times N}$ and $\boldsymbol{B} \in \mathbb{F}^{M\times N}$ of the same dimensions is denoted as

$$\boldsymbol{A} \odot \boldsymbol{B}, \quad \text{and} \quad (\boldsymbol{A} \odot \boldsymbol{B})_{(i,j)} = \boldsymbol{A}_{(i,j)} \cdot \boldsymbol{B}_{(i,j)}. \tag{A.4}$$

Element-wise division. Inverse Hadamard product (element-wise division) between two matrices $\boldsymbol{A} \in \mathbb{F}^{M\times N}$ and $\boldsymbol{B} \in \mathbb{F}^{M\times N}$ of the same dimensions, is denoted as

$$\boldsymbol{A} \oslash \boldsymbol{B}, \quad \text{and} \quad (\boldsymbol{A} \oslash \boldsymbol{B})_{(i,j)} = \frac{\boldsymbol{A}_{(i,j)}}{\boldsymbol{B}_{(i,j)}}. \tag{A.5}$$

Outer product. The outer product between two tensors $\boldsymbol{\mathcal{A}} \in \mathbb{F}^{I_1\times\cdots\times I_M}$ and $\boldsymbol{\mathcal{B}} \in \mathbb{F}^{J_1\times\cdots\times J_N}$ is defined as

$$(\boldsymbol{\mathcal{A}} \circ \boldsymbol{\mathcal{B}})_{(i_1,\ldots,i_M,j_1,\ldots,j_N)} = \boldsymbol{\mathcal{A}}_{(i_1,\ldots,i_M)}\boldsymbol{\mathcal{B}}_{(j_1,\ldots,j_N)} \in \mathbb{F}^{I_1\times\cdots\times I_M\times J_1\times\cdots\times J_N}. \tag{A.6}$$

For the properties, relations between different products, and further details, we refer the reader to [PP12; LT08; DeL08a; GVL13].

A.2 Orthogonal Procrustes problem

The orthogonal Procrustes problem is a least squares problem of transforming a matrix $\boldsymbol{B} \in \mathbb{F}^{M\times P}$ into a matrix $\boldsymbol{A} \in \mathbb{F}^{M\times P}$ by an orthogonal transformation matrix $\boldsymbol{Q} \in \mathbb{F}^{P\times P}$ such that $\|\boldsymbol{A} - \boldsymbol{B}\boldsymbol{Q}\|_{\mathrm{F}}^2$ is minimized [Sch66; GVL13]

$$\arg\min_{\boldsymbol{Q}} \ \|\boldsymbol{A} - \boldsymbol{B}\boldsymbol{Q}\|_{\mathrm{F}}^2 \quad \text{s.t.} \quad \boldsymbol{Q}^{\mathrm{H}}\boldsymbol{Q} = \boldsymbol{I}_P. \tag{A.7}$$

Solution. Rewriting (A.7), we minimize

$$\begin{aligned} \operatorname{tr}\left((\boldsymbol{A} - \boldsymbol{B}\boldsymbol{Q})^{\mathrm{H}}(\boldsymbol{A} - \boldsymbol{B}\boldsymbol{Q})\right) &= \operatorname{tr}\left(\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A} - \boldsymbol{A}^{\mathrm{H}}\boldsymbol{B}\boldsymbol{Q} - \boldsymbol{Q}^{\mathrm{H}}\boldsymbol{B}^{\mathrm{H}}\boldsymbol{A} + \boldsymbol{Q}^{\mathrm{H}}\boldsymbol{B}^{\mathrm{H}}\boldsymbol{B}\boldsymbol{Q}\right) \\ &= \operatorname{tr}\left(\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}\right) + \operatorname{tr}\left(\boldsymbol{B}^{\mathrm{H}}\boldsymbol{B}\right) - 2\operatorname{Re}\left\{\operatorname{tr}\left(\boldsymbol{Q}^{\mathrm{H}}\boldsymbol{B}^{\mathrm{H}}\boldsymbol{A}\right)\right\}. \end{aligned} \tag{A.8}$$

Since $\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}$ and $\boldsymbol{B}^{\mathrm{H}}\boldsymbol{B}$ are fixed, the minimization of (A.8) corresponds to the following maximization problem

$$\arg\max_{\boldsymbol{Q}} \ \operatorname{tr}\left(\boldsymbol{Q}^{\mathrm{H}}\boldsymbol{B}^{\mathrm{H}}\boldsymbol{A}\right) \quad \text{s.t.} \quad \boldsymbol{Q}^{\mathrm{H}}\boldsymbol{Q} = \boldsymbol{I}_P. \tag{A.9}$$

If we express the singular value decomposition (SVD) of $\boldsymbol{S} = \boldsymbol{B}^{\mathrm{H}}\boldsymbol{A}$ as

$$\boldsymbol{S} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{H}} \tag{A.10}$$

and introduce $\boldsymbol{Z} = \boldsymbol{V}^{\mathrm{H}}\boldsymbol{Q}^{\mathrm{H}}\boldsymbol{U}$, we have

$$\mathrm{tr}\left(\boldsymbol{Q}^{\mathrm{H}}\boldsymbol{B}^{\mathrm{H}}\boldsymbol{A}\right) = \mathrm{tr}\left(\boldsymbol{Q}^{\mathrm{H}}\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{H}}\right) = \mathrm{tr}\left(\boldsymbol{Z}\boldsymbol{\Sigma}\right) = \sum_{i=1}^{P} z_{i,i}\sigma_i \leq \sum_{i=1}^{P} \sigma_i. \tag{A.11}$$

Since the matrix $\boldsymbol{Z}$ is a product of orthogonal matrices, $\boldsymbol{Z}$ is also orthogonal and has unit norm columns. Thus, the trace in (A.11) is maximized by setting $\boldsymbol{Z} = \boldsymbol{I}_P$, i.e.,

$$\boldsymbol{Q} = \boldsymbol{U}\boldsymbol{V}^{\mathrm{H}} \tag{A.12}$$

solves the Orthogonal Procrustes problem.

Another solution is proposed by Green [Gre52]

$$\boldsymbol{Q} = \left(\boldsymbol{S}\boldsymbol{S}^{\mathrm{H}}\right)^{-\frac{1}{2}}\boldsymbol{S}, \tag{A.13}$$

which is an equivalent formulation of the SVD solution in (A.12).

Proof. Using the SVD of matrix $\boldsymbol{S}$ in (A.10), we get $\left(\boldsymbol{S}\boldsymbol{S}^{\mathrm{H}}\right)^{-\frac{1}{2}} = \boldsymbol{U}\boldsymbol{\Sigma}^{-1}\boldsymbol{U}^{\mathrm{H}}$. Moreover, we can write the identity matrix as [Gre52; Sch66]

$$\boldsymbol{I}_P = \boldsymbol{\Sigma}^{-1}\boldsymbol{U}^{\mathrm{H}}\boldsymbol{U}\boldsymbol{\Sigma}. \tag{A.14}$$

Consequently, we can express (A.12) as

$$\begin{aligned}\boldsymbol{Q} &= \boldsymbol{U}\boldsymbol{V}^{\mathrm{H}} = \boldsymbol{U}\left(\boldsymbol{\Sigma}^{-1}\boldsymbol{U}^{\mathrm{H}}\boldsymbol{U}\boldsymbol{\Sigma}\right)\boldsymbol{V}^{\mathrm{H}} \\ &= \left(\boldsymbol{U}\boldsymbol{\Sigma}^{-1}\boldsymbol{U}^{\mathrm{H}}\right)\left(\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{H}}\right) \\ &= \left(\boldsymbol{S}\boldsymbol{S}^{\mathrm{H}}\right)^{-\frac{1}{2}}\boldsymbol{S},\end{aligned} \tag{A.15}$$

which is the solution in (A.13).

A.3 Appendix for Chapter 3

Exploiting the Khatri-Rao product with an identity matrix, the least squares solution for the matrix $\boldsymbol{A}$ in (3.19), can be expressed as

$$\boldsymbol{A} = [\widetilde{\boldsymbol{\mathcal{H}}}]_{(1)} \left(\text{bdiag}\left(\boldsymbol{C}(:,1), \ldots, \boldsymbol{C}(:,Q)\right)\right)^{\text{T}+} \tag{A.16}$$

Then, using the Moore-Penrose pseudo-inverse of the block-diagonal matrix, we can rewrite (A.16) as follows

$$\begin{aligned} \boldsymbol{A} &= [\widetilde{\boldsymbol{\mathcal{H}}}]_{(1)} \Big(\left[1 \oslash \text{diag}\left(\|\boldsymbol{C}(:,1)\|^2, \ldots, \|\boldsymbol{C}(:,Q)\|^2 \right)\right] && \text{(A.17)} \\ &\quad \cdot \text{bdiag}\left(\boldsymbol{C}(:,1), \ldots, \boldsymbol{C}(:,Q)\right)^{\text{H}} \Big)^{\text{T}} && \text{(A.18)} \\ &= [\widetilde{\boldsymbol{\mathcal{H}}}]_{(1)} \left(\text{bdiag}\left(\frac{\boldsymbol{C}(:,1)^{\text{H}}}{\|\boldsymbol{C}(:,1)\|^2}, \ldots, \frac{\boldsymbol{C}(:,Q)^{\text{H}}}{\|\boldsymbol{C}(:,Q)\|^2} \right) \right)^{\text{T}}, && \text{(A.19)} \end{aligned}$$

where $\oslash$ denotes the element-wise division. With the identity on the right-hand side of the Khatri-Rao product, equation (3.21) can be rewritten as follows

$$\boldsymbol{C} = [\widetilde{\boldsymbol{\mathcal{H}}}]_{(3)} \left(\begin{bmatrix} \text{diag}\left(\boldsymbol{A}(1,:)\right) \\ \vdots \\ \text{diag}\left(\boldsymbol{A}(Q,:)\right) \end{bmatrix} \right)^{\text{T}+}. \tag{A.20}$$

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

$$\begin{aligned} \boldsymbol{C} &= [\widetilde{\boldsymbol{\mathcal{H}}}]_{(3)} \left(\left[1 \oslash \text{diag}\left(\|\boldsymbol{A}(:,1)\|^2, \ldots, \|\boldsymbol{A}(:,Q)\|^2 \right)\right] \cdot \begin{bmatrix} \text{diag}\left(\boldsymbol{A}(1,:)\right) \\ \vdots \\ \text{diag}\left(\boldsymbol{A}(Q,:)\right) \end{bmatrix}^{\text{H}} \right)^{\text{T}} \\ &= [\widetilde{\boldsymbol{\mathcal{H}}}]_{(3)} \left(\left[\text{diag}\left(\frac{\boldsymbol{A}(1,:)^{\text{H}}}{\|\boldsymbol{A}(:,1)\|^2} \right), \ldots, \text{diag}\left(\frac{\boldsymbol{A}(Q,:)^{\text{H}}}{\|\boldsymbol{A}(:,Q)\|^2} \right) \right] \right)^{\text{T}} \end{aligned} \tag{A.21}$$

A.4 Uniqueness conditions for rank-$(L_r, L_r, 1)$ block-term decomposition

This appendix summarizes some of the uniqueness conditions from [DD20; DeL08a; DVD18] under which the rank-$(L_r, L_r, 1)$ block-term decomposition (BTD) is generically unique [DD20].

Consider a decomposition of tensor $\mathcal{T} \in \mathbb{F}^{I\times J\times K}$ into a sum of R rank-$(L_r, L_r, 1)$ terms

$$\mathcal{T} = \sum_{r=1}^{R} \left(\boldsymbol{A}_r \cdot \boldsymbol{B}_r^{\mathrm{T}}\right) \circ \boldsymbol{c}_r, \tag{A.22}$$

where $\boldsymbol{A}_r \in \mathbb{F}^{I\times L_r}$, $\boldsymbol{B}_r \in \mathbb{F}^{J\times L_r}$, and $\boldsymbol{c}_r \in \mathbb{F}^{K\times R}$, $r \in \{1,\ldots,R\}$, are the first, the second, and the third factor matrices of the rank-$(L_r, L_r, 1)$ block-term decomposition, respectively. The matrices $\boldsymbol{A}_r$ and $\boldsymbol{B}_r$ are of rank at most L_r.

The following presents five sets of bounds on the R and the L_rs under which the decomposition in (A.22) is generically unique (assuming $L_1 \leq \ldots \leq L_R$ and $\min\left(I, J, K, R\right) \geq 2$) [DD20].

1. The first and second factors have full-column rank [DeL08a]

$$I \geq \sum L_r, \quad J \geq \sum L_r \tag{A.23}$$

2. The third and second factors have full-column rank [DVD18]

$$K \geq R, \quad J \geq \sum L_r, \tag{A.24}$$

and

$$I \geq L_R + 1 \tag{A.25}$$

3. The third factor has full-column rank [DeL08a]

$$K \geq R, \tag{A.26}$$

and

$$k'_{\boldsymbol{A}} + k'_{\boldsymbol{B}} \geq R + 2, \tag{A.27}$$

where $k'_{\boldsymbol{X}}$ denotes the k'-rank of a partitioned matrix $\boldsymbol{X}$, which is maximum number r such that any set of r submatrices of $\boldsymbol{X}$ yields a set of linearly independent columns.

4. $I \geq L_2 + \ldots + L_R + 1$ and $J \geq L_{\min(K,R)-1} + \ldots + L_R$ [DD20]

5. The first factor has full-column rank [DD20]

$$I \geq \sum L_r, \tag{A.28}$$

and

$$\begin{aligned} J &\geq L_{R-1} + L_R, \\ (K-1)(J-1) &\geq \sum L_r. \end{aligned} \tag{A.29}$$

For the detailed derivations, we refer the reader to [DD20; DeL08a; DVD18]

A.5 Nonlinear least squares

In Chapter 7, we employ the nonlinear least squares (NLS) algorithm for the refinement steps in the proposed BTD algorithms. The Tensorlab implementation [SVBD13; VDS+16] that we use is based on the Gauss-Newton method with a dogleg trust region [NW06]. This appendix presents its brief description.

