



# Breaking the curse of dimensionality based on tensor train: models and algorithms

Yassine Zniyed

## ► To cite this version:

Yassine Zniyed. Breaking the curse of dimensionality based on tensor train: models and algorithms. Mathematical Physics [math-ph]. Paris-Saclay, 2019. English. NNT: . tel-02918156v1

**HAL Id: tel-02918156**

**<https://theses.hal.science/tel-02918156v1>**

Submitted on 27 Feb 2020 (v1), last revised 20 Aug 2020 (v2)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Breaking the curse of dimensionality based on tensor train: models and algorithms

Thèse de doctorat de l'Université Paris-Saclay  
préparée à l'Université Paris-Sud

École doctorale n°580 Sciences et technologies de l'information et de  
la communication (STIC)  
Spécialité de doctorat: Traitement du signal et de l'image

Thèse présentée et soutenue à Gif-sur-Yvette, le 15/10/2019, par

**Yassine Zniyed**

Composition du Jury:

Inbar Fijalkow Professeur, ENSEA (ETIS)	Présidente
Eric Moreau Professeur, Université de Toulon (LSIS)	Rapporteur
Laurent Albera Maître de conférences, Université de rennes(LTSI)	Rapporteur
Pierre Comon Directeur de recherche, CNRS (GIPSA-LAB)	Examineur
Sebastian Miron Maître de conférences, Université de Lorraine (CRAN)	Examineur
Jean-Philippe Ovarlez Maître de recherche, CentraleSupélec (ONERA)	Examineur
Rémy Boyer Professeur, Université de Lille (CRISTAL)	Directeur de thèse
Gérard Favier Directeur de recherche, CNRS (I3S)	Co-Directeur de thèse
André L.F. de Almeida Professeur adjoint, Université fédérale du Ceará, Brésil	Co-Directeur de thèse



*“God helps those who help themselves.”*

Algernon Sidney



# Résumé

Le traitement des données massives, communément connu sous l'appellation "Big Data", constitue l'un des principaux défis scientifiques de la communauté STIC.

Plusieurs domaines, à savoir économique, industriel ou scientifique, produisent des données hétérogènes acquises selon des protocoles technologiques multi-modales. Traiter indépendamment chaque ensemble de données mesurées est clairement une approche réductrice et insatisfaisante. En faisant cela, des "relations cachées" ou des inter-corrélations entre les données peuvent être totalement ignorées.

Les représentations tensorielles ont reçu une attention particulière dans ce sens en raison de leur capacité à extraire de données hétérogènes et volumineuses une information physiquement interprétable confinée à un sous-espace de dimension réduite. Dans ce cas, les données peuvent être organisées selon un tableau à  $D$  dimensions, aussi appelé tenseur d'ordre  $D$ .

Dans ce contexte, le but de ce travail est que certaines propriétés soient présentes: (i) avoir des algorithmes de factorisation stables (ne souffrant pas de problème de convergence), (ii) avoir un faible coût de stockage (c'est-à-dire que le nombre de paramètres libres doit être linéaire en  $D$ ), et (iii) avoir un formalisme sous forme de graphe permettant une visualisation mentale simple mais rigoureuse des décompositions tensorielles de tenseurs d'ordre élevé, soit pour  $D > 3$ .

Par conséquent, nous nous appuyons sur la décomposition en train de tenseurs (TT) pour élaborer de nouveaux algorithmes de factorisation TT, et des nouvelles équivalences en termes de modélisation tensorielle, permettant une nouvelle stratégie de réduction de dimensionnalité et d'optimisation de critère des moindres carrés couplés pour l'estimation des paramètres d'intérêts nommé JIRAFE. Aussi, un nouveau cadre hiérarchique pour les décompositions TT des tenseurs Big data est proposé.

Ces travaux d'ordre méthodologique ont eu des applications dans le contexte de l'analyse spectrale multidimensionnelle et des systèmes de télécommunications à relais.



# Abstract

Massive and heterogeneous data processing and analysis have been clearly identified by the scientific community as key problems in several application areas. It was popularized under the generic terms of “data science” or “big data”. Processing large volumes of data, extracting their hidden patterns, while performing prediction and inference tasks has become crucial in economy, industry and science.

Treating independently each set of measured data is clearly a reductive approach. By doing that, “hidden relationships” or inter-correlations between the datasets may be totally missed. Tensor decompositions have received a particular attention recently due to their capability to handle a variety of mining tasks applied to massive datasets, being a pertinent framework taking into account the heterogeneity and multi-modality of the data. In this case, data can be arranged as a  $D$ -dimensional array, also referred to as a  $D$ -order tensor.

In this context, the purpose of my thesis is that the following properties are present: (i) having a stable factorization algorithms (not suffering from convergence problems), (ii) having a low storage cost (i.e., the number of free parameters must be linear in  $D$ ), and (iii) having a formalism in the form of a graph allowing a simple but rigorous mental visualization of tensor decompositions of tensors of high order, i.e., for  $D > 3$ .

Therefore, we rely on the tensor train decomposition (TT) to develop new TT factorization algorithms, and new equivalences in terms of tensor modeling, allowing a new strategy of dimensionality reduction and criterion optimization of coupled least squares for the estimation of parameters named JIRAFE for Joint dImensionality Reduction And Factor rETrieval. Moreover, a new hierarchical framework for the TT decomposition of big data tensors is proposed.

This methodological work has had applications in the context of multidimensional spectral analysis and relay telecommunications systems.





# Contents

<b>Résumé</b>	<b>v</b>
<b>Abstract</b>	<b>vii</b>
<b>Résumé étendu</b>	<b>xxi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Context of study and motivation . . . . .	1
1.2 Achieved results . . . . .	2
1.3 Manuscript structure . . . . .	3
1.4 Publications . . . . .	4
<b>2 State-of-the-art and Problem setup</b>	<b>7</b>
2.1 Tensor-related algebraic background . . . . .	7
2.2 Standard tensor decompositions and low rank tensor approximation . . . . .	9
2.2.1 CPD . . . . .	9
2.2.2 Tucker decomposition . . . . .	10
2.2.3 HOSVD . . . . .	10
2.2.4 Low rank tensor approximation . . . . .	11
2.3 Problem setup . . . . .	12
2.3.1 The curse of dimensionality and tensor networks . . . . .	12
2.3.2 Graph-based illustrations of TNs . . . . .	13
2.3.3 TT decomposition and tensor reshaping . . . . .	14
2.3.4 Why to use the TT decomposition ? . . . . .	15
2.3.5 Description of the TT-SVD algorithm . . . . .	16
2.4 Conclusion . . . . .	16
<b>3 Joint dIMensionality Reduction And Factor rETrieval (JIRAFE)</b>	<b>19</b>
3.1 JIRAFE principle . . . . .	19
3.1.1 Introduction . . . . .	19
3.1.2 TD-Train model: Equivalence between a high-order TD and a train of low-order TDs . . . . .	20
3.1.3 CPD-Train model: Equivalence between a high-order CPD and a train of low-order CPD(s) . . . . .	22
3.1.3.1 Symmetric CPD . . . . .	24
3.1.3.2 Permutation and scaling ambiguities . . . . .	25
3.1.4 Estimation algorithms for the CPD-and TD-Trains . . . . .	26
3.1.4.1 Fast Multilinear Projection (FMP) based on TD-Train . . . . .	26
3.1.4.1.1 Algorithmic description . . . . .	26
3.1.4.1.2 On the difference with the HOSVD . . . . .	27
3.1.4.1.3 Analysis of the computational cost and execution time . . . . .	27
3.1.4.2 Fast CPD with CPD-Train . . . . .	29

3.1.4.2.1	CPD-Train based on low-order ALS algorithm	30
3.1.4.2.2	Improved CPD-Train	32
3.1.4.2.3	Non-iterative CPD-Train algorithm in the case of a known factor	33
3.1.4.2.4	Parallel and non-iterative CPD-Train algorithm in case of Toeplitz factors	34
3.1.5	Discussion	35
3.2	Uniqueness of JIRAFE with Linear Dependencies	36
3.2.1	Context of study	36
3.2.2	Equivalence between the PARALIND and the TTD	36
3.2.2.1	Lack of uniqueness of the TTD	36
3.2.2.2	PARALIND-TTD equivalence	37
3.2.3	Uniqueness of the PARALIND-TTD	38
3.2.4	Discussion	39
3.2.4.1	More restrictive conditions	39
3.2.4.2	Estimation scheme architecture	40
3.2.5	Conclusion and perspectives	40
3.3	JIRAFE-based Multidimensional Harmonic Retrieval	41
3.3.1	Context and data model	41
3.3.2	Dimensionality reduction based on a train of low-order tensors	43
3.3.2.1	Vandermonde based TT decomposition	43
3.3.3	Factor retrieval scheme	47
3.3.3.1	JIRAFE with Vandermonde-based rectification	47
3.3.3.2	Shift Invariance Principle (SIP)	49
3.3.3.2.1	The SIP criterion	49
3.3.3.2.2	Comparison with other RecALS scheme	50
3.3.4	Simulation results	51
3.3.4.1	Advantages of the TT approach	51
3.3.4.1.1	VTT over CPD	51
3.3.4.1.2	Impact of parameters $P$ and $N$ on the VTT-RecALS-SIP	52
3.3.4.2	VTT-RecALS-SIP versus the state-of-art estimators	52
3.3.4.2.1	Robustness to noise	53
3.3.4.2.2	Computational times	54
3.3.5	Discussion	55
3.4	JIRAFE-based MIMO channel decomposition	58
3.4.1	Context of study	58
3.4.2	Tensor-based channel modeling	59
3.4.2.1	Canonical Polyadic Decomposition (CPD) of the channel tensor	59
3.4.2.1.1	Expression of the channel tensor	59
3.4.2.1.2	Model assumptions	60
3.4.2.2	Tensor train decomposition (TTD) of the channel tensor	60
3.4.3	Joint Dimensionality Reduction and Factor Retrieval (JIRAFE)	60
3.4.3.1	JIRAFE: dimensionality reduction	60
3.4.3.2	JIRAFE: CPD factors retrieval	61
3.4.3.2.1	Few dominant paths scenario ( $K < 4$ )	61
3.4.3.2.2	More general multi-path scenario ( $K \geq 4$ )	62
3.4.4	Simulation Results	62
3.4.5	Discussion	65
3.5	Conclusion	66