Consider the following nonlinear least squares problem

$$\arg\min_{\boldsymbol{z}} \; f_{\text{BTD}}(\boldsymbol{z}) = \frac{1}{2}\left\|\hat{\boldsymbol{\mathcal{T}}} - \boldsymbol{\mathcal{T}}\right\|_{\text{F}}^2 = \frac{1}{2}\left\|\boldsymbol{\mathcal{F}}\right\|_{\text{F}}^2 \tag{A.30}$$

where $\hat{\boldsymbol{\mathcal{T}}}$ is a tensor reconstructed from the estimated BTD factors as $\hat{\boldsymbol{\mathcal{T}}} = \sum_{r=1}^{R} \left(\tilde{\boldsymbol{A}}_r \cdot \tilde{\boldsymbol{B}}_r^{\mathrm{T}}\right) \circ \tilde{\boldsymbol{c}}_r$. For notational simplicity, we omit the $\sim$ in the matrices $\boldsymbol{A}_r$, $\boldsymbol{B}_r$, and $\boldsymbol{c}_r$, and express them as $\boldsymbol{A} = [\boldsymbol{A}_1, \boldsymbol{A}_2, ..., \boldsymbol{A}_R] \in \mathbb{F}^{I \times \Sigma L_r}$, $\boldsymbol{B} = [\boldsymbol{B}_1, \boldsymbol{B}_2, ..., \boldsymbol{B}_R] \in \mathbb{F}^{J \times \Sigma L_r}$, and $\boldsymbol{C} = [\boldsymbol{c}_1, \boldsymbol{c}_2, ..., \boldsymbol{c}_R] \in \mathbb{F}^{K \times R}$, respectively. Consequently, $\boldsymbol{\mathcal{F}}$ is a residual tensor, and $\boldsymbol{z}$ is a vector of unknowns containing the elements of $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}$ in a vectorized form.

In the nonlinear least squares methods such as Gauss-Newton, the residual tensor $\boldsymbol{\mathcal{F}}$ at a current iterate $\mathring{\boldsymbol{z}}_k$ is approximated as

$$m_k^{\boldsymbol{\mathcal{F}}}(\mathring{\boldsymbol{p}}) = \text{vec}\left(\boldsymbol{\mathcal{F}}(\mathring{\boldsymbol{z}}_k)\right) + \frac{\partial \text{vec}\left(\boldsymbol{\mathcal{F}}(\mathring{\boldsymbol{z}}_k)\right)}{\partial \mathring{\boldsymbol{z}}^{\mathrm{T}}} \mathring{\boldsymbol{p}}_k, \tag{A.31}$$

where $\boldsymbol{\mathcal{F}}(\mathring{\boldsymbol{z}}_k)$ is the objective function at a current iterate $\mathring{\boldsymbol{z}}_k$, and the superscript $\circ$ denotes the concatenation of its argument with its complex conjugate, i.e., $\mathring{\boldsymbol{z}} = (\boldsymbol{z}, \boldsymbol{z}^*)$ (we use $(\boldsymbol{a}, \boldsymbol{b})$ to denote the concatenation $\left[\boldsymbol{a}^{\mathrm{T}}, \boldsymbol{b}^{\mathrm{T}}\right]^{\mathrm{T}}$) [SvD12]. The matrix $\frac{\partial \text{vec}(\boldsymbol{\mathcal{F}})}{\partial \mathring{\boldsymbol{z}}^{\mathrm{T}}}$ is a complex Jacobian $\boldsymbol{J}_k$ at $\mathring{\boldsymbol{z}}_k$ [SvD12] and is composed of two parts $\frac{\partial \text{vec}(\boldsymbol{\mathcal{F}})}{\partial \mathring{\boldsymbol{z}}^{\mathrm{T}}} = \left[\begin{array}{cc} \frac{\partial \text{vec}(\boldsymbol{\mathcal{F}})}{\partial \boldsymbol{z}^{\mathrm{T}}}, & \frac{\partial \text{vec}(\boldsymbol{\mathcal{F}})}{\partial \boldsymbol{z}^{*\mathrm{T}}} \end{array}\right]$ obtained via transpose of the complex gradients of the components of $\text{vec}(\boldsymbol{\mathcal{F}})$ [SvD12]. The complex gradient at $\mathring{\boldsymbol{z}}_k$ contains two parts, the cogradient $\frac{\partial f}{\partial \boldsymbol{z}}$ and the conjugate cogradient $\frac{\partial f}{\partial \boldsymbol{z}^*}$ (Wirtinger calculus [Hjø11]).

In case of the rank-$(L_r, L_r, 1)$ BTD, the complex cogradient of the objective function in (A.30) is expressed as [SVBD13]

$$\frac{\partial f_{\text{BTD}}}{\partial \boldsymbol{z}} = \frac{1}{2}\begin{bmatrix} \operatorname{vec}\left(\boldsymbol{A}^* \left[\left(\boldsymbol{B}^{\text{H}}\boldsymbol{B}\right) \odot \left(\boldsymbol{D}^{\text{T}}\boldsymbol{C}^{\text{H}}\boldsymbol{C}\boldsymbol{D}\right)\right] - \left[\boldsymbol{\mathcal{T}}\right]^*_{(1)} \left((\boldsymbol{C}\boldsymbol{D}) \diamond \boldsymbol{B}\right)\right) \\ \operatorname{vec}\left(\boldsymbol{B}^* \left[\left(\boldsymbol{A}^{\text{H}}\boldsymbol{A}\right) \odot \left(\boldsymbol{D}^{\text{T}}\boldsymbol{C}^{\text{H}}\boldsymbol{C}\boldsymbol{D}\right)\right] - \left[\boldsymbol{\mathcal{T}}\right]^*_{(2)} \left((\boldsymbol{C}\boldsymbol{D}) \diamond \boldsymbol{A}\right)\right) \\ \operatorname{vec}\left(\boldsymbol{C}^* \boldsymbol{D} \left[\left(\boldsymbol{A}^{\text{H}}\boldsymbol{A}\right) \odot \left(\boldsymbol{B}^{\text{H}}\boldsymbol{B}\right)\right] \boldsymbol{D}^{\text{T}} - \left[\boldsymbol{\mathcal{T}}\right]^*_{(3)} \left(\boldsymbol{B} \diamond \boldsymbol{A}\right) \boldsymbol{D}^{\text{T}}\right) \end{bmatrix}, \tag{A.32}$$

where the matrix $\boldsymbol{D}$ is a $R \times \sum_{r=1}^{R} L_r$ block-diagonal matrix $\operatorname{bdiag}\left(\mathbf{1}_{L_1}^{\text{T}}, \ldots, \mathbf{1}_{L_R}^{\text{T}}\right)$ with the rth block on the diagonal is the row vector $\mathbf{1}_{L_r}^{\text{T}}$. For the real-valued BTD decomposition, the real gradient is twice the expression for the complex cogradient [SvD12].

Consequently, the quadratic model of the objective function $f_{\text{BTD}}(\boldsymbol{z})$ in (A.30) can be written as

$$m_k^f(\mathring{\boldsymbol{p}}) = \frac{1}{2}\left\| m_k^{\boldsymbol{\mathcal{F}}}(\mathring{\boldsymbol{p}}) \right\|^2 \tag{A.33}$$

where $\boldsymbol{p}$ is the search direction. The control of the length and the direction of the step can be done by using the trust-region frameworks. Then, the search direction is obtained by solving the following system

$$\boldsymbol{J}_k^{\text{H}} \boldsymbol{J}_k \boldsymbol{p}_k^* = -\boldsymbol{J}_k^{\text{H}} \operatorname{vec}\left(\boldsymbol{\mathcal{F}}(\boldsymbol{z}_k)\right) = -2\frac{\partial f_{\text{BTD}}}{\partial \boldsymbol{z}^*}(\boldsymbol{z}_k), \tag{A.34}$$

where $\boldsymbol{J}_k^{\text{H}} \boldsymbol{J}_k$ is an approximation of the objective function's Hessian (one of the advantages of the Gauss-Newton over the plain Newton's method).

In line search methods, the next iterate is calculated as $\boldsymbol{z}_{k+1} = \boldsymbol{z}_k + \alpha_k \boldsymbol{p}_k$, where the real step length α_k is commonly chosen to satisfy the Wolfe conditions [NW06].

Trust-region methods define a region with radius Δ_k around the current iterate $\boldsymbol{z}_k$ in which a behavior of model function m_k is similar to that of the actual objective function f [NW06]. The next iterate $\boldsymbol{z}_{k+1}$ is then chosen to be the approximate minimizer of the model in this region. Therefore, in trust-region methods, the direction and length of the step are chosen simultaneously. The trust-region radius Δ_k is updated every iteration based on the trustworthiness ρ_k of the model, which is defined as the ratio of the actual reduction $f(\mathring{\boldsymbol{z}}_k) - f\left(\mathring{\boldsymbol{z}}_k + \mathring{\boldsymbol{p}}_k\right)$ of the objective function and the predicted reduction $m_k^f(\mathbf{0}) - m_k^f\left(\mathring{\boldsymbol{p}}_k\right)$. The dogleg method [Pow70] can be employed when the

model's Hessian (or its approximation) is a positive definite matrix. In a dogleg method, the update step $\boldsymbol{p}_k$ is selected as

$$\boldsymbol{p}_k = \begin{cases} \boldsymbol{p}_{k,\mathrm{GN}} & \text{if } \left\|\boldsymbol{p}_{k,\mathrm{GN}}\right\| \leqslant \Delta_k, \\ \frac{\Delta_k}{\|\boldsymbol{p}_{k,\mathrm{sd}}\|^2}\boldsymbol{p}_{k,\mathrm{sd}} & \text{if } \left\|\boldsymbol{p}_{k,GN}\right\| > \Delta_k \text{ and } \left\|\boldsymbol{p}_{k,SD}\right\| \geqslant \Delta_k, \\ \boldsymbol{p}_{k,\mathrm{sd}} + \beta_k\left(\boldsymbol{p}_{k,\mathrm{GN}} - \boldsymbol{p}_{k,\mathrm{sd}}\right) & \text{if } \left\|\boldsymbol{p}_k, GN\right\| > \Delta_k \text{ and } \left\|\boldsymbol{p}_{k,\mathrm{sd}}\right\| < \Delta_k \quad \text{(dogleg step)}, \end{cases} \tag{A.35}$$

where $\boldsymbol{p}_{k,\mathrm{GN}}$ and $\boldsymbol{p}_{k,\mathrm{sd}}$ are the Gauss-Newton and the steepest descent steps. The quantity β_k ensures that $\left\|\boldsymbol{p}_k\right\| \leq \Delta_k$.

There are also other approaches to solve (A.30) by means of nonlinear least squares methods, for example, the Levenberg-Marquardt, an extension of the Gauss-Newton method [SVBD13; NW06].

A.6 Coordinate systems

The following introduces the coordinate systems used in the thesis (Chapter 9) and their transformations.

We consider two common coordinate systems, Cartesian and spherical, where the position of a point p is defined by a vector $\boldsymbol{r}$ with three parameters, $\{x, y, z\}$ and $\{\rho, \phi, \theta\}$ (range, azimuth, and elevation), respectively. The illustration of the parameters is given in Figure A.1. Note that we define the elevation angle as an angle above (positive angle) or below (negative angle) the x-y plane.

The conversion from spherical coordinates to 3D Cartesian coordinates and vice versa is expressed as

$$\begin{aligned} x &= \rho\cos(\theta)\cos(\phi), \\ y &= \rho\cos(\theta)\sin(\phi), \\ z &= \rho\sin(\theta), \end{aligned} \tag{A.36}$$

and

$$\begin{aligned} \phi &= \operatorname{atan2}(y, x), \\ \theta &= \operatorname{atan2}(z, \sqrt{x^2 + y^2}), \\ \rho &= \sqrt{x^2 + y^2 + z^2}, \end{aligned} \tag{A.37}$$

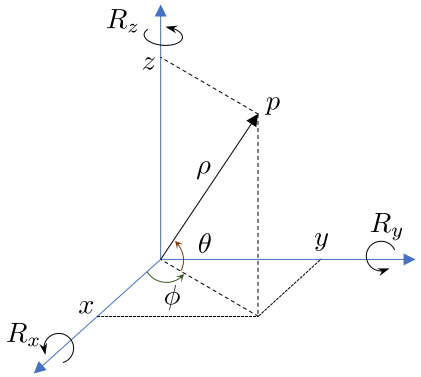


Figure A.1. 3D coordinate system and the axes rotations.

respectively. The function atan2 denotes the four-quadrant inverse tangent [DI15] defined as

$$\operatorname{atan2}(y,x) = \begin{cases} \arctan\left(\frac{y}{x}\right) & \text{if } x > 0, \\ \frac{\pi}{2} - \arctan\left(\frac{x}{y}\right) & \text{if } y > 0, \\ -\frac{\pi}{2} - \arctan\left(\frac{x}{y}\right) & \text{if } y < 0, \\ \arctan\left(\frac{y}{x}\right) \pm \pi & \text{if } x < 0, \\ \text{undefined} & \text{if } x = 0 \text{ and } y = 0. \end{cases} \tag{A.38}$$

Another useful transformation of a coordinate system is a rotation. A basic 3D rotation is a rotation about the x, y, and z- axes of a coordinate system. Each rotation is specified by an angle of rotation. Commonly, the rotation angle is positive for a counterclockwise rotation (when looking along the rotation axis towards the origin).

The rotation matrices that rotate a coordinates vector counterclockwise around the x, y, and z-axes are expressed as

$$\boldsymbol{R}_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix}, \tag{A.39a}$$

$$\boldsymbol{R}_y(\beta) = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}, \tag{A.39b}$$

and

$$\boldsymbol{R}_z(\gamma) = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{A.39c}$$

respectively.

Any arbitrary rotation can be composed of a combination of these three (Euler's rotation theorem). A vector can be rotated in any direction using a sequence of three rotations [GPS02]: $\boldsymbol{r}' = \boldsymbol{A}\boldsymbol{r} = \boldsymbol{R}_z(\gamma)\boldsymbol{R}_y(\beta)\boldsymbol{R}_x(\alpha)\boldsymbol{r}$.