<b>4</b>	<b>A TT-based hierarchical framework</b>	<b>67</b>
4.1	TT-HSVD algorithm	67
4.1.1	Motivations for a new hierarchical framework	67
4.1.2	Hierarchical methodology to compute the TT decomposition	68
4.1.2.1	Description of the TT-HSVD algorithm	68
4.1.2.2	Graph-based representation with patterns	69
4.1.2.2.1	Splitting/Splitting pattern	69
4.1.2.2.2	Mixed patterns: data processing and TT-core generation	70
4.1.2.2.3	TT-core generation pattern	70
4.1.2.3	Application of the pattern formalism	71
4.1.3	Algebraic analysis of the TT-SVD and TT-HSVD algorithms	71
4.1.3.1	Structure of the estimated TT-cores for the TT-SVD algorithm	72
4.1.3.2	Presentation and analysis of the TT-HSVD algorithm	74
4.1.3.2.1	Structure of the estimated TT-cores	74
4.1.3.3	Comparison of the two schemes	75
4.1.4	Computational complexity and simulation results	76
4.1.4.1	Numerical computation of the SVD	76
4.1.4.2	Complexity analysis and simulations	77
4.1.5	Discussion	80
4.2	TT-based applications	82
4.2.1	Introduction	82
4.2.2	System Model	82
4.2.3	Tensor Train decomposition (TTD)	84
4.2.3.1	Definition of the TTD	84
4.2.3.2	The TTD multiplicative ambiguities	84
4.2.4	TT-based receiver	84
4.2.4.1	TT-cores structure	84
4.2.4.2	Estimation algorithm: TT-MRS	85
4.2.5	Simulation Results	87
4.2.6	Discussion	89
<b>5</b>	<b>Conclusion and prospects</b>	<b>91</b>
<b>A</b>	<b>Proof of Theorem 2</b>	<b>93</b>
<b>B</b>	<b>Proof of Theorem 4</b>	<b>97</b>
<b>C</b>	<b>Proof of Theorem 5</b>	<b>101</b>
<b>D</b>	<b>Algebraic analysis of the patterns</b>	<b>103</b>
	<b>Bibliography</b>	<b>107</b>



# List of Figures

1	CPD d'un tenseur ordre 3 et de rang $R$ .	xxii
2	La décomposition TT d'un tenseur d'ordre $D$ .	xxiii
3	L'algorithme TT-SVD appliqué à un tenseur d'ordre 4.	xxiv
4	L'algorithme TT-HSVD appliqué à un tenseur d'ordre 4.	xxv
5	Combinaison des TT-coeurs estimés d'une manière séquentielle.	xxvi
6	Combinaison des TT-coeurs estimés d'une manière hiérarchique.	xxvi
7	CPD d'ordre 3 du <i>dème</i> coeur TT.	xxviii
2.1	(a) Vector $\mathbf{x}$ of size $I_1 \times 1$ , (b) matrix $\mathbf{X}$ of size $I_1 \times I_2$ , (c) 3-order tensor $\mathcal{X}$ of size $I_1 \times I_2 \times I_3$ , (d) $\times_3^1$ product of two 3-order tensors.	9
2.2	(a) Graph-based Tucker decomposition of a 6-order tensor, (b) TT-based ( <i>left</i> ) and TN-based ( <i>right</i> ) decompositions, (c) HT decomposition.	14
2.3	TT decomposition of a $D$ -order tensor.	15
2.4	TT-SVD applied to a 4-order tensor.	17
3.1	CPD of the $q$ -th 3-order core of the associated Q-order CPD-Train.	24
3.2	Number of flops vs the multilinear rank 4-order tensor with $N = 500$ .	28
3.3	NMSE vs SNR in dB with Fast Multilinear Projection for a 6-order Tucker with $N = 4$ , $T = 2$ , 1000 runs	29
3.4	NMSE vs SNR in dB with JIRAFE for an 8-order CPD	32
3.5	NMSE vs SNR in dB with improved CPD-Train for an 8-order CPD	33
3.6	NMSE vs SNR in dB with CPD-Train-TOMFAC for an 8-order CPD with Toeplitz factors, $N = 6$ , and $R = 3$	35
3.7	A possible TTD of tensor $\mathcal{A}$ .	43
3.8	Representation of the 3-order $M \times N_p \times M$ tensor $\mathcal{V}_p$ .	44
3.9	VTTD of tensor $\mathcal{X}$ corresponding to eq. (3.25).	44
3.10	3-order CPD of the $p$ -th TT-core. Matrices $\mathbf{M}_{p-1}$ and $\mathbf{M}_p$ are latent quantities.	47
3.11	VTT-RecALS representation	49
3.12	MSE vs SNR in dB for $P = 6$ with $M = 3$ , $N = 6$ .	52
3.13	MSE vs tensor dimension for $P = 6$ with $M = 2$ , SNR = 5dB (impact of $N$ ).	53
3.14	MSE vs tensor order for $N = 6$ with $M = 2$ , SNR = 5dB (impact of $P$ ).	53
3.15	MSE vs SNR in dB for $P = 4$ with $M = 2$ , $N = 6$ .	54
3.16	MSE vs SNR in dB for $P = 4$ with $M = 3$ , $N = 6$ .	55
3.17	MSE vs SNR in dB for $P = 6$ with $M = 2$ , $N = 6$ .	55
3.18	MSE vs SNR in dB for $P = 6$ with $M = 2$ , $N = 8$ .	56
3.19	CPU time versus the number of measurements for $P = 6$ , $M = 2$ , $N_i = (4, 6)$ , and SNR= 5dB.	56
3.20	CPU time versus the number of measurements for $P = 6$ , $M = 2$ , $N_i = (6, 8)$ , and SNR= 5dB.	57
3.21	MSE vs SNR in dB with Alg. 9 for $K = 3$ .	63
3.22	MSE vs SNR in dB with Alg. 2 for $K = 4$ .	64

3.23	MSE vs SNR in dB with Alg. 2 for $K = 5$ .	64
3.24	Mean number of iteration for $K = 2$ .	65
4.1	TT-HSVD applied to a 4-order tensor.	69
4.2	Balanced (left) and Unbalanced (right) TT-HSVD applied to a 8-order tensor.	70
4.3	Splitting/Splitting pattern.	71
4.4	Splitting/Generation pattern (left), Generation/Splitting pattern (right).	72
4.5	Generation/Generation pattern.	73
4.6	Graph-based representation using patterns for a balanced TT-HSVD applied to an 8-order tensor.	73
4.7	Graph-based representation using patterns for an unbalanced TT-HSVD applied to an 8-order tensor.	74
4.8	Graph-based illustration of the TT-SVD algorithm.	75
4.9	Graph-based illustration of the TT-HSVD algorithm.	75
4.10	The estimated ranks for the true TT-rank $R_4$ .	80
4.11	One-way two-hop MIMO-OFDM relay system illustration.	83
4.12	NMSE vs SNR in dB with the TTD.	88
4.13	NMSE of $\mathcal{H}^{(RD)}$ for the proposed algorithm.	88
4.14	NMSE of symbols $\mathbf{S}$ for the proposed algorithm.	89

# List of Tables

1	Comparaison du temps de calcul des deux algorithmes ( $D = 8, R = 3$ ).	xxvii
2	Comparaison du temps de calcul pour $R = 3, D = 8$ .	xxix
3.1	Computation times for ( $T = 2, N = 10$ )	28
3.2	Computation times for ( $R = 3, N = 6$ )	31
3.3	Computation times for ( $R = 3, Q = 8$ )	31
3.4	Computation times for ( $R = 3, N = 4$ )	34
3.5	Summary of chosen parameters in Section 3.3.4.2	54
3.6	Comparison of the computation time with SNR= 15dB.	65
4.1	Comparison of the computation time of both algorithms ( $D = 8, R = 3$ ).	79
4.2	Comparison of the computation time of TT-HSVD and TT-SVD.	79
4.3	Comparison of the robustness of TT-HSVD and TT-SVD.	80
4.4	Description of tensors used in eq. (4.8).	83





# List of Symbols

$x$	scalar quantity
$\mathbf{x}$	column vector
$\mathbf{X}$	matrix
$\mathcal{X}$	tensor
$\mathbf{X}(:, i)$	the $i$ th column of the matrix $\mathbf{X}$
$\mathbf{X}(i, :)$	the $i$ th row of the matrix $\mathbf{X}$
$\mathcal{X}(i, :, :)$	the $i$ -th horizontal slice
$\mathcal{X}(:, j, :)$	the $j$ -th lateral slice
$\mathcal{X}(:, :, k)$	the $k$ -th frontal slice
$\text{unfold}_q \mathcal{X}$	the unfolding of tensor $\mathcal{X}$ over its $q$ -th mode
$\mathbf{I}_R$	identity matrix of size $R \times R$
$\mathcal{I}_{k,R}$	the $k$ -order identity tensor of size $R \times \cdots \times R$
$\mathbf{e}_n^{(N)}$	the $n$ th basis vector of $\mathbb{R}^N$
$\otimes$	the Kronecker product
$\odot$	the Khatri-Rao product
$\times_n$	$n$ -mode product
$\times_n^m$	tensor-tensor product
$(\cdot)^{-1}$	the inverse operator
$(\cdot)^T$	the transpose operator
$(\cdot)^\dagger$	the pseudo-inverse operator
$\text{rank}(\cdot)$	rank operator
$\langle \cdot \rangle$	spanned subspace
$\triangleq$	an equality by definition
$O(\cdot)$	a dominant term in a complexity expression
$\text{diag}(\cdot)$	a diagonal matrix from its vector argument
$\text{vec}(\cdot)$	a vector by stacking the columns of its matrix argument
$\text{unvec}_{I \times J}(\cdot)$	the inverse operator of $\text{vec}(\cdot)$
$\ \cdot\ _F$	Frobenius norm
$\text{CritStop}$	a stop criterion for the iterative algorithms



*To my loved ones*



# Résumé étendu

## Introduction

Le traitement des données massives, communément connu sous l'appellation "Big Data", constitue l'un des principaux défis scientifiques de la communauté STIC. Plusieurs domaines, à savoir économique, industriel ou scientifique, produisent des données hétégorènes. Traiter indépendamment chaque ensemble de données mesurées est clairement une approche réductrice. En faisant cela, des "relations cachées" ou des inter-corrélations entre les données peuvent être totalement manquées. Les représentations tensorielles ont reçu une attention particulière dans ce sens en raison de leur capacité de représenter à la fois des données hétégorènes et volumineuses. Dans ce cas, les données peuvent être organisées dans un tableau de  $D$  dimensions, aussi appelé tenseur d'ordre  $D$ . Dans ce contexte, certaines propriétés sont souhaitées: (i) des algorithmes de décomposition stables et évolutifs (c'est-à-dire qui s'adapte avec l'ordre et dimensions du tenseur), (ii) un faible coût de stockage (c'est-à-dire que le nombre de paramètres doit être linéaire en  $D$ ), et (iii) un formalisme graphique adéquat permettant une visualisation simple mais rigoureuse pour les décompositions tensorielles où  $D > 3$ . Notez que pour un tenseur d'ordre  $D$  et de dimension  $N \times \dots \times N$ , le coût de stockage  $O(N^D)$  croît de manière exponentielle avec l'ordre  $D$ . Ce problème est appelé "la malédiction de la dimensionalité" [106]. Dans la littérature, deux décompositions sont populaires, la décomposition canonique polyadique [63, 61, 25, 44, 137] (CPD) et la décomposition de Tucker [142]. Ces deux décompositions ont plusieurs applications dans les télécommunications [128, 3], le traitement d'antennes [127], le radar [110], la compression des données [148], etc. Dans la section suivante nous parlerons des avantages et des inconvénients de ces méthodes, surtout pour des ordres/dimensions très élevés. Après nous exposerons quelques alternatives pour certains de ces problèmes.

## Position du problème

### La CPD

La CPD est la décomposition d'un tenseur  $\mathcal{X} \in \mathcal{C}^{N_1 \times \dots \times N_D}$  d'ordre  $D$  et de rang canonique  $R$  comme sommation de  $R$  tenseurs de rang canonique égale à 1.

$$\mathcal{X} = \sum_{r=1}^R s_r \underbrace{(\mathbf{P}_1(:, r) \circ \dots \circ \mathbf{P}_D(:, r))}_{\mathcal{X}_r} \text{ avec } \text{rank} \mathcal{X}_r = 1$$

$\mathbf{P}_d$  de dimension  $N_d \times R$  est la matrice facteur du mode  $d$ ,  $R$  correspond au nombre minimum de tenseurs de rang 1 requis pour un recouvrement parfait de  $\mathcal{X}$ . Cette décomposition est très compacte, et a un coût de stockage  $O(R \cdot N \cdot D)$ , où  $N = \max\{N_1, \dots, N_D\}$ . Sur la figure 1 on représente la CPD d'un tenseur d'ordre 3 et de rang  $R$ .

Pour décomposer un tenseur  $\mathcal{X}$  d'ordre  $D$  sous la forme CPD, on doit retrouver les

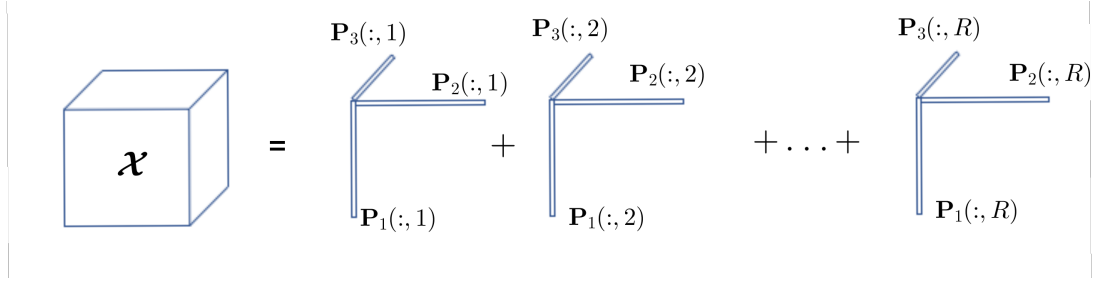


FIGURE 1: CPD d'un tenseur ordre 3 et de rang  $R$ .

facteurs  $\mathbf{P}_d$  qui minimisent

$$\min_{\mathbf{P}_1, \dots, \mathbf{P}_D} \left\| \mathbf{X}_{N_d \times \frac{N_1 \dots N_D}{N_d}} - \mathbf{P}_d \cdot \left( \mathbf{P}_D \odot \dots \odot \mathbf{P}_{d+1} \odot \mathbf{P}_{d-1} \odot \dots \odot \mathbf{P}_1 \right)^T \right\|_F^2 \quad (1)$$

C'est un problème d'optimisation non linéaire. Bro et *al.* [22] ont proposé l'algorithme ALS pour *alternating least squares*. Ils ont proposé de remplacer le problème de eq. (1) par  $D$  problèmes d'optimisation:

$$\mathbf{P}_d^{(t)} = \min_{\mathbf{P}_d} \left\| \mathbf{X}_{N_d \times \frac{N_1 \dots N_D}{N_d}} - \mathbf{P}_d \left( \mathbf{P}_D^{(t-1)} \odot \dots \odot \mathbf{P}_{d+1}^{(t-1)} \odot \mathbf{P}_{d-1}^{(t)} \odot \dots \odot \mathbf{P}_1^{(t)} \right)^T \right\|_F^2$$

Ce problème est facile à implémenter, mais reste un algorithme itératif qui a une convergence lente [118, 74], qui devient de plus en plus difficile quand l'ordre  $D$  augmente [21], avec une complexité computationnelle qui est exponentielle en  $D$ . En outre, sans aucune contrainte, le problème de récupération du rang  $R$  est NP-difficile [62], et du recouvrement des facteurs est mal posé [41, 77]. Finalement, on doit avoir un formalisme graphique simple qui va nous permettre de visualiser des tenseurs d'ordre  $D > 3$ .

## La décomposition de Tucker

La deuxième approche est la décomposition de Tucker. Ça permet de décomposer un tenseur  $\mathcal{X} \in \mathcal{R}^{N_1 \times \dots \times N_D}$  d'ordre  $D$  et de dimension  $N_1 \times \dots \times N_D$  sous forme d'un autre tenseur compressé  $\mathcal{G}$  de même ordre, appelé tenseur coeur, de dimension  $R_1 \times \dots \times R_D$ , multiplié par une matrice facteur  $\mathbf{F}_d$  sur chaque mode  $d$ . Nous avons donc

$$\mathcal{X} = \sum_{r_1, \dots, r_D=1}^{R_1, \dots, R_D} \mathcal{G}(r_1, \dots, r_D) \left( \mathbf{F}_1(:, r_1) \circ \mathbf{F}_2(:, r_2) \circ \dots \circ \mathbf{F}_D(:, r_D) \right)$$

Les facteurs  $\mathbf{F}_d$  ont une dimension  $N_d \times R_d$ , d'où la notion du rang multilinéaire  $(R_1, \dots, R_D)$ . La coût de stockage dans ce cas est  $O(R \cdot N \cdot D + R^D)$ , où  $R = \max\{R_1, \dots, R_D\}$ .