A.7 Appendix for Chapter 9

This appendix is a summary of the derivations in [YGL+19]. Considering the noiseless case for simplicity, the received signal $\boldsymbol{Y}_p^{[m_R]} \in \mathbb{C}^{N_R \times T}$ at the m_Rth receiving array in the pth pulse in (9.3) can be rewritten as

$$\boldsymbol{Y}_p^{[m_R]} = \sum_{m_T=1}^{M_T} \sum_{r=1}^{R} \alpha_r^{[m_R]} \varrho_{r,p}^{[m_R,m_T]} \boldsymbol{b}_r^{[m_R]} \boldsymbol{a}_r^{[m_T]\mathrm{T}} \boldsymbol{S}^{[m_T]\mathrm{T}} \in \mathbb{C}^{N_R \times T}. \tag{A.40}$$

By stacking the matrices $\boldsymbol{Y}_p^{[m_R]}$ with fixed m_R and varying p along the 3-mode, the data received at the m_Rth array for P pulses can be represented as a three-dimensional tensor $\boldsymbol{\mathcal{Y}}'^{[m_R]}_{(:,:,p)} = \boldsymbol{Y}_p^{[m_R]} \in \mathbb{C}^{N_R \times T}$. Let $\boldsymbol{f}_r^{[m_R,m_T]} = \left[\varrho_{r,1}^{[m_R,m_T]}, \ \ldots, \ \varrho_{r,P}^{[m_R,m_T]} \right]^{\mathrm{T}} \in \mathbb{C}^P$ and $\boldsymbol{c}_r^{[m_T]} = \boldsymbol{S}^{[m_T]} \boldsymbol{a}_r^{[m_T]} \in \mathbb{C}^T$. Then the tensor $\boldsymbol{\mathcal{Y}}'^{[m_R]}$ can be expressed as

$$\boldsymbol{\mathcal{Y}}'^{[m_R]} = \sum_{l=1}^{M_T} \sum_{r=1}^{R} \alpha_r^{[m_R]} \boldsymbol{b}_r^{[m_R]} \circ \boldsymbol{c}_r^{[m_T]} \circ \boldsymbol{f}_r^{[m_R,m_T]} \in \mathbb{C}^{N_R \times T \times P}. \tag{A.41}$$

After permuting the first and the third indices in $\boldsymbol{\mathcal{Y}}'^{[m_R]}$, it can be rewritten as tensor $\boldsymbol{\mathcal{Y}}^{[m_R]} \in \mathbb{C}^{P \times T \times N_R}$

$$\boldsymbol{\mathcal{Y}}^{[m_R]} = \sum_{r=1}^{R} \alpha_r^{[m_R]} \left(\sum_{l=1}^{M_T} \boldsymbol{f}_r^{[m_R,m_T]} \circ \boldsymbol{c}_r^{[m_T]} \right) \circ \boldsymbol{b}_r^{[m_R]} \in \mathbb{C}^{P \times T \times N_R}. \tag{A.42}$$

Let us express the matrix of reflection coefficients $\boldsymbol{F}_r^{[m_R]}$ as $\boldsymbol{F}_r^{[m_R]} = \left[\boldsymbol{f}_r^{[m_R,1]}, \ \ldots, \ \boldsymbol{f}_r^{[m_R,M_T]} \right] \in \mathbb{C}^{P \times M_T}$ and combine the transmitted signal and

the transmit steering vectors in $\boldsymbol{C}_r = \begin{bmatrix} \boldsymbol{c}_r^{[1]}, & \ldots, & \boldsymbol{c}_r^{[M_T]} \end{bmatrix} \in \mathbb{C}^{T \times M_T}$. Then the tensor $\boldsymbol{\mathcal{Y}}^{[m_R]}$ in (A.42) can be expressed as

$$\boldsymbol{\mathcal{Y}}^{[m_R]} = \sum_{r=1}^{R} \alpha_r^{[m_R]} \left(\boldsymbol{F}_r^{[m_R]} \boldsymbol{C}_r^{\mathrm{T}} \right) \circ \boldsymbol{b}_r^{[m_R]} \in \mathbb{C}^{P \times T \times N_R}, \tag{A.43}$$

where $\boldsymbol{\mathcal{Y}}^{[m_R]} \in \mathbb{C}^{P \times T \times N_R}$ is the received data tensor in (9.8) (in the noiseless case).

A.8 Further reading

Within the thesis, we have explored specific concepts of multilinear algebra and optimization relevant to our discussions. For readers who want to learn more about these subjects, we have compiled our selection of overview papers and textbooks on linear algebra, tensor decompositions, and numerical optimization below.

Reference	Description
	Overview Papers on Different Tensor Decompositions
[KB09]	Overview of basic operators and notations of multilinear algebra, tensor decompositions (Canonical Polyadic (CP), Tucker, and their extensions), and applications.
[CMD+15]	Overview of tensor decompositions with various application examples
[SDF+17]	Survey paper on tensor decompositions for signal processing and machine learning with a focus on the decomposition algorithms.
[SVBD15]	Algorithms for structured data fusion (coupled tensor and matrix factorizations).
	Selected decompositions
[DeL08b], [DeL08a], [DN08]	Overview of block-term decompositions, their uniqueness properties and computation
[DD20], [SVBD13]	Uniqueness and computation of rank-$(L_r, L_r, 1)$ block-term decomposition, link to CP decomposition.

Continued on the next page

[SD15], [SDD15]	Uniqueness and computation of the coupled CP and coupled rank-$(L_r, L_r, 1)$ decompositions.
[FA14], [CLA09]	Overview, algorithms, and discussion on CP and constrained CP decompositions.
[KBB99]	PARAFAC2 and its computation via direct fitting algorithm.
[DDV00a]	Computation and properties of the multilinear generalization of the singular value decomposition (HOSVD).
	Textbooks
[GVL13]	Theory on linear and multilinear algebra, matrix decompositions, products, and operators with focus on linear systems and efficient computations.
[BV04]	Comprehensive introduction to convex optimization with theory, applications, algorithms, and examples.
[BV18a]	Introduction to linear algebra and least squares with straightforward explanations and examples.
[Str19]	Linear algebra for data science: fundamental concepts, important matrix factorizations, basics on probability, statistics, and optimization.
[NW06]	Well-balanced book on fundamentals of numerical optimization with a comprehensive description of different methods to solve optimization problems.
[Bis06], [Mur12], [GBC16]	Comprehensive introduction to machine learning with fundamentals of probability theory and a broad scope of other machine learning related topics.
[PP12]	Quick reference on matrix-related products and operators
	Others
[LT08], [LC18]	Tensor-related operators and products.
[SvD12]	Nonlinear optimization and nonlinear least squares methods for the optimization of real functions in complex variables

Table A.1. Further reading.

Notation

Symbols & Notation

e, π, j	Euler's number, π, and imaginary unit: $\mathrm{e}^{j\pi} + 1 = 0$
$\mathbb{R}$	Set of real numbers
$\mathbb{C}$	Set of complex numbers
$\mathbb{F}$	Set of real or complex numbers (when the difference is unimportant)
a	A scalar
$\boldsymbol{a}$	A column vector
$\boldsymbol{A}$	A matrix
$\boldsymbol{\mathcal{A}}$	A tensor
$\boldsymbol{a}_{(i)}$, $(\boldsymbol{a})_{(i)}$, a_i	The ith element of the vector $\boldsymbol{a} \in \mathbb{F}^I$
$\boldsymbol{A}_{(i,j)}$, $(\boldsymbol{A})_{(i,j)}$, $a_{i,j}$	The (i,j)th element of the matrix $\boldsymbol{A} \in \mathbb{F}^{I \times J}$
$\boldsymbol{\mathcal{A}}_{(i,j,k)}$, $(\boldsymbol{\mathcal{A}})_{(i,j,k)}$, $a_{i,j,k}$	The (i,j,k)th element of the tensor $\boldsymbol{\mathcal{A}} \in \mathbb{F}^{I \times J \times K}$
$\boldsymbol{A}_{(i,:)}$	The ith row of the matrix $\boldsymbol{A}$
$\boldsymbol{A}_{(:,j)}$	The jth column of the matrix $\boldsymbol{A}$
$\{\boldsymbol{A}\}_{\mathcal{C}(\mathcal{Q})}$	Columns of the matrix $\boldsymbol{A}$ with the indices from the set $\mathcal{Q}$
$\{\boldsymbol{A}\}_{\mathcal{R}(\mathcal{Q})}$	Rows of the matrix $\boldsymbol{A}$ with the indices from the set $\mathcal{Q}$
$\mathbf{1}_N$	Column vector of ones of size N
$\boldsymbol{O}_{M \times N}$	The $M \times N$ matrix of zeros
$\boldsymbol{I}_M$	An identity matrix of size $M \times M$
$\boldsymbol{\mathcal{I}}_{N,R}$	An N-way super-diagonal tensor of size $R \times R \times \ldots \times R$ with ones along its super-diagonal
$\boldsymbol{A}^{\mathrm{T}}$	Transpose of $\boldsymbol{A}$
$\boldsymbol{A}^{\mathrm{H}}$	Hermitian transpose of $\boldsymbol{A}$
$\boldsymbol{A}^{-1}$	Inverse of $\boldsymbol{A}$
$\boldsymbol{A}^{+}$	Moore-Penrose pseudoinverse of $\boldsymbol{A}$
$\boldsymbol{A}^{\mathrm{H}-}$	Hermitian transpose and inverse: $\left(\boldsymbol{A}^{\mathrm{H}}\right)^{-1} = \left(\boldsymbol{A}^{-1}\right)^{\mathrm{H}}$
$\boldsymbol{A}^{\mathrm{T}+}$	Transpose and Moore-Penrose pseudoinverse: $\left(\boldsymbol{A}^{\mathrm{T}}\right)^{+} = \left(\boldsymbol{A}^{+}\right)^{\mathrm{T}}$
$\left[\boldsymbol{\mathcal{A}}\right]_{(n)}$	The n-mode unfolding of the tensor $\boldsymbol{\mathcal{A}}$ (Section 2.1)

$\mathrm{vec}(\boldsymbol{A})$, $\mathrm{vec}(\boldsymbol{\mathcal{A}})$	Transformation of the matrix $\boldsymbol{A}$ or the tensor $\boldsymbol{\mathcal{A}}$ into a column vector
$\mathrm{diag}(\boldsymbol{a})$	A square diagonal matrix with diagonal elements being the entries of the input vector
$\mathrm{diag}(\boldsymbol{A})$	A vector containing the diagonal elements of the matrix $\boldsymbol{A}$
$\mathrm{bdiag}\{\boldsymbol{A}_1,\ldots,\boldsymbol{A}_N\}$	A block-diagonal matrix with the input matrices $\boldsymbol{A}_1,\ldots,\boldsymbol{A}_N$ on the main diagonal
$\mathrm{rank}(\boldsymbol{A})$	Rank of a matrix $\boldsymbol{A}$
$\mathrm{rank}_n(\boldsymbol{\mathcal{A}})$	n-rank of a tensor $\boldsymbol{\mathcal{A}}$
$\mathrm{tr}(\boldsymbol{A})$	Trace of the matrix $\boldsymbol{A}$ (sum of diagonal elements)
$\mathrm{Re}\{x\}$	Real part of complex variable x
$\mathrm{Im}\{x\}$	Imaginary part of complex variable x
$\lvert x\rvert$	Absolute value of complex variable x
$\det(\boldsymbol{A})$	Determinant of the matrix $\boldsymbol{A}$
$\min(\cdot)$, $\max(\cdot)$	Minimum, maximum
z^*	Complex conjugate of z
$\boldsymbol{A}\otimes\boldsymbol{B}$	Kronecker product between $\boldsymbol{A}\in\mathbb{F}^{M\times N}$ and $\boldsymbol{B}\in\mathbb{F}^{P\times Q}$ (Appendix A.1)
$\boldsymbol{A}\diamond\boldsymbol{B}$	Column-wise Khatri-Rao (column-wise Kronecker) product between $\boldsymbol{A}\in\mathbb{F}^{M\times N}$ and $\boldsymbol{B}\in\mathbb{F}^{P\times N}$ (Appendix A.1)
$\boldsymbol{A}\diamond_{\mathrm{s}}\boldsymbol{B}$	Partition-wise Khatri-Rao product between $\boldsymbol{A}\in\mathbb{F}^{M\times\sum_{i=r}^{R}N_r}$ and $\boldsymbol{B}\in\mathbb{F}^{P\times\sum_{i=r}^{R}L_r}$ (Appendix A.1)
$\boldsymbol{A}\circ\boldsymbol{B}$	Outer product between matrices (vectors, tensors) $\boldsymbol{A}\in\mathbb{F}^{M\times N}$ and $\boldsymbol{B}\in\mathbb{F}^{P\times Q}$ (Appendix A.1)
$\boldsymbol{A}\odot\boldsymbol{B}$	Hadamard (element-wise) product between two matrices (vectors, tensors) of equal dimensions (Appendix A.1)
$\boldsymbol{A}\oslash\boldsymbol{B}$	Inverse Hadamard product (element-wise division) between two matrices (vectors, tensors) of equal dimensions (Appendix A.1)
$\boldsymbol{\mathcal{A}}\times_n\boldsymbol{U}$	The n-mode product between a tensor $\boldsymbol{\mathcal{A}}$ and a matrix $\boldsymbol{U}$ (Section 2.1)
$\lVert\boldsymbol{a}\rVert$	Euclidean (two-) norm of the vector $\boldsymbol{a}$
$\lVert\boldsymbol{A}\rVert_{\mathrm{F}}$	Frobenius norm of the matrix $\boldsymbol{A}$
$\lVert\boldsymbol{\mathcal{A}}\rVert_{\mathrm{F}}$	Frobenius norm of the tensor $\boldsymbol{\mathcal{A}}$ (see the definition on page 17)
ΣL_r	The sum $\sum_{r=1}^{R}L_r$

Acronyms & Abbreviations

3D	Three-Dimensional
5G	Fifth Generation (of wireless communication systems)
6G	Sixth Generation (of wireless communication systems)
ALS	Alternating Least Squares
AWGN	Additive White Gaussian Noise
BC	Broadcast
BM	Best Matching
BS	Base Station
BTD	Block-Term Decomposition
CANDECOMP	Canonical Decomposition
CCDF	Complementary Cumulative Distribution Functions
CF	Closed Form
COBRAS	COupled Block-term decomposition for multi-static RAdar Systems
CON PS	Condition number - Paired Solutions
CP	Canonical Polyadic
CPD	Canonical Polyadic Decomposition
CS	Common Subspace
CSIT	Channel State Information at the Transmitter
DoF	Degree-of-Freedom
EEG	Electroencephalography
FlexCoBF	Flexible Coordinated Beamforming
GESD	Generalized Eigenspace Decomposition
GEVD	Generalized Eigenvalue Decomposition
GRAD	Gradiometer
GSVD	Generalized Singular Value Decomposition
HO GSVD	Higher Order Generalized Singular Value Decomposition
HOSVD	Higher-Order Singular Value Decomposition
i.i.d.	independent and identically distributed
IC	Interference Channel
JD	Joint Diagonalization
LaRGE	Linear Regression of Global Eigenvalues
LinS	Linear Sum
LS	Least Squares
LSKRF	Least Squares Khatri-Rao Factorization
MAG	Magnetometer
MEG	Magnetoencephalography