De Lathauwer [89] a proposé d'ajouter une contrainte d'orthonormalité sur les facteurs  $\mathbf{F}_d$ , c'est la décomposition HOSVD pour *Higher-order SVD*, nous avons donc

$$\mathcal{G} = \mathcal{X} \times_1 \mathbf{F}_1^T \times_2 \mathbf{F}_2^T \times_3 \dots \times_D \mathbf{F}_D^T$$

L'avantage de cette décomposition est qu'elle est basée sur un algorithme HOSVD [89] stable (c'est-à-dire non-itératif). Mais, malheureusement  $\mathcal{G}$  n'est généralement pas parcimonieux, donc nous avons un coût de stockage qui est exponentiel en  $D$ . Cette décomposition, à l'inverse de la SVD, n'est pas révélatrice du rang canonique, et en plus n'est pas optimale pour un rang multilinéaire donné.

Pour résumer: (i) la CPD est une décomposition compacte, mais qui n'a pas un algorithme stable de décomposition, et (ii) la décomposition de Tucker/HOSVD est stable pour la décomposition, mais a un coût de stockage exponentiel en  $D$ . L'idéal est d'avoir une décomposition stable avec un coût de stockage linéaire en  $D$ . Dans la section suivante, on va présenter une décomposition qui a les deux propriétés souhaitées.

## Solution alternative: La décomposition Tensor Train

Dans cette section, nous exposerons les réseaux de tenseurs, en particulier la décomposition Tensor Train (TTD) [104]. L'idée de cette décomposition est de transformer le tenseur d'origine  $\mathcal{X}$  d'ordre  $D > 3$  en un ensemble de tenseurs  $\mathcal{G}_d$  d'ordre 3. La décomposition d'un tenseur  $\mathcal{X} \in \mathcal{C}^{N_1 \times \dots \times N_D}$  d'ordre  $D$  avec des rangs TT  $(R_1, \dots, R_{D-1})$  est exprimée comme

$$\mathcal{X}(i_1, i_2, \dots, i_D) = \sum_{r_1, \dots, r_{D-1}=1}^{R_1, \dots, R_{D-1}} \mathbf{G}_1(i_1, r_1) \mathcal{G}_2(r_1, i_2, r_2) \mathcal{G}_3(r_2, i_3, r_3) \cdots \cdots \mathcal{G}_{D-1}(r_{D-2}, i_{D-1}, r_{D-1}) \mathbf{G}_D(r_{D-1}, i_D)$$

Ou d'une manière plus compacte

$$\mathcal{X} = \mathbf{G}_1 \times_2^1 \mathcal{G}_2 \times_3^1 \mathcal{G}_3 \times_4^1 \cdots \times_{D-1}^1 \mathcal{G}_{D-1} \times_D^1 \mathbf{G}_D. \quad (2)$$

Graphiquement, et pour avoir une vision conceptuelle de la TTD, nous représentons dans la figure 2, la TTD d'un tenseur d'ordre  $D$ .

Notez que le coût de stockage de la TTD est  $O((D-2) \cdot N \cdot R^2)$ , où  $R = \max\{R_1, \dots, R_{D-1}\}$ .

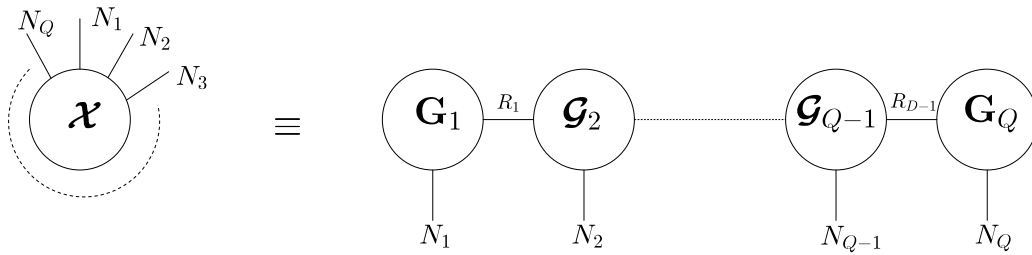


FIGURE 2: La décomposition TT d'un tenseur d'ordre  $D$ .

En plus, cette décomposition se base sur un algorithme stable appelé TT-SVD [104]. Cet algorithme calcule les TT-coeurs  $\mathcal{G}_d$  en extrayant de manière séquentielle les sous-espaces dominants (grâce à des SVD tronquées) des dépliements matriciels du tenseur d'origine. Dans la figure 3, nous illustrons l'algorithme TT-SVD pour un tenseur d'ordre 4. À chaque étape, le tenseur original est redimensionné en une matrice ayant un mode dans une dimension et une combinaison des modes restants dans l'autre dimension, de sorte à générer un TT-coeur après l'application d'une SVD. Notez que les matrices  $\mathbf{U}_d$  contiennent les vecteurs singuliers gauches,



et les matrices  $\mathbf{V}_d$  contiennent les valeurs singulières ainsi que les vecteurs singuliers droits. Les TT-coeurs  $\mathcal{G}_d$  sont ensuite obtenus après un redimensionnement des matrices  $\mathbf{U}_d$ .

L'avantage de la TTD est la stabilité de l'algorithme de décomposition, et le coût de stockage qui est linéaire en  $D$ . Néanmoins, l'algorithme TT-SVD reste un algorithme séquentiel qui n'est pas forcément adapté pour les Big Data, d'où la nécessité de proposer un nouveau algorithme évolutif pour la TTD. En plus, nous essayons dans nos recherches de tirer profit des propriétés de la TTD pour les décompositions usuelles CPD et Tucker, en établissant des relations d'équivalence entre ces modèles. Dans la section suivante, nous présenterons deux résultats dans ce sens.

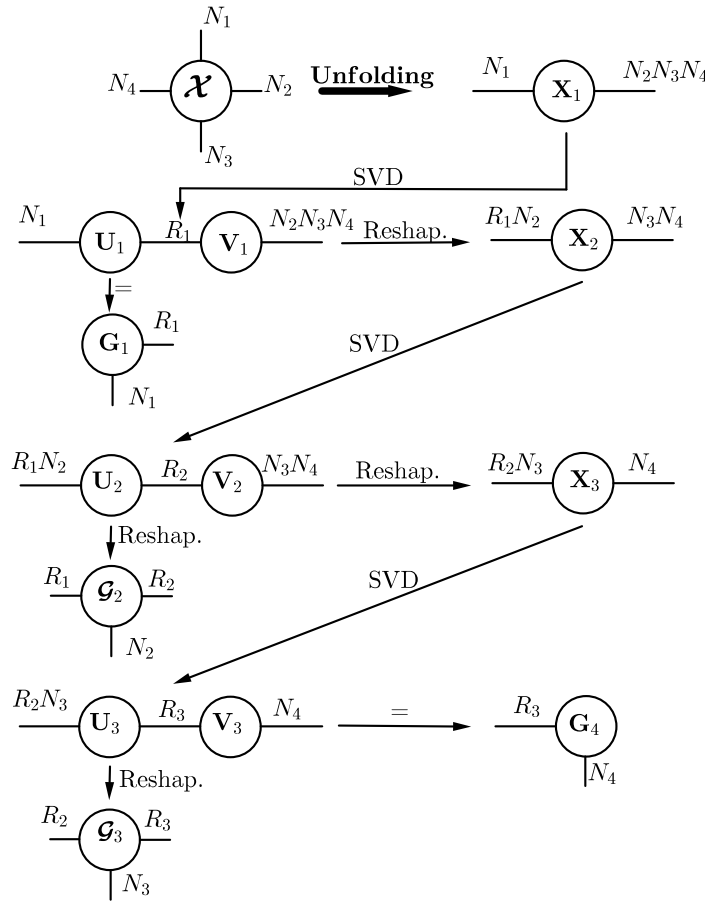


FIGURE 3: L'algorithme TT-SVD appliqué à un tenseur d'ordre 4.

## Résultats

### Un nouveau algorithme hiérarchique: TT-HSVD

L'algorithme TT-HSVD, pour *TT hierarchical SVD*, qu'on propose est un algorithme hiérarchique pour la TTD. C'est un algorithme adapté pour les Big Data, qui donne les mêmes résultats algébriques que l'algorithme TT-SVD, en plus du fait qu'il est adapté pour une architecture parallèle, et est moins complexe. La différence avec l'algorithme TT-SVD est la manière de faire le dépliement des matrices. En effet, le TT-SVD redimensionne le tenseur d'origine en une matrice ayant un mode

dans une dimension et une combinaison des modes restants dans l'autre dimension. Ici, pour le TT-HSVD, nous permetons d'avoir des dépliements en des matrices ayant une combinaison de plusieurs modes sur les deux dimensions de la matrice. Dans la figure 4, nous illustrons l'algorithme TT-HSVD pour un tenseur d'ordre 4. À l'inverse de l'algorithme TT-SVD, le premier dépliement choisit pour l'algorithme TT-HSVD est la matrice  $\mathbf{X}_2$  de dimension  $N_1 N_2 \times N_3 N_4$ . Nous pouvons voir qu'après l'application de la SVD sur cette matrice on obtient 2 matrices  $\mathbf{U}_2$ , contenant les vecteurs singuliers gauches, et  $\mathbf{V}_2$ , contenant les vecteurs singuliers droits et les valeurs singulieres, qui peuvent chacune être traiter séparément.