MIMO	Multiple-Input Multiple-Output
ML-GSVD	Multilinear Generalized Singular Value Decomposition
MU	Multi-User
NLS	Nonlinear Least-Squares
NOMA	Non-Orthogonal Multiple Access
OFDM	Orthogonal Frequency Division Multiplexing
OMA	Orthogonal Multiple Access
PARAFAC	Parallel Factors
PARAFAC2	Parallel Factors 2
PESDR	Prediction Error to Standard Deviation Ratio
PF	Penalty Function
PS	Private Subspace
QoS	Quality of Service
REC PS	Reconstruction error - Paired Solutions
RES	RESidual
RES BT	Residual-Block-Terms
RMSE	Root Mean Square Error
RSMA	Rate Splitting Multiple Access
SDMA	Space Division Multiple Access
SECSI	Semi-Algebraic framework for approximate CP decomposition via Simultaneous matrix diagonalization
SEF	Somatosensory Evoked Magnetic Fields
SEP	Somatosensory Evoked Electrical Potentials
SFE	Squared Factor Error
SIC	Successive Interference Cancellation
SINR	Signal-to-Interference-Plus-Noise Ratio
SMD	Simultaneous Matrix Diagonalization
SNR	Signal-to-Noise Ratio
SPWV	Smoothed Pseudo Wigner-Ville Distribution
SRE	Squared Reconstruction Error
SVD	Singular Value Decomposition
TC	Tensor Contractions
UE	User Equipment
ULA	Uniform Linear Array
URA	Uniform Rectangular Array
VC	Virtual Channel
ZF	Zero Forcing
ZMCSCG	Zero Mean Circularly Symmetric Complex Gaussian

List of Figures

List of Tables

List of Algorithms

Bibliography

Journal Publications

[KAH22] L. Khamidullina, A. L. F. de Almeida, and M. Haardt, "Multilinear Generalized Singular Value Decomposition (ML-GSVD) and Its Application to Multiuser MIMO Systems", *IEEE Transactions on Signal Processing*, vol. 70, pp. 2783–2797, 2022. DOI: 10.1109/TSP.2022.3178902.

[KPH21] L. Khamidullina, I. Podkurkov, and M. Haardt, "Conditional and Unconditional Cramér-Rao Bounds for Near-Field Localization in Bistatic MIMO Radar Systems", *IEEE Transactions on Signal Processing*, vol. 69, pp. 3220–3234, 2021. DOI: 10.1109/TSP.2021.3082469.

[KSP+23] L. Khamidullina, G. Seidl, I. A. Podkurkov, A. A. Korobkov, and M. Haardt, "Enhanced Solutions for the Block-Term Decomposition in Rank-$(L_r, L_r, 1)$ Terms", *IEEE Transactions on Signal Processing*, vol. 71, pp. 2608–2621, 2023. DOI: 10.1109/TSP.2023.3289730.

[PSK+21] I. Podkurkov, G. Seidl, L. Khamidullina, A. Nadeev, and M. Haardt, "Tensor-Based Near-Field Localization Using Massive Antenna Arrays", *IEEE Transactions on Signal Processing*, vol. 69, pp. 5830–5845, 2021. DOI: 10.1109/TSP.2021.3105797.

Conference Publications

[AKK+22] B. Ahmad, L. Khamidullina, A. A. Korobkov, A. Manina, J. Haueisen, and M. Haardt, "Joint Model Order Estimation for Multiple Tensors with A Coupled Mode and Applications to the Joint Decomposition of EEG, MEG Magnetometer, and Gradiometer Tensors", in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2022)*, May 2022, pp. 1186–1190. DOI: 10.1109/ICASSP43922.2022.9747735.

[KAH20] 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. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2020)*, 2020, pp. 4587–4591. DOI: 10.1109/ICASSP40776.2020.9053691.

[KAH21] L. Khamidullina, A. L. F. de Almeida, and M. Haardt, "ML-GSVD-based MIMO-NOMA Networks", in *Proc. 25th International ITG Workshop on Smart Antennas (WSA 2021)*, Nov. 2021, pp. 1–6.

[KAH23] L. Khamidullina, A. L. F. de Almeida, and M. Haardt, "Rate Splitting and Precoding Strategies for Multi-User MIMO Broadcast Channels with Common and Private Streams", in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2023)*, 2023, pp. 1–5. DOI: 10.1109/ICASSP49357.2023.10095138.

[KCH19] L. Khamidullina, Y. Cheng, and M. Haardt, “Constrained Tensor Decompositions for Semi-blind MIMO Detection”, in *Proc. 23rd International ITG Workshop on Smart Antennas (WSA 2019)*, 2019, pp. 1–6.

[MGK+21] A. Manina, M. Grasis, L. Khamidullina, A. Korobkov, J. Haueisen, and M. Haardt, “Coupled CP Decomposition of EEG and MEG Magnetometer and Gradiometer Measurements via the Coupled SECSI Framework”, in *Proc. 55th Asilomar Conference on Signals, Systems, and Computers*, 2021, pp. 1661–1667. DOI: 10.1109/IEEECONF53345.2021.9723118.

[PHT+18] I. Podkurkov, L. Hamidullina, E. Traikov, M. Haardt, and A. Nadeev, “Tensor-Based Near-Field Localization in Bistatic MIMO Radar Systems”, in *Proc. 22nd International ITG Workshop on Smart Antennas (WSA 2018)*, 2018, pp. 1–8.

[SKKH22] I. Safiullin, L. Khamidullina, A. A. Korobkov, and M. Haardt, “Enhanced Computation of the Coupled Block-Term Decomposition in Multilinear Rank Terms”, in *Proc. IEEE 12th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, Trondheim, Norway, Jun. 2022, pp. 400–404. DOI: 10.1109/SAM53842.2022.9827801.

All References

[AA16] 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. DOI: 10.1371/journal.pone.0164546.

[ABB03] 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. of the National Academy of Sciences*, vol. 100, no. 6, pp. 3351–3356, Mar. 2003. DOI: 10.1073/pnas.0530258100.

[AD07] Arthur and David, Vassilvitskii, Sergei, “kmeans++: The Advantages of Careful Seeding”, in *Proc. 8th Annual ACM-SIAM Symposium on Discrete Algorithms*, H. Gabow, Ed., Philadelphia, PA: Society for Industrial and Applied Mathematics, 2007.

[AHA19] W. A. Al-Hussaibi and F. H. Ali, “Efficient User Clustering, Receive Antenna Selection, and Power Allocation Algorithms for Massive MIMO-NOMA Systems”, *IEEE Access*, vol. 7, pp. 31 865–31 882, 2019. DOI: 10.1109/ACCESS.2019.2902331.

[AKK+22] B. Ahmad, L. Khamidullina, A. A. Korobkov, A. Manina, J. Haueisen, and M. Haardt, “Joint Model Order Estimation for Multiple Tensors with A Coupled Mode and Applications to the Joint Decomposition of EEG, MEG Magnetometer, and Gradiometer Tensors”, in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2022)*, May 2022, pp. 1186–1190. DOI: 10.1109/ICASSP43922.2022.9747735.

[ATG+18] M. Aldababsa, M. Toka, S. Gökçeli, G. K. Kurt, and O. Kucur, “A tutorial on nonorthogonal multiple access for 5G and beyond”, *Wireless communications and mobile computing*, vol. 2018, 2018.

[BAC+14] H. Becker, L. Albera, P. Comon, *et al.*, “EEG extended source localization: Tensor-based vs. conventional methods”, *NeuroImage*, vol. 96, pp. 143–157, 2014. DOI: https://doi.org/10.1016/j.neuroimage.2014.03.043.

[Bai92] Z. Bai, "The CSD, GSVD, their Applications and Computations", in *IMA preprint series 958*, University of Minnesota, Apr. 1992.

[BAK99] R. Bro, C. A. Andersson, and H. A. L. Kiers, "PARAFAC2—Part II. Modeling chromatographic data with retention time shifts", *Journal of Chemometrics*, vol. 13, no. 3-4, pp. 295–309, 1999. DOI: https://doi.org/10.1002/(SICI)1099-128X(199905/08)13:3/4<295::AID-CEM547>3.0.CO;2-Y.

[BBV19] C. Beltrán, P. Breiding, and N. Vannieuwenhoven, "Pencil-Based Algorithms for Tensor Rank Decomposition are not Stable", *SIAM Journal on Matrix Analysis and Applications*, vol. 40, no. 2, pp. 739–773, 2019. DOI: 10.1137/18M1200531.

[BDLP+16] J. C. Bruder, M. Dümpelmann, D. Lachner-Piza, M. Mader, A. Schulze-Bonhage, and J. VAN, "Physiological Ripples Associated with Sleep Spindles Differ in Waveform Morphology from Epileptic Ripples", *International Journal of Neural Systems*, vol. 27, Nov. 2016. DOI: 10.1142/S0129065717500113.

[BEO16] T. Bao, M. N. El Korso, and H. H. Ouslimani, "Cramér-Rao bound and statistical resolution limit investigation for near-field source localization", *Digital Signal Processing*, vol. 48, pp. 137–147, 2016. DOI: https://doi.org/10.1016/j.dsp.2015.09.019.

[Bis06] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, 2006.

[BK03] R. Bro and H. A. L. Kiers, "A new efficient method for determining the number of components in PARAFAC models", *Journal of Chemometrics*, vol. 17, no. 5, pp. 274–286, 2003. DOI: https://doi.org/10.1002/cem.801.

[BSL+15] A. Benjebbour, K. Saito, A. Li, Y. Kishiyama, and T. Nakamura, "Non-orthogonal multiple access (NOMA): Concept, performance evaluation and experimental trials", Oct. 2015, pp. 1–6. DOI: 10.1109/WINCOM.2015.7381343.

[BV04] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004. DOI: 10.1017/CBO9780511804441.

[BV18a] S. Boyd and L. Vandenberghe, *Introduction to Applied Linear Algebra: Vectors, Matrices, and Least Squares*. Cambridge University Press, 2018. DOI: 10.1017/9781108583664.

[BV18b] P. Breiding and N. Vannieuwenhoven, "The Condition Number of Join Decompositions", *SIAM Journal on Matrix Analysis and Applications*, vol. 39, no. 1, pp. 287–309, 2018. DOI: 10.1137/17M1142880.

[CAVP21] H. Chen, F. Ahmad, S. Vorobyov, and F. Porikli, "Tensor Decompositions in Wireless Communications and MIMO Radar", *IEEE Journal of Selected Topics in Signal Processing*, vol. 15, no. 3, pp. 438–453, 2021. DOI: 10.1109/JSTSP.2021.3061937.

[CC70] J. D. Carroll and J. J. Chang, "Analysis of individual differences in multidimensional scaling via an n-way generalization of "Eckart-Young" decomposition", *Psychometrika*, vol. 35, pp. 283–319, 1970.

[CDDS19] Z. Chen, Z. Ding, X. Dai, and R. Schober, "Asymptotic Performance Analysis of GSVD-NOMA Systems With a Large-Scale Antenna Array", *IEEE Transactions on Wireless Communications*, vol. 18, no. 1, pp. 575–590, Jan. 2019. DOI: 10.1109/TWC.2018.2883102.

[CHGH18] Y. Cheng, M. Haardt, T. Götz, and J. Haueisen, "Using PARAFAC2 for Multi-Way Component Analysis of Somatosensory Evoked Magnetic Fields and Somatosensory Evoked Electrical Potentials", in *Proc. IEEE 10th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, 2018, pp. 385–389. DOI: 10.1109/SAM.2018.8448504.

[CHRDG07] J. P. C. L. da Costa, M. Haardt, F. Roemer, and G. Del Galdo, "Enhanced Model Order Estimation using Higher-Order Arrays", in *Proc. Conference Record of the Forty-First Asilomar Conference on Signals, Systems and Computers*, Nov. 2007, pp. 412–416. DOI: 10.1109/ACSSC.2007.4487242.

[CLA09] P. Comon, X. Luciani, and A. L. F. de Almeida, "Tensor Decompositions, Alternating Least Squares and other Tales", *Journal of Chemometrics*, vol. 23, pp. 393–405, Aug. 2009. DOI: 10.1002/cem.1236.

[CLK+15] F. Cong, Q.-H. Lin, L.-D. Kuang, X.-F. Gong, P. Astikainen, and T. Ristaniemi, "Tensor decomposition of EEG signals: A brief review", *Journal of Neuroscience Methods*, vol. 248, pp. 59–69, 2015. DOI: https://doi.org/10.1016/j.jneumeth.2015.03.018.

[CMB+94] G. Curio, B.-M. Mackert, M. Burghoff, R. Koetitz, K. Abraham-Fuchs, and W. Härer, "Localization of evoked neuromagnetic 600 Hz activity in the cerebral somatosensory system", *Electroencephalography and Clinical Neurophysiology*, vol. 91, no. 6, pp. 483–487, 1994. DOI: https://doi.org/10.1016/0013-4694(94)90169-4.

[CMB+97] G. Curio, B.-M. Mackert, M. Burghoff, J. Neumann, G. Nolte, M. Scherg, and P. Marx, "Somatotopic source arrangement of 600 Hz oscillatory magnetic fields at the human primary somatosensory hand cortex", *Neuroscience Letters*, vol. 234, no. 2, pp. 131–134, 1997. DOI: https://doi.org/10.1016/S0304-3940(97)00690-3.

[CMD+15] A. Cichocki, D. P. Mandic, L. De Lathauwer, G. Zhou, Q. Zhao, C. F. Caiafa, and A. Phan, "Tensor Decompositions for Signal Processing Applications: From two-way to multiway component analysis", *IEEE Signal Processing Magazine*, vol. 32, pp. 145–163, 2015. DOI: https://doi.org/10.1109/MSP.2013.2297439.

[CNH+18] Y. Cheng, K. Naskovska, M. Haardt, T. Götz, and J. Haueisen, "A new coupled PARAFAC2 decomposition for joint processing of somatosensory evoked magnetic fields and somatosensory evoked electrical potentials", in *Proc. 52nd Asilomar Conference on Signals, Systems, and Computers*, 2018, pp. 806–810. DOI: 10.1109/ACSSC.2018.8645393.

[Coh95] L. Cohen, *Time-Frequency Analysis: Theory and Applications*. USA: Prentice-Hall, Inc., 1995.

[CQA17] J. Chen, J. Qiu, and C. Ahn, "Construction worker's awkward posture recognition through supervised motion tensor decomposition", *Automation in Construction*, vol. 77, pp. 67–81, 2017. DOI: 10.1016/j.autcon.2017.01.020.

[CQC+18] Y. Cai, Z. Qin, F. Cui, G. Y. Li, and J. A. McCann, "Modulation and Multiple Access for 5G Networks", *IEEE Communications Surveys Tutorials*, vol. 20, no. 1, pp. 629–646, 2018. DOI: 10.1109/COMST.2017.2766698.

[CRHH21] Y. Cheng, M. Riesmeyer, J. Haueisen, and M. Haardt, "Using the Multi-Linear Rank-$(L_r, L_r, 1)$ Decomposition for the Detection of the 200 Hz Band Activity in Somatosensory Evoked Magnetic Fields and Somatosensory Evoked Electrical Potentials", *IEEE Access*, vol. 9, pp. 106 232–106 244, 2021. DOI: 10.1109/ACCESS.2021.3100759.

[CWL+23] M. Cui, Z. Wu, Y. Lu, X. Wei, and L. Dai, "Near-Field MIMO Communications for 6G: Fundamentals, Challenges, Potentials, and Future Directions", *IEEE Communications Magazine*, vol. 61, no. 1, pp. 40–46, 2023. DOI: 10.1109/MCOM.004.2200136.

[CZYG19] G. Chen, Z.-F. Zhu, J.-X. Yang, and X.-F. Gong, "Multi-target Localization with MIMO Radar via Coupled Canonical Polyadic Decomposition", in *Proc. IEEE International Conference on Signal, Information and Data Processing (ICSIDP)*, 2019. DOI: 10.1109/ICSIDP47821.2019.9172968.

[DAP16] Z. Ding, F. Adachi, and H. V. Poor, "The Application of MIMO to Non-Orthogonal Multiple Access", *IEEE Transactions on Wireless Communications*, vol. 15, no. 1, pp. 537–552, 2016. DOI: 10.1109/TWC.2015.2475746.

[DB08] L. De Lathauwer and A. de Baynast, "Blind Deconvolution of DS-CDMA Signals by Means of Decomposition in Rank-$(1, L, L)$ Terms", *IEEE Transactions on Signal Processing*, vol. 56, no. 4, pp. 1562–1571, 2008. DOI: 10.1109/TSP.2007.910469.

[DCGC16] M. Dai, B. Clerckx, D. Gesbert, and G. Caire, "A Rate Splitting Strategy for Massive MIMO With Imperfect CSIT", *IEEE Transactions on Wireless Communications*, vol. 15, no. 7, pp. 4611–4624, 2016. DOI: 10.1109/TWC.2016.2543212.

[DCVP+05] W. De Clercq, B. Vanrumste, J.-M. Papy, W. Van Paesschen, and S. Van Huffel, "Modeling common dynamics in multichannel signals with applications to artifact and background removal in EEG recordings", *IEEE Transactions on Biomedical Engineering*, vol. 52, no. 12, pp. 2006–2015, 2005. DOI: 10.1109/TBME.2005.857669.

[DD14] I. Domanov and L. De Lathauwer, "Canonical polyadic decomposition of third-order tensors: Reduction to generalized eigenvalue decomposition", *SIAM Journal on Matrix Analysis and Applications*, vol. 35, no. 2, pp. 636–660, 2014. DOI: 10.1137/130916084.

[DD15] O. Debals and L. De Lathauwer, "Stochastic and deterministic tensorization for blind signal separation", in *Proc. International Conference on Latent Variable Analysis and Signal Separation*, E. Vincent, A. Yeredor, Z. Koldovský, and P. Tichavský, Eds., Cham: Springer International Publishing, 2015, pp. 3–13.

[DD17] I. Domanov and L. De Lathauwer, "Canonical polyadic decomposition of third-order tensors: Relaxed uniqueness conditions and algebraic algorithm", *Linear Algebra and its Applications*, vol. 513, pp. 342–375, 2017. DOI: https://doi.org/10.1016/j.laa.2016.10.019.

[DD20] I. Domanov and L. De Lathauwer, "On Uniqueness and Computation of the Decomposition of a Tensor into Multilinear Rank-$(L_r, L_r, 1)$ Terms", *SIAM Journal on Matrix Analysis and Applications*, vol. 41, no. 2, pp. 747–803, 2020. DOI: 10.1137/18M1206849.

[DDP16] Z. Ding, L. Dai, and H. V. Poor, "MIMO-NOMA Design for Small Packet Transmission in the Internet of Things", *IEEE Access*, vol. 4, pp. 1393–1405, 2016. DOI: 10.1109/ACCESS.2016.2551040.

[DDV00a] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A Multilinear Singular Value Decomposition", *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, Mar. 2000. DOI: 10.1137/S0895479896305696.

[DDV00b] L. De Lathauwer, B. De Moor, and J. Vandewalle, "On the Best Rank-1 and Rank-$(R_1, R_2, \ldots, R_N)$ Approximation of Higher-Order Tensors", *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1324–1342, 2000. DOI: 10.1137/S0895479898346995.

[DeL06] L. De Lathauwer, "A Link between the Canonical Decomposition in Multilinear Algebra and Simultaneous Matrix Diagonalization", *SIAM Journal on Matrix Analysis and Applications*, vol. 28, no. 3, pp. 642–666, Jan. 2006. DOI: 10.1137/040608830.

[DeL08a] L. De Lathauwer, "Decompositions of a Higher-Order Tensor in Block Terms—Part II: Definitions and Uniqueness", *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 3, pp. 1033–1066, 2008. DOI: 10.1137/070690729.

[DeL08b] L. De Lathauwer, "Decompositions of a Higher-Order Tensor in Block Terms—Part I: Lemmas for Partitioned Matrices", *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 3, pp. 1022–1032, 2008. DOI: 10.1137/060661685.

[DeL11] L. De Lathauwer, "Blind Separation of Exponential Polynomials and the Decomposition of a Tensor in Rank-$(L_r, L_r, 1)$ Terms", *SIAM Journal on Matrix Analysis and Applications*, vol. 32, no. 4, pp. 1451–1474, 2011. DOI: 10.1137/100805510.

[DI15] F. de Dinechin and M. Istoan, "Hardware implementations of fixed-point Atan2", in *Proc.22nd IEEE Symposium on Computer Arithmetic*, Lyon, France, Jun. 2015.

[DKS22] R. Diab, A. Krishnamoorthy, and R. Schober, "Precoding and Decoding Schemes for Downlink MIMO-RSMA with Simultaneous Diagonalization and User Exclusion", in *Proc. IEEE International Conference on Communications Workshops (ICC Workshops)*, 2022, pp. 586–591. DOI: 10.1109/ICCWorkshops53468.2022.9814475.

[dLA+19] M. da Rosa Zanatta, F. L. Lópes de Mendonça, F. Antreich, D. Valle de Lima, R. Kehrle Miranda, G. Del Galdo, and J. P. C. L. da Costa, "Tensor-based time-delay estimation for second and third generation global positioning system", *Digital Signal Processing*, vol. 92, pp. 1–19, 2019. DOI: https://doi.org/10.1016/j.dsp.2019.04.003.

[DLK+17] Z. Ding, X. Lei, G. K. Karagiannidis, R. Schober, J. Yuan, and V. K. Bhargava, "A Survey on Non-Orthogonal Multiple Access for 5G Networks: Research Challenges and Future Trends", *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 10, pp. 2181–2195, 2017. DOI: 10.1109/JSAC.2017.2725519.

[DMC21] O. Dizdar, Y. Mao, and B. Clerckx, "Rate-Splitting Multiple Access to Mitigate the Curse of Mobility in (Massive) MIMO Networks", *IEEE Transactions on Communications*, vol. 69, no. 10, pp. 6765–6780, 2021. DOI: 10.1109/TCOMM.2021.3098695.

[DN08] L. De Lathauwer and D. Nion, "Decompositions of a Higher-Order Tensor in Block Terms—Part III: Alternating Least Squares Algorithms", *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 3, pp. 1067–1083, 2008. DOI: 10.1137/070690730.

[DNDT17] T. X. Doan, H. Q. Ngo, T. Q. Duong, and K. Tourki, "On the Performance of Multi-group Multicast Cell-Free Massive MIMO", *IEEE Communications Letters*, vol. 21, no. 12, pp. 2642–2645, Dec. 2017. DOI: 10.1109/LCOMM.2017.2745578.

[dVD+07] M. de Vos, A. Vergult, L. De Lathauwer, W. de Clercq, S. van Huffel, P. Dupont, A. Palmini, and W. van Paesschen, "Canonical decomposition of ictal scalp EEG reliably detects the seizure onset zone", *NeuroImage*, vol. 37, no. 3, pp. 844–854, 2007. DOI: 10.1016/j.neuroimage.2007.04.041.

[DVD18] I. Domanov, N. Vervliet, and L. De Lathauwer, *Decomposition of a Tensor into Multilinear Rank-$(M_r, N_r, \cdot)$ Terms*, Version 20121115, 2018.

[DWD+18] L. Dai, B. Wang, Z. Ding, Z. Wang, S. Chen, and L. Hanzo, "A Survey of Non-Orthogonal Multiple Access for 5G", *IEEE Communications Surveys Tutorials*, vol. 20, no. 3, pp. 2294–2323, 2018. DOI: 10.1109/COMST.2018.2835558.

[DYHL15] J. Du, C. Yuan, Z. Hu, and H. Lin, "A Novel Tensor-Based Receiver for Joint Symbol and Channel Estimation in Two-Hop Cooperative MIMO Relay Systems", *IEEE Communications Letters*, vol. 19, no. 11, pp. 1961–1964, 2015. DOI: 10.1109/LCOMM.2015.2473172.

[EP12] M. N. El Korso and M. Pesavento, "Performance analysis for near field source localization", in *Proc. IEEE 7th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, 2012, pp. 197–200. DOI: 10.1109/SAM.2012.6250465.

[EVD22] E. Evert, M. Vandecappelle, and L. De Lathauwer, "A recursive eigenspace computation for the canonical polyadic decomposition", *SIAM Journal on Matrix Analysis and Applications*, vol. 43, no. 1, pp. 274–300, 2022. DOI: 10.1137/21M1423026.

[EY36] C. Eckart and G. M. Young, "The approximation of one matrix by another of lower rank", *Psychometrika*, vol. 1, pp. 211–218, 1936.

[FA14] G. Favier and A. de Almeida, "Overview of constrained PARAFAC models", *EURASIP Journal on Advances in Signal Processing*, vol. 2014, Sep. 2014. DOI: 10.1186/1687-6180-2014-142.

[FBH+22] A. Fawzi, M. Balog, A. Huang, *et al.*, "Discovering faster matrix multiplication algorithms with reinforcement learning", *Nature*, vol. 610, pp. 47–53, Oct. 2022. DOI: 10.1038/s41586-022-05172-4.

[FDLC21] A. R. Flores, R. C. De Lamare, and B. Clerckx, "Tomlinson-Harashima Precoded Rate-Splitting With Stream Combiners for MU-MIMO Systems", *IEEE Transactions on Communications*, vol. 69, no. 6, pp. 3833–3845, 2021. DOI: 10.1109/TCOMM.2021.3065145.

[FG06] T. Fu and X. Gao, "Simultaneous Diagonalization With Similarity Transformation for Non-Defective Matrices", in *Proc. IEEE International Conference on Acoustics Speech and Signal Processing Proceedings (ICASSP 2006)*, vol. 4, May 2006. DOI: 10.1109/ICASSP.2006.1661174.

[FL22] A. R. Flores and R. C. de Lamare, "Robust and Adaptive Power Allocation Techniques for Rate Splitting Based MU-MIMO Systems", *IEEE Transactions on Communications*, vol. 70, no. 7, pp. 4656–4670, 2022. DOI: 10.1109/TCOMM.2022.3179780.

[FLC20] A. R. Flores, R. C. de Lamare, and B. Clerckx, "Linear Precoding and Stream Combining for Rate Splitting in Multiuser MIMO Systems", *IEEE Communications Letters*, vol. 24, no. 4, pp. 890–894, 2020. DOI: 10.1109/LCOMM.2020.2969158.

[FS12] 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. Information Theory*, Jul. 2012, pp. 2321–2325. DOI: 10.1109/ISIT.2012.6283927.

[GAT+16] S. K. Goh, H. A. Abbass, K. C. Tan, A. Al-Mamun, C. Guan, and C. C. Wang, "Multi-way analysis of EEG artifacts based on Block Term Decomposition", in *International Joint Conference on Neural Networks (IJCNN 2016)*, 2016, pp. 913–920. DOI: 10.1109/IJCNN.2016.7727296.

[GBC16] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[GHH14] S. Ge, M. Han, and X. Hong, "A fully automatic ocular artifact removal from EEG based on fourth-order tensor method", *Biomedical Engineering Letters*, vol. 4, pp. 55–63, Mar. 2014. DOI: 10.1007/s13534-014-0118-2.

[GHWH14] T. Götz, R. Huonker, O. W. Witte, and J. Haueisen, "Thalamocortical impulse propagation and information transfer in eeg and meg", *Journal of Clinical Neurophysiology*, vol. 31, pp. 253–260, 2014.

[GLD+16] X.-F. Gong, Q.-H. Lin, O. Debals, N. Vervliet, and L. De Lathauwer, "Coupled rank-$(L_m, L_n, \bullet)$ block term decomposition by coupled block simultaneous generalized Schur decomposition", in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2016)*, Mar. 2016, pp. 2554–2558. DOI: 10.1109/ICASSP.2016.7472138.

[GOF+20] J. H. d. M. Goulart, P. M. R. de Oliveira, R. C. Farias, V. Zarzoso, and P. Comon, "Alternating Group Lasso for Block-Term Tensor Decomposition and Application to ECG Source Separation", *IEEE Transactions on Signal Processing*, vol. 68, pp. 2682–2696, 2020. DOI: 10.1109/TSP.2020.2985591.

[GP20] E. Gujral and E. E. Papalexakis, "OnlineBTD: Streaming Algorithms to Track the Block Term Decomposition of Large Tensors", in *Proc. IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, 2020, pp. 168–177. DOI: 10.1109/DSAA49011.2020.00029.