D'après les résultats algébriques obtenus, nous avons trouvé que si nous avons

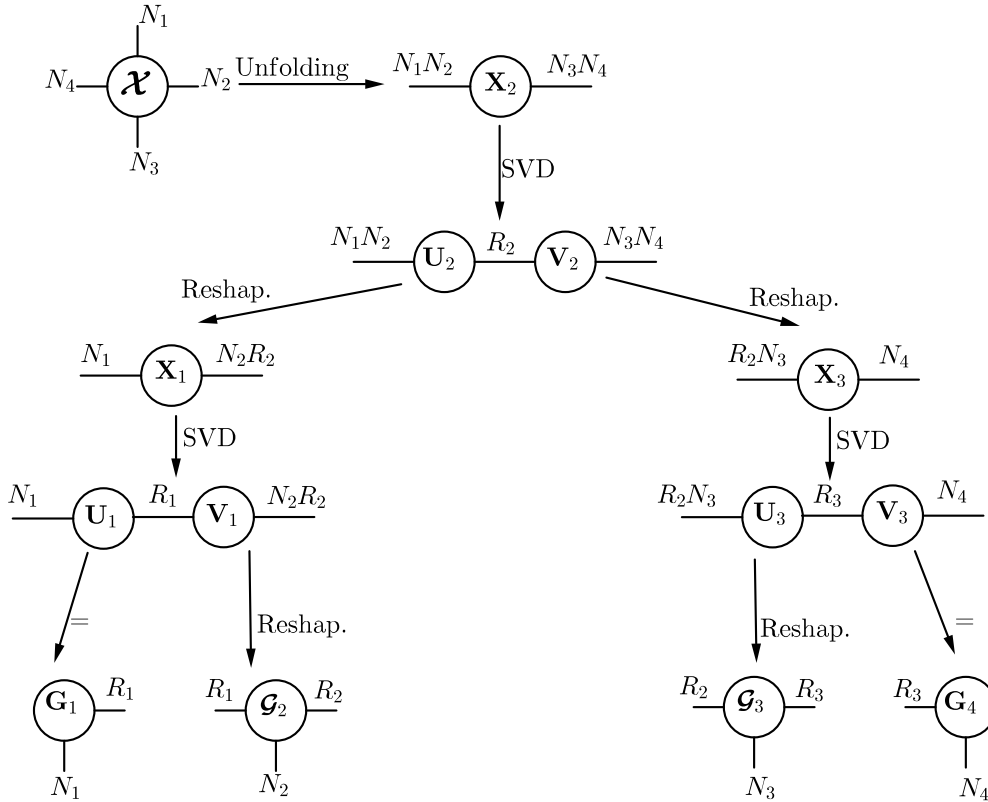


FIGURE 4: L'algorithme TT-HSVD appliqué à un tenseur d'ordre 4.

un tenseur  $\mathcal{X}$  d'ordre  $D$  qui suit un modèle TT de la forme eq. (2) et de rangs TT  $(R_1, \dots, R_{D-1})$ , les deux algorithmes TT-SVD et TT-HSVD vont estimer les mêmes rangs TT, équivalents aux vrais rangs, et des TT-coeurs équivalents aux vrais TT-coeurs à des matrices de changement de base près. D'où les 2 résultats suivants:

$$\begin{aligned}\hat{\mathbf{G}}_1^{seq} &= \mathbf{G}_1 \mathbf{P}_1 \\ \hat{\mathbf{G}}_d^{seq} &= \mathbf{P}_{d-1}^{-1} \times_2^1 \mathbf{G}_d \times_3^1 \mathbf{P}_d, \quad (2 \leq d \leq D-1) \\ \hat{\mathbf{G}}_D^{seq} &= \mathbf{P}_{D-1}^{-1} \mathbf{G}_D.\end{aligned}$$

et

$$\begin{aligned}\hat{\mathbf{G}}_1^{hrl} &= \mathbf{G}_1 \mathbf{Q}_1 \\ \hat{\mathcal{G}}_d^{hrl} &= \mathbf{Q}_{d-1}^{-1} \times_2^1 \mathcal{G}_d \times_3^1 \mathbf{Q}_d, \quad (2 \leq d \leq D-1) \\ \hat{\mathbf{G}}_D^{hrl} &= \mathbf{Q}_{D-1}^{-1} \mathbf{G}_D.\end{aligned}$$

Où  $\mathbf{P}_d$  et  $\mathbf{Q}_d$  sont des matrices de changement de base, de dimension  $R_d \times R_d$ . Graphiquement, Nous pouvons combiner les TT-coeurs estimés par les algorithmes TT-SVD et TT-HSVD tel que les figures 5 et 6 respectivement.

Donc après cette analyse que nous avons faite pour les deux algorithmes, Nous

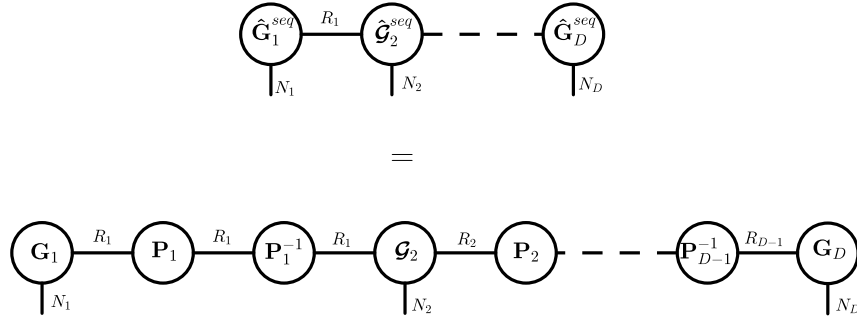


FIGURE 5: Combinaison des TT-coeurs estimés d'une manière séquentielle.

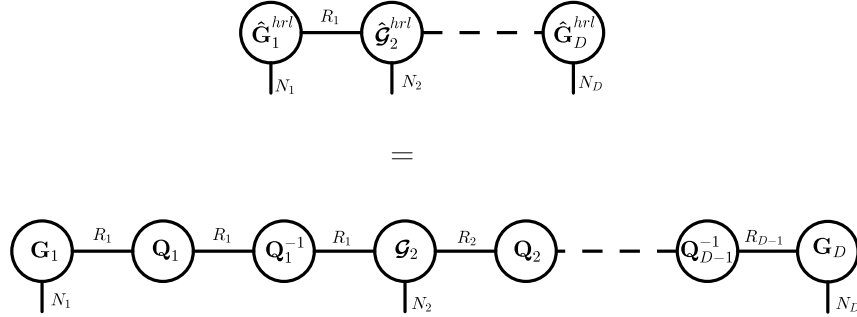


FIGURE 6: Combinaison des TT-coeurs estimés d'une manière hiérarchique.

pouvons voir que les deux algorithmes vont estimer le même tenseur d'origine après la simplification des matrices de changement de base. Avec l'avantage que le TT-HSVD peut être implémenté dans une architecture parallèle, et en plus nous avons trouvé que la complexité computationnelle du TT-SVD est  $\kappa_{seq} = O(RN^D) + O(R^2N^{(D-1)}) + O(R^2N^{(D-2)}) + \dots + O(R^2N^2)$ , alors que le TT-HSVD a une complexité  $\kappa_{hrl} = O(RN^D) + O(R^2N^{\frac{D}{2}}) + O(R^2N^{\frac{D}{4}}) + \dots + O(R^2N^2)$  pour un tenseur d'ordre  $D$  et de dimension  $N \times \dots \times N$ . Donc un gain computationnelle très important quand nous avons un tenseur d'ordre très élevé.

Nous donnons dans le tableau 1, quelques résultats de simulations obtenus. Nous pouvons voir que nous avons un gain très important du TT-HSVD par rapport à TT-SVD, qui augmente significativement avec les dimensions. En particulier, pour  $D = 10$ , l'algorithme TT-SVD n'a pas pu calculer la décomposition vu la taille de

la première matrice de dépliement qui est de taille  $4 \times 4^9$ , ce qui demande la création d’une matrice de vecteurs singuliers droite de taille  $4^9 \times 4^9$  pour la SVD, ce qui dépasse les capacités du calculateur. D’autres résultats de simulations peuvent être trouver dans le papier [157]

TABLE 1: Comparaison du temps de calcul des deux algorithmes ( $D = 8, R = 3$ ).

Les dimensions du tenseur	$N = 9$	$N = 10$	$N = 11$	$N = 12$	$N = 13$
TT-SVD	5.9	13.3	29	88.1	—
TT-HSVD	1.9	4	8.3	17.3	43.9
Gain	3.1	3.3	3.5	5.1	$\infty$

### Équivalence entre CPD/Tucker et TTD: les décompositions CPD-Train et Tucker-Train

Un autre volet de nos recherches vise à tirer profit de la TTD, qui n’est pas très utilisée par la communauté traitement du signal, pour proposer de nouvelles méthodes d’estimation des paramètres d’intérêt des décompositions CPD et Tucker/HOSVD, qui sont les plus connues et utilisées par la communauté. En effet, nous avons établi des relations d’équivalence des décompositions CPD et Tucker avec la TTD. Dans les sections suivantes, les deux équivalence Tucker-Train et CPD-Train seront présentés.