[GP98] D. C. Ghiglia and M. D. Pritt, *Two-Dimensional Phase Unwrapping: Theory, Algorithms, and Software*. Wiley New York, 1998, vol. 4.

[GPBS22] A. Ghosh, P. Pahari, P. Basak, and A. Sarkar, "Epileptic-Seizure onset detection using PARAFAC model with Cross-Wavelet Transformation on Multi-Channel EEG", *Physical and engineering sciences in medicine*, vol. 45, pp. 602–612, Feb. 2022. DOI: https://doi.org/10.1007/s13246-022-01127-1.

[GPP20] E. Gujral, R. Pasricha, and E. Papalexakis, "Beyond Rank-1: Discovering Rich Community Structure in Multi-Aspect Graphs", in *Proc. The Web Conference 2020*, New York, USA, Apr. 2020, pp. 452–462.

[GPS02] H. Goldstein, C. Poole, and J. Safko, *Classical Mechanics*. Addison Wesley, 2002.

[Gre52] B. Green, "The orthogonal approximation of an oblique structure in factor analysis", *Psychometrika*, vol. 17, no. 4, pp. 429–440, Dec. 1952. DOI: 10.1007/BF02288918.

[GRK22] P. V. Giampouras, A. A. Rontogiannis, and E. Kofidis, "Block-Term Tensor Decomposition Model Selection and Computation: The Bayesian Way", *IEEE Transactions on Signal Processing*, vol. 70, pp. 1704–1717, 2022. DOI: 10.1109/TSP.2022.3159029.

[GVL13] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Fourth. The Johns Hopkins University Press, 2013. DOI: https://doi.org/10.56021/9781421407944.

[GVL96] G. Golub and C. Van Loan, *Matrix Computations* (Johns Hopkins Studies in the Mathematical Sciences). Johns Hopkins University Press, 1996.

[Gvv+17] E. E. Geertsema, M. A. van 't Klooster, N. E. van Klink, F. S. Leijten, P. C. van Rijen, G. H. Visser, S. N. Kalitzin, and M. Zijlmans, "Non-harmonicity in high-frequency components of the intra-operative corticogram to delineate epileptogenic tissue during surgery", *Clinical Neurophysiology*, vol. 128, no. 1, pp. 153–164, 2017. DOI: https://doi.org/10.1016/j.clinph.2016.11.007.

[HAK+17a] X. Han, L. Albera, A. Kachenoura, L. Senhadji, and H. Shu, “Low rank canonical polyadic decomposition of tensors based on group sparsity”, in *Proc. 25th European Signal Processing Conference (EUSIPCO 2017)*, 2017, pp. 668–672. DOI: 10.23919/EUSIPCO.2017.8081291.

[HAK+17b] X. Han, L. Albera, A. Kachenoura, H. Shu, and L. Senhadji, “Block term decomposition with rank estimation using group sparsity”, in *Proc. IEEE 7th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP 2017)*, 2017, pp. 1–5. DOI: 10.1109/CAMSAP.2017.8313206.

[Har70] R. A. Harshman, “Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-model factor analysis”, *UCLA Working Papers in Phonetics*, vol. 16, pp. 1–84, Dec. 1970.

[Har72] R. A. Harshman, “PARAFAC2: Mathematical and technical notes”, *UCLA Working Papers in Phonetics*, vol. 22, pp. 30 –44, 1972.

[HCS+14] B. Hunyadi, D. Camps, L. Sorber, W. van Paesschen, M. de Vos, S. Van Huffel, and L. De Lathauwer, “Block term decomposition for modelling epileptic seizures”, *EURASIP Journal on Advances in Signal Processing*, vol. 2014, no. 1, 2014. DOI: 10.1186/1687-6180-2014-139.

[HCS96] M. Haardt, R. N. Challa, and S. Shamsunder, “Improved bearing and range estimation via high-order subspace based Unitary ESPRIT”, in *Proc. 30th Asilomar Conf. Signals, Systems and Computers*, Pacific Grove, CA, USA, Nov. 1996, pp. 380–384. DOI: 10.1109/ACSSC.1996.600930.

[HD19] M. F. Hanif and Z. Ding, “Robust Power Allocation in MIMO-NOMA Systems”, *IEEE Wireless Communications Letters*, vol. 8, no. 6, pp. 1541–1545, 2019. DOI: 10.1109/LWC.2019.2926277.

[Hjø11] A. Hjørungnes, *Complex-Valued Matrix Derivatives: With Applications in Signal Processing and Communications*. Cambridge University Press, 2011. DOI: 10.1017/CBO9780511921490.

[HRC17] C. Hao, B. Rassouli, and B. Clerckx, “Achievable DoF Regions of MIMO Networks With Imperfect CSIT”, *IEEE Transactions on Information Theory*, vol. 63, no. 10, pp. 6587–6606, 2017. DOI: 10.1109/TIT.2017.2735422.

[HRML21] D. Hassan, S. Redif, J. G. McWhirter, and S. Lambotharan, “Polynomial GSVD Beamforming for Two-User Frequency-Selective MIMO Channels”, *IEEE Transactions on Signal Processing*, vol. 69, pp. 948–959, Jan. 2021. DOI: 10.1109/TSP.2021.3052040.

[HSM+01] J. Haueisen, B. Schack, T. Meier, G. Curio, and Y. Okada, “Multiplicity in the high-frequency signals during the short-latency somatosensory evoked cortical activity in humans”, *Clinical Neurophysiology*, vol. 112, no. 7, pp. 1316–1325, 2001. DOI: https://doi.org/10.1016/S1388-2457(01)00504-1.

[IADK17] S. M. R. Islam, N. Avazov, O. A. Dobre, and K.-s. Kwak, “Power-Domain Non-Orthogonal Multiple Access (NOMA) in 5G Systems: Potentials and Challenges”, *IEEE Communications Surveys Tutorials*, vol. 19, no. 2, pp. 721–742, 2017. DOI: 10.1109/COMST.2016.2621116.

[IKS20] M. S. Ibrahim, A. Konar, and N. D. Sidiropoulos, “Fast Algorithms for Joint Multicast Beamforming and Antenna Selection in Massive MIMO”, *IEEE Transactions on Signal Processing*, vol. 68, pp. 1897–1909, Mar. 2020. DOI: 10.1109/TSP.2020.2979545.

[Ito82] K. Itoh, "Analysis of the phase unwrapping problem", *Applied Optics*, vol. 21, no. 14, p. 2470, Jul. 1982.

[JC16] H. Joudeh and B. Clerckx, "Sum-Rate Maximization for Linearly Precoded Downlink Multiuser MISO Systems With Partial CSIT: A Rate-Splitting Approach", *IEEE Transactions on Communications*, vol. 64, no. 11, pp. 4847–4861, 2016. DOI: 10.1109/TCOMM.2016.2603991.

[JL06] N. Jindal and Z. Luo, "Capacity Limits of Multiple Antenna Multicast", in *Proc. IEEE Int. Symp. Information Theory*, Jul. 2006, pp. 1841–1845. DOI: 10.1109/ISIT.2006.261753.

[JNN+17] V. Jayabal, S. Nagarajan, M. Narayanan, *et al.*, "Magnetoencephalographic imaging of ictal high-frequency oscillations (80-200 Hz) in pharmacologically resistant focal epilepsy", *Epilepsia*, vol. 59, Nov. 2017. DOI: 10.1111/epi.13940.

[JNTS15] J. Joung, H. D. Nguyen, P. H. Tan, and S. Sun, "Multicast Linear Precoding for MIMO-OFDM Systems", *IEEE Communications Letters*, vol. 19, no. 6, pp. 993–996, Jun. 2015.

[KAH20] 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. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2020)*, 2020, pp. 4587–4591. DOI: 10.1109/ICASSP40776.2020.9053691.

[KAH21] L. Khamidullina, A. L. F. de Almeida, and M. Haardt, "ML-GSVD-based MIMO-NOMA Networks", in *Proc. 25th International ITG Workshop on Smart Antennas (WSA 2021)*, Nov. 2021, pp. 1–6.

[KAH22] L. Khamidullina, A. L. F. de Almeida, and M. Haardt, "Multilinear Generalized Singular Value Decomposition (ML-GSVD) and Its Application to Multiuser MIMO Systems", *IEEE Transactions on Signal Processing*, vol. 70, pp. 2783–2797, 2022. DOI: 10.1109/TSP.2022.3178902.

[KAH23] L. Khamidullina, A. L. F. de Almeida, and M. Haardt, "Rate Splitting and Precoding Strategies for Multi-User MIMO Broadcast Channels with Common and Private Streams", in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2023)*, 2023, pp. 1–5. DOI: 10.1109/ICASSP49357.2023.10095138.

[KB09] T. G. Kolda and B. W. Bader, "Tensor Decompositions and Applications", *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009. DOI: https://doi.org/10.1137/07070111X.

[KBB99] H. A. L. Kiers, J. M. F. ten Berge, and R. Bro, "PARAFAC2 -Part I. A direct fitting algorithm for the PARAFAC2 model", *Journal of Chemometrics*, vol. 13, pp. 275–294, May 1999. DOI: https://doi.org/10.1002/(SICI)1099-128X(199905/08)13:3/4%3C275::AID-CEM543%3E3.0.CO;2-B.

[KCH19] L. Khamidullina, Y. Cheng, and M. Haardt, "Constrained Tensor Decompositions for Semi-blind MIMO Detection", in *Proc. 23rd International ITG Workshop on Smart Antennas (WSA 2019)*, 2019, pp. 1–6.

[KDHH21] A. A. Korobkov, M. K. Diugurova, J. Haueisen, and M. Haardt, "Multi-dimensional model order estimation using LineAr Regression of Global Eigenvalues (LaRGE) with applications to EEG and MEG recordings", in *Proc. 28th European Signal Processing Conference (EUSIPCO 2020)*, Jan. 2021, pp. 1005–1009. DOI: 10.23919/Eusipco47968.2020.9287523.

[KDHH22] A. A. Korobkov, M. K. Diugurova, J. Haueisen, and M. Haardt, "Robust Multi-Dimensional Model Order Estimation Using LineAr Regression of Global Eigenvalues (LaRGE)", *IEEE Transactions on Signal Processing*, vol. 70, pp. 5751–5764, 2022. DOI: 10.1109/TSP.2022.3222737.

[KGD21] I. Kempf, P. J. Goulart, and S. R. Duncan, *A Higher-Order Generalized Singular Value Decomposition for Rank Deficient Matrices*, arXiv:2102.09822, Jul. 2021. arXiv: 2102.09822 [math.NA].

[Kie00] H. A. L. Kiers, "Towards a standardized notation and terminology in multiway analysis", *Journal of Chemometrics*, vol. 14, no. 3, pp. 105–122, 2000. DOI: https://doi.org/10.1002/1099-128X(200005/06)14:3<105::AID-CEM582>3.0.CO;2-I.

[Kiz16] R. Kizilirmak, "Non-Orthogonal Multiple Access (NOMA) for 5G Networks", in Dec. 2016. DOI: 10.5772/66048.

[KL80] P. Kroonenberg and J. de Leeuw, "Principal Component Analysis of Three-mode Data by Means of Alternating Least Squares Algorithm", *Psychometrika*, vol. 45, pp. 69–97, Mar. 1980. DOI: 10.1007/BF02293599.

[KLP11] I. H. Kim, D. J. Love, and S. Y. Park, "Optimal and Successive Approaches to Signal Design for Multiple Antenna Physical Layer Multicasting", *IEEE Transactions on Communications*, vol. 59, no. 8, pp. 2316–2327, Aug. 2011. DOI: 10.1109/TCOMM.2011.060911.080464.

[KPH21] L. Khamidullina, I. Podkurkov, and M. Haardt, "Conditional and Unconditional Cramér-Rao Bounds for Near-Field Localization in Bistatic MIMO Radar Systems", *IEEE Transactions on Signal Processing*, vol. 69, pp. 3220–3234, 2021. DOI: 10.1109/TSP.2021.3082469.

[Kru77] J. B. Kruskal, "Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics", *Linear Algebra and its Applications*, vol. 18, no. 2, pp. 95–138, 1977. DOI: https://doi.org/10.1016/0024-3795(77)90069-6.

[KS22] A. Krishnamoorthy and R. Schober, "Downlink MIMO-RSMA with Successive Null-Space Precoding", *IEEE Transactions on Wireless Communications*, pp. 1–1, 2022. DOI: 10.1109/TWC.2022.3173463.

[KSAD15] S. Kouchaki, S. Sanei, E. L. Arbon, and D.-J. Dijk, "Tensor Based Singular Spectrum Analysis for Automatic Scoring of Sleep EEG", *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 23, no. 1, pp. 1–9, 2015. DOI: 10.1109/TNSRE.2014.2329557.

[KSP+23] L. Khamidullina, G. Seidl, I. A. Podkurkov, A. A. Korobkov, and M. Haardt, "Enhanced Solutions for the Block-Term Decomposition in Rank-$(L_r, L_r, 1)$ Terms", *IEEE Transactions on Signal Processing*, vol. 71, pp. 2608–2621, 2023. DOI: 10.1109/TSP.2023.3289730.

[KWWE07] A. Khisti, G. Wornell, A. Wiesel, and Y. Eldar, "On the Gaussian MIMO Wiretap Channel", in *Proc. IEEE International Symposium on Information Theory*, Jun. 2007, pp. 2471–2475. DOI: 10.1109/ISIT.2007.4557590.

[LA11] X. Luciani and L. Albera, "Semi-algebraic canonical decomposition of multiway arrays and Joint Eigenvalue Decomposition", in *Proc. 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2011, pp. 4104–4107. DOI: 10.1109/ICASSP.2011.5947255.

[LASA12] 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. DOI: 10.1371/journal.pone.0030098.

[LC18] N. Lee and A. Cichocki, "Fundamental tensor operations for large-scale data analysis using tensor network formats", *Multidimensional Syst. Signal Process.*, vol. 29, no. 3, pp. 921–960, Jul. 2018. DOI: 10.1007/s11045-017-0481-0.

[LLN+20] Z. Lin, T. Lv, W. Ni, J. A. Zhang, and R. P. Liu, "Tensor-Based Multi-Dimensional Wideband Channel Estimation for mmWave Hybrid Cylindrical Arrays", *IEEE Transactions on Communications*, vol. 68, no. 12, pp. 7608–7622, Dec. 2020. DOI: 10.1109/TCOMM.2020.3023934.

[Llo82] S. Lloyd, "Least squares quantization in PCM", *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982. DOI: 10.1109/TIT.1982.1056489.

[LS01] X. Liu and N. Sidiropoulos, "Cramer-Rao lower bounds for low-rank decomposition of multidimensional arrays", *IEEE Transactions on Signal Processing*, vol. 49, no. 9, pp. 2074–2086, 2001. DOI: 10.1109/78.942635.

[LT08] S. Liu and G. Trenkler, "Hadamard, Khatri-Rao, Kronecker and other matrix products", *International Journal of Information and Systems Sciences*, vol. 4, Jan. 2008.

[LY18] Z. Li and S. Yang, "A Linearly Precoded Rate Splitting Approach and Its Optimality for MIMO Broadcast Channels", in *Proc. IEEE Information Theory Workshop (ITW 2018)*, 2018, pp. 1–5. DOI: 10.1109/ITW.2018.8613400.

[LYS21] Z. Li, S. Yang, and S. Shamai, "On Linearly Precoded Rate Splitting for Gaussian MIMO Broadcast Channels", *IEEE Transactions on Information Theory*, vol. 67, no. 7, pp. 4693–4709, 2021. DOI: 10.1109/TIT.2021.3076888.

[MC20] Y. Mao and B. Clerckx, "Beyond Dirty Paper Coding for Multi-Antenna Broadcast Channel With Partial CSIT: A Rate-Splitting Approach", *IEEE Transactions on Communications*, vol. 68, no. 11, pp. 6775–6791, 2020. DOI: 10.1109/TCOMM.2020.3014153.

[MCBA20] B. Makki, K. Chitti, A. Behravan, and M.-S. Alouini, "A Survey of NOMA: Current Status and Open Research Challenges", *IEEE Open Journal of the Communications Society*, vol. 1, pp. 179–189, 2020. DOI: 10.1109/OJCOMS.2020.2969899.

[MCF16] W. Mei, Z. Chen, and J. Fang, "GSVD-Based Precoding in MIMO Systems With Integrated Services", *IEEE Signal Processing Letters*, vol. 23, no. 11, pp. 1528–1532, Nov. 2016. DOI: 10.1109/LSP.2016.2606349.

[MCL18] Y. Mao, B. Clerckx, and V. Li, "Rate-Splitting Multiple Access for Downlink Communication Systems: Bridging, Generalizing and Outperforming SDMA and NOMA", *EURASIP Journal on Wireless Communications and Networking*, vol. 2018, May 2018. DOI: 10.1186/s13638-018-1104-7.

[MDC+22] Y. Mao, O. Dizdar, B. Clerckx, R. Schober, P. Popovski, and H. V. Poor, "Rate-splitting multiple access: Fundamentals, survey, and future research trends", *IEEE Communications Surveys & Tutorials*, vol. 24, no. 4, pp. 2073–2126, 2022. DOI: 10.1109/COMST.2022.3191937.

[MDFT16] Z. Ma, Z. Ding, P. Fan, and S. Tang, "A General Framework for MIMO Uplink and Downlink Transmissions in 5G Multiple Access", in *IEEE 83rd Vehicular Technology Conference (VTC Spring)*, 2016, pp. 1–4. DOI: 10.1109/VTCSpring.2016.7504070.

[MFG+18] A. H. Mooij, B. Frauscher, S. A. M. Goemans, G. J. M. Huiskamp, K. P. J. Braun, and M. Zijlmans, "Ripples in scalp EEGs of children: co-occurrence with sleep-specific transients and occurrence across sleep stages", *Sleep*, vol. 41, no. 11, zsy169, Aug. 2018. DOI: 10.1093/sleep/zsy169.

[MGK+21] A. Manina, M. Grasis, L. Khamidullina, A. Korobkov, J. Haueisen, and M. Haardt, "Coupled CP Decomposition of EEG and MEG Magnetometer and Gradiometer Measurements via the Coupled SECSI Framework", in *Proc. 55th Asilomar Conference on Signals, Systems, and Computers*, 2021, pp. 1661–1667. DOI: 10.1109/IEEECONF53345.2021.9723118.

[MMDC22] A. Mishra, Y. Mao, O. Dizdar, and B. Clerckx, "Rate-Splitting Multiple Access for Downlink Multiuser MIMO: Precoder Optimization and PHY-Layer Design", *IEEE Transactions on Communications*, vol. 70, no. 2, pp. 874–890, 2022. DOI: 10.1109/TCOMM.2021.3138437.

[MRJ+17] A. H. Mooij, R. C. M. A. Raijmann, F. E. Jansen, K. P. J. Braun, and M. Zijlmans, "Physiological Ripples (±100 Hz) in Spike-Free Scalp EEGs of Children With and Without Epilepsy", *Brain Topography*, vol. 30, pp. 739 –746, 2017.

[Mur12] K. Murphy, *Machine Learning: A Probabilistic Perspective* (Adaptive Computation and Machine Learning series). MIT Press, 2012.

[NAD+94] M. R. Nuwer, M. Aminoff, J. Desmedt, *et al.*, "IFCN recommended standards for short latency somatosensory evoked potentials. Report of an IFCN committee", *Electroencephalography and Clinical Neurophysiology*, vol. 91, no. 1, pp. 6–11, 1994. DOI: https://doi.org/10.1016/0013-4694(94)90012-4.

[NCdH18] K. Naskovska, Y. Cheng, A. L. F. de Almeida, and M. Haardt, "Efficient Computation of the PARAFAC2 Decomposition via Generalized Tensor Contractions", in *Proc. 52nd Asilomar Conference on Signals, Systems, and Computers*, Oct. 2018, pp. 323–327. DOI: 10.1109/ACSSC.2018.8645170.

[ND09] D. Nion and L. De Lathauwer, "A link between the decomposition of a third-order tensor in rank-(L,L,1) terms and Joint Block Diagonalization", in *Proc. 3rd IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP 2009)*, 2009, pp. 89–92. DOI: 10.1109/CAMSAP.2009.5413231.

[NKHH17] K. Naskovska, A. A. Korobkov, M. Haardt, and J. Haueisen, "Analysis of the photic driving effect via joint EEG and MEG data processing based on the coupled CP decomposition", in *Proc. 25th European Signal Processing Conference (EUSIPCO 2017)*, 2017, pp. 1285–1289. DOI: 10.23919/EUSIPCO.2017.8081415.

[NLK+20] K. Naskovska, S. Lau, A. A. Korobkov, J. Haueisen, and M. Haardt, "Coupled CP Decomposition of Simultaneous MEG-EEG Signals for Differentiating Oscillators During Photic Driving", *Frontiers in Neuroscience*, vol. 14, 2020. DOI: 10.3389/fnins.2020.00261.

[NW06] J. Nocedal and S. Wright, *Numerical Optimization* (Springer Series in Operations Research and Financial Engineering). Springer New York, 2006. DOI: https://doi.org/10.1007/978-0-387-40065-5.

[OFMS11] R. Oostenveld, P. Fries, E. Maris, and J.-M. Schoffelen, "FieldTrip: Open Source Software for Advanced Analysis of MEG, EEG, and Invasive Electrophysiological Data", *Computational intelligence and neuroscience*, vol. 2011, p. 156 869, Jan. 2011. DOI: 10.1155/2011/156869.

[OZ19] P. Oliveira and V. Zarzoso, "Block Term Decomposition of ECG Recordings for Atrial Fibrillation Analysis: Temporal and Inter-Patient Variability", *Journal of Communication and Information Systems*, vol. 34, no. 1, 2019. DOI: 10.14209/jcis.2019.12.

[PCP+13] C. Porcaro, G. Coppola, F. Pierelli, S. Seri, G. Di Lorenzo, L. Tomasevic, C. Salustri, and F. Tecchio, "Multiple frequency functional connectivity in the hand somatosensory network: An EEG study", *Clinical Neurophysiology*, vol. 124, no. 6, pp. 1216–1224, 2013. DOI: https://doi.org/10.1016/j.clinph.2012.12.004.

[PHT+18] I. Podkurkov, L. Hamidullina, E. Traikov, M. Haardt, and A. Nadeev, "Tensor-Based Near-Field Localization in Bistatic MIMO Radar Systems", in *Proc. 22nd International ITG Workshop on Smart Antennas (WSA 2018)*, 2018, pp. 1–8.

[Pow70] M. Powell, "A New Algorithm for Unconstrained Optimization", in *Nonlinear Programming*, ser. Publication no. 25 of the Mathematics Research Center, The University of Wisconsin, J. B. Rosen, O. L. Mangasarian, and K. Ritter, Eds., Burlington: Elsevier Science, 1970, pp. 31–65. DOI: 10.1016/B978-0-12-597050-1.50006-3.

[PP12] K. B. Petersen and M. S. Pedersen, *The Matrix Cookbook*, Version 20121115, Nov. 2012.

[PS81] C. Paige and M. Saunders, "Towards a Generalized Singular Value Decomposition", *SIAM Journal on Numerical Analysis*, vol. 18, no. 3, pp. 398–405, Jun. 1981. DOI: 10.1137/0718026.

[PSK+21] I. Podkurkov, G. Seidl, L. Khamidullina, A. Nadeev, and M. Haardt, "Tensor-Based Near-Field Localization Using Massive Antenna Arrays", *IEEE Transactions on Signal Processing*, vol. 69, pp. 5830–5845, 2021. DOI: 10.1109/TSP.2021.3105797.

[PSVLA11] 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. DOI: 10.1371/journal.pone.0028072.

[PZNH19] I. Podkurkov, J. Zhang, A. F. Nadeev, and M. Haardt, "Efficient Multidimensional Wideband Parameter Estimation for OFDM Based Joint Radar and Communication Systems", *IEEE Access*, vol. 7, pp. 112 792–112 808, 2019. DOI: 10.1109/ACCESS.2019.2927326.

[QV20] Y. Qi and M. Vaezi, "Secure Transmission in MIMO-NOMA Networks", *IEEE Communications Letters*, vol. 24, no. 12, pp. 2696–2700, 2020. DOI: 10.1109/LCOMM.2020.3016999.

[RAZ16] L. N. Ribeiro, A. L. F. de Almeida, and V. Zarzoso, "Enhanced block term decomposition for atrial activity extraction in atrial fibrillation ECG", in *Proc. IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM 2016)*, 2016, pp. 1–5. DOI: 10.1109/SAM.2016.7569709.

[RCH08] M. Rajih, P. Comon, and R. Harshman, "Enhanced line search: A novel method to accelerate PARAFAC", *SIAM Journal on Matrix Analysis and Applications*, vol. 30, Sep. 2008. DOI: 10.1137/06065577.

[RDD20] C. Rao, Z. Ding, and X. Dai, "The Distribution Characteristics of Ordered GSVD Singular Values and its Applications in MIMO-NOMA", *IEEE Communications Letters*, vol. 24, no. 12, pp. 2719–2722, Dec. 2020. DOI: 10.1109/LCOMM.2020.3017796.

[RDD21] C. Rao, Z. Ding, and X. Dai, "GSVD-Based MIMO-NOMA Security Transmission", *IEEE Wireless Communications Letters*, pp. 1–1, 2021. DOI: 10.1109/LWC.2021.3071365.

[RGK21] A. A. Rontogiannis, P. V. Giampouras, and E. Kofidis, "Rank-Revealing Block-Term Decomposition for Tensor Completion", in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2021)*, 2021, pp. 2915–2919. DOI: 10.1109/ICASSP39728.2021.9415104.

[RH10] F. Roemer and M. Haardt, "Tensor-Based Channel Estimation and Iterative Refinements for Two-Way Relaying With Multiple Antennas and Spatial Reuse", *IEEE Transactions on Signal Processing*, vol. 58, no. 11, pp. 5720–5735, Nov. 2010. DOI: 10.1109/TSP.2010.2062179.

[RH13] F. Roemer and M. Haardt, "A semi-algebraic framework for approximate CP decompositions via simultaneous matrix diagonalizations (SECSI)", *Elsevier Signal Processing*, vol. 93, no. 9, pp. 2722–2738, Sep. 2013. DOI: 10.1016/j.sigpro.2013.02.016.

[RHMZ15] L. N. Ribeiro, A. R. Hidalgo-Muñoz, and V. Zarzoso, "Atrial signal extraction in atrial fibrillation electrocardiograms using a tensor decomposition approach", in *Proc. 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC 2015)*, 2015, pp. 6987–6990. DOI: 10.1109/EMBC.2015.7320000.

[RHS10] M. Richards, W. Holm, and J. Scheer, *Principles of Modern Radar: Basic Principles* (Electromagnetics and Radar). Institution of Engineering and Technology, 2010.

[RKG21a] A. A. Rontogiannis, E. Kofidis, and P. V. Giampouras, "Block-Term Tensor Decomposition: Model Selection and Computation", in *Proc. 28th European Signal Processing Conference (EUSIPCO 2020)*, 2021, pp. 1976–1980. DOI: 10.23919/Eusipco47968.2020.9287527.

[RKG21b] A. A. Rontogiannis, E. Kofidis, and P. V. Giampouras, "Block-Term Tensor Decomposition: Model Selection and Computation", *IEEE Journal of Selected Topics in Signal Processing*, vol. 15, no. 3, pp. 464–475, 2021. DOI: 10.1109/JSTSP.2021.3051488.

[RSH12] F. Roemer, C. Schroeter, and M. Haardt, "A semi-algebraic framework for approximate CP decompositions via joint matrix diagonalization and generalized unfoldings", in *Proc. Asilomar Conference on Signals, Systems and Computers*, 2012, pp. 2023–2027. DOI: 10.1109/ACSSC.2012.6489396.

[SB00] N. D. Sidiropoulos and R. Bro, "On the uniqueness of multilinear decomposition of N-way arrays", *Journal of Chemometrics*, vol. 14, no. 3, pp. 229–239, 2000. DOI: https://doi.org/10.1002/1099-128X(200005/06)14:3<229::AID-CEM587>3.0.CO;2-N.

[SBG00] N. Sidiropoulos, R. Bro, and G. Giannakis, "Parallel factor analysis in sensor array processing", *IEEE Transactions on Signal Processing*, vol. 48, no. 8, pp. 2377–2388, Aug. 2000. DOI: 10.1109/78.852018.