#### Tucker-Train

Une équivalence algébrique a été établit entre le modèle de Tucker et la TTD. En termes de résultats, nous avons trouvé qu’un tenseur d’ordre  $D$  suivant un modèle de Tucker peut être exprimé sous forme d’un train de tenseurs d’ordre 3 suivant aussi un modèle de Tucker et ayant chacun une structure connue. Ce modèle a été appelé Tucker-Train. En plus de l’équivalence algébrique, nous avons trouvé la structure des TT-coeurs résultants de l’algorithme TT-SVD ou TT-HSVD, leur rang multilinéaire et des facteurs communs entre ces tenseurs. L’avantage de cette équivalence est de pouvoir retrouver les paramètres d’intérêts d’un modèle Tucker d’ordre très élevé à partir des TT-coeurs qui sont d’ordre 3. À partir de cette réduction de dimensionnalité, des gains en complexité computationnelle ont été trouvé.

#### CPD-Train

Pareil que le modèle de Tucker. Nous avons trouvé qu’un tenseur suivant une CPD d’ordre  $D$  et de rang canonique  $R$  est équivalent à un train de tenseurs CPD d’ordre 3 et de rang canonique  $R$  aussi. Nous avons trouvé aussi la structure des TT-coeurs résultants des algorithmes de décomposition. En effet, les TT-coeurs ont une structure CPD de rang canonique  $R$  où les facteurs du modèle initial sont les seconds facteurs des TT-coeurs avec des matrices de changement de base  $\mathbf{M}_d$  communes de dimension  $R \times R$  entre les TT-coeurs. Mathématiquement, nous avons les relations

suivantes:

$$\begin{aligned} \mathbf{G}_1 &= \mathbf{P}_1 \mathbf{M}_1^{-1}, \\ \mathcal{G}_d &= \mathcal{I}_{3,R} \times_1 \mathbf{M}_{d-1} \times_2 \mathbf{P}_d \times_3 \mathbf{M}_d^{-T}, \quad 2 \leq d \leq D-1 \\ \mathbf{G}_D &= \mathbf{M}_{D-1} \mathbf{P}_D^T \end{aligned}$$

Graphiquement, nous représentons dans la figure 7 la structure des TT-coeurs résultants de la TTD d'un tenseur CPD de rang canonique  $R$ . Comme application

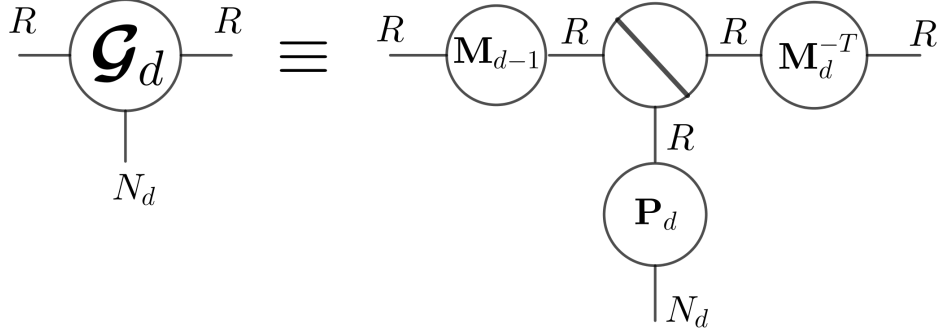


FIGURE 7: CPD d'ordre 3 du *dème* coeur TT.

de ce résultat est le recouvrement des facteurs  $\mathbf{P}_d$  à partir des TT-coeurs en utilisant plusieurs méthodes. Un exemple de méthode de recouvrement est donné dans Algo. 1. Un résultat des simulations est donné dans le Tableau 2, où on compare le temps

---

**Algorithm 1** Réduction de dimensionnalité et recouvrement des facteurs CPD en se basant sur le modèle CPD-train

---

**Entrée:** Tenseur  $\mathcal{X}$  d'ordre  $D$  et de rang canonique  $R$ .

**Sortie:** Facteurs CPD estimés:  $\hat{\mathbf{P}}_1, \dots, \hat{\mathbf{P}}_D$ .

- 1: Reduction de dimensionnalité: (Estimation des TT-coeurs)

$$[\mathbf{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_{D-1}, \mathbf{G}_D] = \text{TT-SVD}(\mathcal{X}, R).$$

- 2: Recouvrement des facteurs CPD:

$$[\hat{\mathbf{M}}_1, \hat{\mathbf{P}}_2, \hat{\mathbf{M}}_2^{-T}] = \text{Tri-ALS}(\mathcal{G}_2, R).$$

- 3: **for**  $d = 3 \dots Q - 1$  **do**
  - 4:    $[\hat{\mathbf{P}}_d, \hat{\mathbf{M}}_d^{-T}] = \text{Bi-ALS}(\mathcal{G}_d, \hat{\mathbf{M}}_{d-1}, R)$
  - 5: **end for**
  - 6:  $\hat{\mathbf{P}}_1 = \mathbf{G}_1 \hat{\mathbf{M}}_1$ , et  $\hat{\mathbf{P}}_D = \mathbf{G}_D^T \hat{\mathbf{M}}_{D-1}^{-T}$
- 

de calcul de notre méthode avec l'algorithme ALS pour un tenseur CPD d'ordre  $D = 8$  et de rang  $R = 3$ , en variant les dimensions.

Les papiers [156, 157] présente plus de résultats de simulations par rapports au temps de calcul et robustesse de la méthode proposée. D'autres algorithmes adaptés au modèle CPD-Train et traitant plusieurs cas de modèle ont fait sujet de l'article

TABLE 2: Comparaison du temps de calcul pour  $R = 3, D = 8$ .

Dimensions	Algo. 1	ALS	Gain
$N = 4$	0,79 (s)	19,26 (s)	24,37
$N = 5$	0,91 (s)	114,16 (s)	125,45
$N = 6$	1,31 (s)	431,95 (s)	329,73

[157], à savoir des algorithmes semi-itératifs et non-itératifs pour les modèles CPD dans un cas général, et un algorithme parallèle pour les modèles CPDs avec des facteurs ayant une structure Toeplitz.

## Conclusion

Le travail décrit dans cette thèse porte sur le fléau de la dimensionnalité et le défi consistait à reformuler un tenseur d'ordre élevé en un ensemble de tenseurs d'ordre inférieur. En particulier, l'objectif était de proposer de nouvelles méthodes de traitement tensoriel adaptées aux tenseurs Big Data, avec les avantages d'être robustes, évolutives et avec la possibilité d'exécution parallèle. Parmi tous les réseaux de tenseurs existants, le train des tenseurs, en raison de sa simplicité, s'est révélé être un bon modèle pour résoudre ce problème.

Le modèle de train des tenseurs est exploité dans ce travail pour proposer de nouveaux résultats d'équivalence entre les décompositions tensorielles usuelles et modèle en train des tenseurs. En effet, un nouveau estimateur appelé JIRAFE, pour la réduction de dimensionnalité et l'estimation conjointe des facteurs, est proposé. Il s'agit d'un schéma en deux étapes consistant à diviser le tenseur d'ordre supérieur en une collection de plusieurs tenseurs d'ordre inférieur, et ensuite de l'étape d'estimation des facteurs grâce à une nouvelle stratégie d'optimisation.

Nous avons également proposé un nouveau algorithme hiérarchique, appelé TT-HSVD, qui permet de récupérer simultanément les cœurs et les rangs TT d'une manière hiérarchique, afin de faire face à la complexité calculatoire élevée de l'algorithme de l'état de l'art, à savoir le TT-SVD. Cet algorithme est bien adapté aux tenseurs Big Data et constitue une bonne alternative à l'algorithme TT-SVD pour traiter le problème de la malédiction de la dimensionnalité.

Ces deux résultats fondamentaux ont eu des applications dans le contexte de l'analyse spectrale multidimensionnelle et des systèmes de télécommunication à relais.



## Chapter 1

# Introduction

### 1.1 Context of study and motivation

Tensors are multidimensional data arrays that allow to treat several “modalities” of data dependently. Doing otherwise, hidden relationships or inter-correlations between the modes may be totally missed, which is clearly a reductive approach. In this last decade, tensor decompositions interest has been expanded to different fields, *e.g.*, signal processing, numerical linear algebra, data mining, graph analysis and more. This interest for tensor decompositions is justified by the fact that this concept allows to decompose high-order tensors into physically interpretable and meaningful lower-order factors usually under mild conditions without the need to add constraints of the decompositions, which is a real advantage over matrix models. This puts tensor decompositions and multi-linear algebra as logical extensions of the matrices and linear algebra. However, this generalization to the high-order tensors is not straightforward since multiple definitions in the matrix case are not the same for tensors, as for example, the rank definition, where different non-equivalent rank definitions exist in the tensor case, each corresponding to a different tensor model unlike the matrix case.

In practice, measurements are usually corrupted by noise, and a best low rank approximations are wanted. Unfortunately, these approximations are generally ill-posed. This ill-posedness results in the stability [92] (non existence of closed-form solution) of the numerical computation of some tensor decompositions, such as the Canonical Polyadic decomposition [63, 61, 25]. Other tensor decompositions can be computationally based on the Singular Value Decomposition (SVD) and provide stable (*i.e.*, non-iterative) decomposition algorithms, and also provide the best/optimal approximations regarding their matricized versions thank to the Eckart-Young theorem, but without the guarantee of optimality at the tensor level, such as the High-order SVD (HOSVD) [89, 5]. This last representation may also suffer from a huge computational and memory resources need. This is due to the fact that the computational complexity and the storage cost grow exponentially in the number of dimensions of the heterogeneous data, also known as the order. This has been identified as the *Curse of dimensionality*.

Recently, a new scheme of tensor decompositions has been proposed, namely tensor networks [28], with the aim to break the curse of dimensionality [106] by decomposing high-order tensors into distributed sparsely interconnected cores of lower-order. This principle allows to develop more sophisticated tensor models with the ability of performing distributed computing and capturing multiple interactions and couplings, instead of the standard tensor decompositions. These tools found applications in very large-scale problems in scientific computing, such as super-compression, tensor completion, blind source separation, and machine learning, to mention a few. of Multiple tensor networks exist, *e.g.*, Hierarchical Tucker [59],



which is an alternative to the Tucker decomposition where the core tensor is replaced by lower-order tensors resulting in a network where some cores are directly connected with factor matrices. Different configurations of networks may exist for a given high-order tensor.

One of the simplest networks is the Tensor Train (TT) decomposition [105]. This decomposition can be considered as a special case of the Hierarchical Tucker one, where all the tensors of the underlying networks are aligned (*i.e.*, no cycles exist in the representation), and where the leaf matrices are all equal to the identity. The two main advantages of this decomposition are its compactness, its storage cost that is linear in the order, and its computation, since it relies on a stable (*i.e.*, non-iterative) SVD-based algorithm called the TT-SVD algorithm [105]. However, the TT-SVD can still be computationally an expensive algorithm when dealing with big data tensors, since it is a sequential algorithm. Moreover, the tensor train decomposition is still not widely used by the signal processing community in the estimation problems.

The goal of this thesis is to take advantage of the assets of the tensor train decomposition to be able to propose new estimation schemes for the problems usually modeled by the classical tensor models, especially for high-order tensors. Also, the goal would be to improve the big data tensor processing by developing some new scalable and hierarchical schemes adapted for high-order tensor decompositions. In this thesis, the equivalence between the classical tensor decompositions and the tensor train decomposition is investigated, while emphasizing the core tensors structure and the relations between the different notions of ranks, to see if some new TT-based estimation schemes are possible. Also, the parallelization of the TT computation is studied, with the aim to apply it to some realistic applications naturally modeled with the tensor train decomposition.

## 1.2 Achieved results

The results obtained in this thesis can be divided in two parts. Both parts have methodological and application results. In the following, we will give a brief summary about each part.

- In the first part, we provide results on the equivalence between both the CPD and the TD, and the TT decomposition for an estimation purpose [157, 156]. These results are constructive and focused on the TT-ranks and TT-cores structure when the TT-SVD algorithm is applied to a CPD or a TD. From a methodological perspective, this result allows to propose a new estimation scheme for high-order tensors, called JIRAFE for Joint dImensionality Reduction And Factor rEtrieval. Several JIRAFE-based algorithms are proposed in this part having advantages in terms of storage cost, computational complexity, algorithmic stability, and factor estimation robustness to noise. Also some results on the partial and full uniqueness [161] are given in the case where JIRAFE is considered for factors having linear dependencies.

Regarding the applications of these results. Since these results can be applied to any high-order CPD-based problems, we choose to apply JIRAFE to the multidimensional harmonic retrieval (MHR) problem. The classic MHR problem was reformulated under a Vandermonde based TT format [158]. The TT-cores structure is provided, explained and exploited in the context of a sequential optimization strategy of a sum of coupled LS problems. Finally, a new rectification strategy for Vandermonde factors is given. In this application, JIRAFE

showed advantages in terms of flexibility, robustness to noise, computational cost and automatic pairing of the parameters with respect to the tensor order. A second application of JIRAFE was the decomposition of MIMO channels [160]. In this application, we use the JIRAFE framework to find the transmit and receive spatial signatures as well as the complex path loss from a new extended channel tensor model.

- In the second part and from a methodological perspective, a new TT-based hierarchical framework [155] for big data tensors was proposed. This new algorithm is called TT-HSVD for Tensor Train Hierarchical SVD. This algorithm delivers simultaneously the TT-cores and the TT-ranks in a hierarchical way by considering a new reshaping strategy different from that of the state-of-the-art TT-SVD algorithm. This reshaping strategy allows to parallelize and deal with smaller matrices in the computation of the TT-cores. An algebraic analysis of the two algorithms shows that both, the TT-SVD and the TT-HSVD, compute the same TT-ranks and TT-core tensors up to a specific bases. This algorithm can be considered in all the applications needing a TT decomposition instead of the TT-SVD, including in JIRAFE.

As an application, this framework is applied to MIMO-OFDM relaying systems [159], allowing a joint channel and symbol estimation.

### 1.3 Manuscript structure

The presented document relies mainly on the three following chapters.

- Chapter 2 is dedicated to the state-of-the-art. We first recall some tensor-related basic definitions and lemmas which are required for the understanding of the manuscript and the proofs development. Then, tensor decompositions existing in the literature will be exposed, namely the CPD, the TD, and the HOSVD. After that, the curse of dimensionality is discussed and the TT decomposition will be defined as a solution to break the curse of dimensionality.
- Chapter 3 is divided to four main sections. The first methodological section details the JIRAFE principle. In this section both CPD-TT and TD-TT equivalences are exposed, and the JIRAFE-based algorithms are given for both cases. The second section discusses the JIRAFE principle in the case of PARALIND tensors. Uniqueness conditions in the case of TT-PARALIND are given and discussed in this section. The two remaining sections are applications. The third one presents a new Vandermonde-based tensor train representation for the multidimensional harmonic retrieval problem. It is an application of JIRAFE in the case of structured high-order CPD. Finally, an application for the decomposition of MIMO channel tensors. In this last section, the CPD-TT equivalence is reformulated with respect to the structure of the channel tensor. The TT-cores structure in this case change giving the fact that the rank factor assumptions are different compared to those of the first section, and a JIRAFE-based algorithm is proposed.
- Chapter 4 details the new TT-based hierarchical framework for big data tensors. In this section, a parallel algorithm for the computation of the TT-cores is proposed, called TT-HSVD. This algorithm compute the same TT-ranks and

TT-cores up to specific bases as the TT-SVD algorithm. TT-HSVD has the ability to allow a parallel processing and has a less computational complexity compared to the TT-SVD. Also, a new graph-based representation using patterns is detailed in this section. An application of this framework can be the MIMO-OFDM relaying systems that is exposed in the second section of this chapter. In this second section, we propose a new TT-based receiver allowing a joint channels and symbol estimation.

The conclusions and prospects of this work are given in Chapter 5.

## 1.4 Publications

The work described in this document has led to the following publications.

### Peer-reviewed international journals:

- [J1] **Y. Zniyed**, R. Boyer, A.L.F. de Almedia and G. Favier, *High-order tensor factorization via trains of coupled third-order CP and Tucker decompositions*, 2019, to appear in Linear Algebra and its Applications (LAA).
- [J2] **Y. Zniyed**, R. Boyer, A.L.F. de Almedia and G. Favier, *A TT-based hierarchical framework for decomposing high-order tensors*, SIAM Journal on Scientific Computing (SISC), 2019, accepted for publication.
- [J3] **Y. Zniyed**, R. Boyer, A.L.F. de Almedia and G. Favier, *Multidimensional Harmonic Retrieval Based on Vandermonde Tensor Train*, Elsevier, Signal Processing (SP), vol. 163, pp.75-86, 2019.
- [J4] **Y. Zniyed**, R. Boyer, A.L.F. de Almedia and G. Favier, *Tensor Train Representation of Massive MIMO Channels using the Joint Dimensionality Reduction and Factor Retrieval (JIRAFE) Method*, Elsevier, Signal Processing (SP), 2019, submitted.

### International conferences:

- [C1] **Y. Zniyed**, R. Boyer, A.L.F. de Almedia and G. Favier, *High-Order CPD Estimation with Dimensionality Reduction Using A Tensor Train Model*, EUSIPCO, Rome, Italy, 2018.
- [C2] **Y. Zniyed**, R. Boyer, A.L.F. de Almedia and G. Favier, *Tensor-Train Modeling for MIMO-OFDM Tensor Coding-And-Forwarding Relay Systems*, EUSIPCO conference, 2019, accepted in Advances on tensor and multi-dimensional data representation session.
- [C3] **Y. Zniyed**, S. Miron, R. Boyer and D. Brie, *Uniqueness of Tensor Train Decomposition with Linear Dependencies*, submitted to the special session Structured matrix and tensor methods at CAMSAP'2019.
- [C4] A. Boudehane, **Y. Zniyed**, A. Tenenhaus, L. Le Brusquet and R. Boyer, *Breaking The Curse of Dimensionality for Coupled Matrix-Tensor Factorization*, submitted to the special session Structured matrix and tensor methods at CAMSAP'2019.

### National conferences:

- [C5] Y. Zniyed, S. Miron, R. Boyer and D. Brie, *Uniqueness of Tensor Train Decomposition with Linear Dependencies*, GRETSI, Lille, 2019.