[SBG17] Y. Shen, B. Baingana, and G. B. Giannakis, "Tensor Decompositions for Identifying Directed Graph Topologies and Tracking Dynamic Networks", *IEEE Transactions on Signal Processing*, vol. 65, no. 14, pp. 3675–3687, 2017. DOI: 10.1109/TSP.2017.2698369.

[SBL+18] M. Sadeghi, E. Björnson, E. G. Larsson, C. Yuen, and T. Marzetta, "Joint Unicast and Multi-Group Multicast Transmission in Massive MIMO Systems", *IEEE Transactions on Wireless Communications*, vol. 17, no. 10, pp. 6375–6388, Oct. 2018. DOI: 10.1109/TWC.2018.2854554.

[Sch66] P. H. Schoenemann, "A generalized solution of the Orthogonal Procrustes Problem", *Psychometrika*, vol. 31, no. 1, Mar. 1966.

[SD15] M. Sørensen and L. De Lathauwer, "Coupled Canonical Polyadic Decompositions and (Coupled) Decompositions in Multilinear Rank-$(L_{r,n}, L_{r,n}, 1)$ Terms—Part I: Uniqueness", *SIAM Journal on Matrix Analysis and Applications*, vol. 36, pp. 496–522, Apr. 2015. DOI: 10.1137/140956853.

[SDD15] M. Sørensen, I. Domanov, and L. De Lathauwer, "Coupled Canonical Polyadic Decompositions and (Coupled) Decompositions in Multilinear Rank-$(L_{r,n}, L_{r,n}, 1)$ Terms—Part II: Algorithms", *SIAM Journal on Matrix Analysis and Applications*, vol. 36, pp. 1015–1045, Jan. 2015. DOI: 10.1137/140956865.

[SDD18] M. Sørensen, I. Domanov, and L. De Lathauwer, "Coupled Canonical Polyadic Decompositions and Multiple Shift Invariance in Array Processing", *IEEE Transactions on Signal Processing*, vol. 66, no. 14, pp. 3665–3680, Jul. 2018.

[SDF+17] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, "Tensor Decomposition for Signal Processing and Machine Learning", *IEEE Transactions on Signal Processing*, vol. 65, no. 13, pp. 3551–3582, Jul. 2017. DOI: 10.1109/TSP.2017.2690524.

[SDL06] N. D. Sidiropoulos, T. N. Davidson, and Z.-Q. Luo, "Transmit beamforming for physical-layer multicasting", *IEEE Transactions on Signal Processing*, vol. 54, no. 6, pp. 2239–2251, Jun. 2006. DOI: 10.1109/TSP.2006.872578.

[SGCH19] Q. Song, H. Ge, J. Caverlee, and X. Hu, "Tensor Completion Algorithms in Big Data Analytics", *ACM Trans. Knowl. Discov. Data*, vol. 13, no. 1, Jan. 2019. DOI: 10.1145/3278607.

[SK09] Y. C. B. Silva and A. Klein, "Linear Transmit Beamforming Techniques for the Multigroup Multicast Scenario", *IEEE Transactions on Vehicular Technology*, vol. 58, no. 8, pp. 4353–4367, Oct. 2009.

[SKKH22] I. Safiullin, L. Khamidullina, A. A. Korobkov, and M. Haardt, "Enhanced Computation of the Coupled Block-Term Decomposition in Multilinear Rank Terms", in *Proc. IEEE 12th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, Trondheim, Norway, Jun. 2022, pp. 400–404. DOI: 10.1109/SAM53842.2022.9827801.

[SL01] A. Shashua and A. Levin, "Linear image coding for regression and classification using the tensor-rank principle", in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, vol. 1, 2001, pp. I–I. DOI: 10.1109/CVPR.2001.990454.

[SN89] P. Stoica and A. Nehorai, "MUSIC, maximum likelihood, and Cramer-Rao bound", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 5, pp. 720–741, May 1989.

[SRH10] 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 (WSA)*, Feb. 2010, pp. 414–420.

[SRH13] 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 Processing*, vol. 93, no. 9, pp. 2462–2473, Sep. 2013.

[SS07] A. Stegeman and N. D. Sidiropoulos, "On Kruskal's uniqueness condition for the Candecomp/Parafac decomposition", *Linear Algebra and its Applications*, vol. 420, no. 2-3, pp. 540–552, 2007. DOI: 10.1016/j.laa.2006.08.010.

[SSAA15] 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. DOI: 10.1371/journal.pone.0121396.

[ST10] D. Senaratne and C. Tellambura, "Generalized Singular Value Decomposition for Coordinated Beamforming in MIMO Systems", in *Proc. IEEE Global Telecommunications Conference GLOBECOM 2010*, Dec. 2010, pp. 1–6. DOI: 10.1109/GLOCOM.2010.5684109.

[ST13] D. Senaratne and C. Tellambura, "GSVD Beamforming for Two-User MIMO Downlink Channel", *IEEE Transactions on Vehicular Technology*, vol. 62, no. 6, pp. 2596–2606, Jul. 2013. DOI: 10.1109/TVT.2013.2241091.

[Str19] G. Strang, *Linear Algebra and Learning from Data*. Wellesley-Cambridge Press, 2019.

[SVBD13] L. Sorber, M. Van Barel, and L. De Lathauwer, "Optimization-Based Algorithms for Tensor Decompositions: Canonical Polyadic Decomposition, Decomposition in Rank-($L_r, L_r, 1$) Terms, and a New Generalization", *SIAM Journal on Optimization*, vol. 23, no. 2, pp. 695–720, Apr. 2013. DOI: 10.1137/120868323.

[SVBD15] L. Sorber, M. Van Barel, and L. De Lathauwer, "Structured Data Fusion", *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 4, pp. 586–600, 2015. DOI: 10.1109/JSTSP.2015.2400415.

[SvD12] L. Sorber, M. van Barel, and L. De Lathauwer, "Unconstrained Optimization of Real Functions in Complex Variables", *SIAM Journal on Optimization*, vol. 22, no. 3, pp. 879–898, 2012. DOI: 10.1137/110832124.

[SWC16] P. R. Singh, Y. Wang, and P. Chargé, "Bistatic MIMO radar for near field source localisation using PARAFAC", *Electronics Letters*, vol. 52, no. 12, pp. 1060–1061, Apr. 2016. DOI: 10.1049/el.2015.4380.

[SWC17a] P. R. Singh, Y. Wang, and P. Chargé, "Near field targets localization using bistatic MIMO system with symmetric arrays", in *Proc. 25th European Signal Processing Conf. (EUSIPCO)*, Kos, Greece, Aug. 2017, pp. 2403–2407. DOI: 10.23919/EUSIPCO.2017.8081641.

[SWC17b] P. R. Singh, Y. Wang, and P. Chargé, "Near field targets localization using bistatic MIMO system with spherical wavefront based model", in *Proc. 25th European Signal Processing Conf. (EUSIPCO)*, Kos, Greece, Aug. 2017, pp. 2408–2412. DOI: 10.23919/EUSIPCO.2017.8081642.

[TB06] G. Tomasi and R. Bro, "A comparison of algorithms for fitting the PARAFAC model", *Computational Statistics and Data Analysis*, vol. 50, no. 7, pp. 1700–1734, 2006. DOI: https://doi.org/10.1016/j.csda.2004.11.013.

[TH09] S. Taulu and R. Hari, "Removal of magnetoencephalographic artifacts with temporal signal-space separation: Demonstration with single-trial auditory-evoked responses", *Human Brain Mapping*, vol. 30, no. 5, pp. 1524–1534, 2009. DOI: https://doi.org/10.1002/hbm.20627.

[TJR19] M. Taherisadr, M. Joneidi, and N. Rahnavard, "EEG Signal Dimensionality Reduction and Classification using Tensor Decomposition and Deep Convolutional Neural Networks", in *Proc. IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP)*, 2019, pp. 1–6. DOI: 10.1109/MLSP.2019.8918754.

[TPC19] P. Tichavský, A.-H. Phan, and A. Cichocki, "Sensitivity in Tensor Decomposition", *IEEE Signal Processing Letters*, vol. 26, no. 11, pp. 1653–1657, Sep. 2019. DOI: 10.1109/LSP.2019.2943060.

[TPC21] P. Tichavský, A.-H. Phan, and A. Cichocki, "Krylov-Levenberg-Marquardt Algorithm for Structured Tucker Tensor Decompositions", *IEEE Journal of Selected Topics in Signal Processing*, vol. 15, no. 3, pp. 550–559, Feb. 2021. DOI: 10.1109/JSTSP.2021.3059521.

[Tuc66] L. R. Tucker, "Some mathematical notes on three-mode factor analysis", *Psychometrika*, vol. 31, pp. 279–311, 1966.

[Van76] C. F. Van Loan, "Generalizing the Singular Value Decomposition", *SIAM Journal on Numerical Analysis*, vol. 13, pp. 76–83, Mar. 1976. DOI: 10.1137/0713009.

[VDS+16] N. Vervliet, O. Debals, L. Sorber, M. van Barel, and L. De Lathauwer, *Tensorlab 3.0*, Mar. 2016.

[VVM12] N. Vannieuwenhoven, R. Vandebril, and K. Meerbergen, "A new truncation strategy for the higher-order singular value decomposition", eng, *SIAM Journal on Scientific Computing*, vol. 34, no. 2, A1027 –A1052, 2012-01-01.

[WWW+17] Y. Wu, J. Wang, J. Wang, R. Schober, and C. Xiao, "Secure Transmission With Large Numbers of Antennas and Finite Alphabet Inputs", *IEEE Transactions on Communications*, vol. 65, no. 8, pp. 3614–3628, Aug. 2017. DOI: 10.1109/TCOMM.2017.2703654.

[XQZT19] F. Xiong, Y. Qian, J. Zhou, and Y. Y. Tang, "Hyperspectral Unmixing via Total Variation Regularized Nonnegative Tensor Factorization", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 4, pp. 2341–2357, 2019. DOI: 10.1109/tgrs.2018.2872888.

[XTW13] Z. Xiang, M. Tao, and X. Wang, "Coordinated multicast beamforming in multicell networks", *IEEE Transactions on Wireless Communications*, vol. 12, no. 1, pp. 12–21, Jan. 2013. DOI: 10.1109/TWC.2012.101112.112295.

[YD18] J. Yu and M. Dong, "Distributed Low-Complexity Multi-Cell Coordinated Multicast Beamforming with Large-Scale Antennas", in *Proc. IEEE 19th Int. Workshop Signal Processing Advances in Wireless Communications (SPAWC)*, Jun. 2018, pp. 1–5. DOI: 10.1109/SPAWC.2018.8446030.

[YD19] Z. Yang and M. Dong, "Low-Complexity Coordinated Relay Beamforming Design for Multi-Cluster Relay Interference Networks", *IEEE Transactions on Wireless Communications*, vol. 18, no. 4, pp. 2215–2228, Apr. 2019. DOI: 10.1109/TWC.2019.2901477.

[Yer02] A. Yeredor, "Non-orthogonal joint diagonalization in the least-squares sense with application in blind source separation", *IEEE Transactions on Signal Processing*, vol. 50, no. 7, pp. 1545–1553, Jul. 2002. DOI: 10.1109/TSP.2002.1011195.

[YGL+19] J.-X. Yang, X.-F. Gong, H. Li, Y.-G. Xu, and Z.-W. Liu, "Using Coupled Multilinear Rank-$(L, L, 1)$ Block Term Decomposition in Multi-Static-Multi-Pulse MIMO Radar to Localize Targets", in *Proc. Advances in Neural Networks – ISNN 2019*, Springer International Publishing, 2019, pp. 565–574.

[Zar17] V. Zarzoso, "Parameter estimation in block term decomposition for noninvasive atrial fibrillation analysis", in *Proc. IEEE 7th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP 2017)*, 2017, pp. 1–5. DOI: 10.1109/CAMSAP.2017.8313173.

[ZFHW19] G. Zhang, X. Fu, K. Huang, and J. Wang, "Hyperspectral Super-Resolution: A Coupled Nonnegative Block-Term Tensor Decomposition Approach", in *Proc. IEEE 8th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP 2019)*, Dec. 2019, pp. 470–474. DOI: 10.1109/CAMSAP45676.2019.9022476.

[ZFW+20] G. Zhang, X. Fu, J. Wang, X.-L. Zhao, and M. Hong, "Spectrum Cartography via Coupled Block-Term Tensor Decomposition", *IEEE Transactions on Signal Processing*, vol. 68, pp. 3660–3675, 2020. DOI: 10.1109/TSP.2020.2993530.

[ZGK+21] Q. Zhang, B. Guo, W. Kong, X. Xi, Y. Zhou, and F. Gao, "Tensor-based dynamic brain functional network for motor imagery classification", *Biomedical Signal Processing and Control*, vol. 69, p. 102 940, 2021. DOI: https://doi.org/10.1016/j.bspc.2021.102940.

[ZJQ15] E. Zhou, H. Jiang, and H. Qi, "4-D parameter estimation in bistatic MIMO radar for near-field target localization", in *Proc. IEEE Int. Wireless Symp.*, Shenzhen, China, Mar. 2015. DOI: 10.1109/IEEE-IWS.2015.7164609.

[ZLL+18] J. Zeng, T. Lv, R. P. Liu, X. Su, M. Peng, C. Wang, and J. Mei, "Investigation on Evolving Single-Carrier NOMA Into Multi-Carrier NOMA in 5G", *IEEE Access*, vol. 6, pp. 48 268–48 288, 2018. DOI: 10.1109/ACCESS.2018.2868093.

[ZPR12] H. Zhu, N. Prasad, and S. Rangarajan, "Precoder Design for Physical Layer Multicasting", *IEEE Transactions on Signal Processing*, vol. 60, no. 11, pp. 5932–5947, Jun. 2012. DOI: 10.1109/TSP.2012.2210710.

[ZWL10] S. Zhu, D. Wang, and T. Li, "Data clustering with size constraints", *Knowledge-Based Systems*, vol. 23, no. 8, pp. 883–889, 2010. DOI: 10.1016/j.knosys.2010.06.003.

[ZXO+19] W. Zuo, J. Xin, H. Ohmori, N. Zheng, and A. Sano, "Subspace-Based Algorithms for Localization and Tracking of Multiple Near-Field Sources", *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, pp. 156–171, 1 Mar. 2019. DOI: 10.1109/JSTSP.2019.2897953.

Colophon

This thesis was typeset with LaTeX using the KOMA-Script and Ricardo Langner's *Clean Thesis* style.