**Seminars:**

- [S1] “Model Reduction and Factor Estimation With Tensor Factor Graph”, GT PASADENA seminar, Gif-sur-Yvette, 2017.
- [S2] “Big tensor factorization using graph-based representation”, Ph.D. students day of the Laboratory of signals and systems, Gif-sur-Yvette, 2018.
- [S3] “Joint dimensionality reduction and factor retrieval: principle and applications”, journée GDR, Ivry-sur-Seine, 2019.



## Chapter 2

# State-of-the-art and Problem setup

## 2.1 Tensor-related algebraic background

In this part, we will give all the basic and necessary definitions in relation with tensors that will be useful for the understanding and the proof of theorems later in the manuscript.

**Definition 1.** A  $Q$ -order tensor  $\mathcal{X}$  of size  $I_1 \times \cdots \times I_Q$  is an element of the finite-dimension real vector space  $\mathbb{R}^{I_1 \times \cdots \times I_Q}$ . It is uniquely defined by the set of its coordinates  $\mathcal{X}_{i_1, \dots, i_Q}$ , where  $1 \leq i_q \leq I_q$  for  $1 \leq q \leq Q$ .

Below, one gives the definitions of the different products that will be used in the sequel

**Definition 2.** The outer product of three vectors  $\mathbf{a}$  ( $I \times 1$ ),  $\mathbf{b}$  ( $J \times 1$ ) and  $\mathbf{c}$  ( $K \times 1$ ) is an  $I \times J \times K$  three-way tensor  $\mathbf{a} \circ \mathbf{b} \circ \mathbf{c}$  with elements  $(\mathbf{a} \circ \mathbf{b} \circ \mathbf{c})_{i,j,k} = a_i b_j c_k$ .

Note that for the case of two vectors we have  $\mathbf{a} \circ \mathbf{b} = \mathbf{a}\mathbf{b}^T$ .

**Definition 3.** Let  $\mathbf{A}$  and  $\mathbf{B}$  be two matrices of size, respectively,  $I \times J$  and  $K \times L$ , the Kronecker product  $\mathbf{A} \otimes \mathbf{B}$  of size  $(IK) \times (JL)$  is given by

$$\mathbf{A} \otimes \mathbf{B} = \begin{pmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \cdots \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

**Definition 4.** The Khatri-Rao product  $\mathbf{A} \odot \mathbf{B}$  is a subset of columns from  $\mathbf{A} \otimes \mathbf{B}$ , it contains the product of any column of  $\mathbf{A}$  with only the corresponding column of  $\mathbf{B}$  instead of the interaction with all columns. It is a product between two matrices of same number of columns. Let  $\mathbf{A} = [\mathbf{a}_1 \ \mathbf{a}_2 \ \cdots \ \mathbf{a}_R]$  and  $\mathbf{B} = [\mathbf{b}_1 \ \mathbf{b}_2 \ \cdots \ \mathbf{b}_R]$ , the Khatri-Rao product of  $\mathbf{A}$  and  $\mathbf{B}$  is defined as

$$\mathbf{A} \odot \mathbf{B} = [\mathbf{a}_1 \otimes \mathbf{b}_1 \ \mathbf{a}_2 \otimes \mathbf{b}_2 \ \cdots \ \mathbf{a}_R \otimes \mathbf{b}_R].$$

Another important notion in the tensor representations is the  $n$ -mode product.

**Definition 5.** The  $n$ -mode product of a tensor  $\mathcal{X}$  of size  $I_1 \times I_2 \times \cdots \times I_N$  with a matrix  $\mathbf{A}$  of size  $J \times I_n$  is denoted by  $\mathcal{X} \times_n \mathbf{A}$  and is of size  $I_1 \times \cdots \times I_{n-1} \times J \times I_{n+1} \times \cdots \times I_N$ . And we have

$$(\mathcal{X} \times_n \mathbf{A})_{i_1 \cdots i_{n-1} j i_{n+1} \cdots i_N} = \sum_{i_n=1}^{I_n} x_{i_1 i_2 \cdots i_N} u_{j i_n}.$$

This product has the following properties:

- $\mathcal{X} \times_m \mathbf{A} \times_n \mathbf{B} = \mathcal{X} \times_n \mathbf{B} \times_m \mathbf{A} \quad (m \neq n).$
- $\mathcal{X} \times_n \mathbf{A} \times_n \mathbf{B} = \mathcal{X} \times_n (\mathbf{BA}).$

**Definition 6.** The contraction product  $\times_q^p$  between two tensors  $\mathcal{A}$  and  $\mathcal{B}$  of size  $N_1 \times \cdots \times N_Q$  and  $M_1 \times \cdots \times M_P$ , where  $N_q = M_p$ , is a tensor of order  $Q + P - 2$  such as [28]

$$\begin{aligned} & [\mathcal{A} \times_q^p \mathcal{B}]_{n_1, \dots, n_{q-1}, n_{q+1}, \dots, n_Q, m_1, \dots, m_{p-1}, m_{p+1}, \dots, m_P} \\ &= \sum_{k=1}^{N_q} [\mathcal{A}]_{n_1, \dots, n_{q-1}, k, n_{q+1}, \dots, n_Q} [\mathcal{B}]_{m_1, \dots, m_{p-1}, k, m_{p+1}, \dots, m_P}. \end{aligned}$$

Using the outer product definition, we can define a rank-one tensor as follows.

**Definition 7.** As the rank-1 matrix, and regarding the outer product definition, a  $Q$ -order rank-1 tensor  $\mathcal{X}$  is defined as the outer product of  $N$  vectors, i.e.,  $\mathcal{X} = \mathbf{a}_1 \circ \mathbf{a}_2 \circ \cdots \circ \mathbf{a}_Q$ .

We now introduce some standard lemmas and definitions that will be useful in the sequel, especially for the proofs of our main Theorems.

**Definition 8.** Let  $R \leq N$ . If matrices  $\mathbf{A}$  and  $\mathbf{B}$  of size  $N \times R$  span the same  $R$ -dimensional column space, then it exists a nonsingular  $R \times R$  matrix  $\mathbf{M}$ , usually called a change-of-basis matrix, such as

$$\mathbf{B} = \mathbf{A}\mathbf{M} \text{ or equivalently } \mathbf{B}\mathbf{M}^{-1} = \mathbf{A}.$$

In this case, we have the following equality in terms of orthogonal projectors

$$\mathbf{B}\mathbf{B}^\dagger = \mathbf{A}\mathbf{M}\mathbf{M}^{-1}\mathbf{A}^\dagger = \mathbf{A}\mathbf{A}^\dagger.$$

**Lemma 1.** Let  $\mathbf{A}$  and  $\mathbf{B}$  be two matrices of respective sizes  $I \times J$  and  $K \times J$ , where  $I \geq J$ , and  $K \geq J$ . The Khatri-Rao product of both matrices does not decrease the rank [131]:

$$\text{rank}(\mathbf{A} \odot \mathbf{B}) \geq \max\{\text{rank}(\mathbf{A}), \text{rank}(\mathbf{B})\},$$

which implies that, if  $\mathbf{A}$  or  $\mathbf{B}$  have full rank,  $\mathbf{A} \odot \mathbf{B}$  also has full rank, i.e.,

$$\text{rank}(\mathbf{A}) = \text{rank}(\mathbf{B}) = J \Rightarrow \text{rank}(\mathbf{A} \odot \mathbf{B}) = J.$$

In tensor-based data processing, it is standard to unfold a tensor into matrices. We refer to Eq. (5) in [50], for a general matrix unfolding formula, also called tensor reshaping. The  $q$ -th generalized unfolding  $\mathbf{X}_{(q)}$ , of size  $(\prod_{s=1}^q N_s) \times (\prod_{s=q+1}^Q N_s)$ , of the tensor  $\mathcal{X} \in \mathbb{R}^{N_1 \times N_2 \times \cdots \times N_Q}$  using the native reshape function of the MATLAB software, is defined by:

$$\mathbf{X}_{(q)} = \text{reshape} \left( \mathcal{X}; \prod_{s=1}^q N_s, \prod_{s=q+1}^Q N_s \right).$$

Another type of reshaping is the tensor unfolding, that transforms the tensor  $\mathcal{X} \in \mathbb{R}^{N_1 \times N_2 \times \cdots \times N_Q}$  into a 3-order tensor  $\mathcal{X}_q$  of size  $(N_1 \cdots N_{q-1}) \times N_q \times (N_{q+1} \cdots N_Q)$  where:

$$\mathcal{X}_q = \text{reshape} \left( \mathcal{X}; \prod_{s=1}^{q-1} N_s, N_q, \prod_{s=q+1}^Q N_s \right).$$

The last transformation is the square matrix tensorization, that transforms a square matrix  $\mathbf{I} \in \mathbb{R}^{N \times N}$ , with  $N = N_1 N_2$ , to a 3-order tensor according to:

$$\mathcal{T} = \text{reshape}(\mathbf{I}; N_1, N_2, N), \quad (2.1)$$

or,

$$\tilde{\mathcal{T}} = \text{reshape}(\mathbf{I}; N, N_1, N_2), \quad (2.2)$$

where  $\mathcal{T} \in \mathbb{R}^{N_1 \times N_2 \times N}$  and  $\tilde{\mathcal{T}} \in \mathbb{R}^{N \times N_1 \times N_2}$ . These two last reshaping will be used in Theorem 1 to define a one-to-one mapping between an identity matrix and a sparse tensor filled with binary values.

In this work, factor graph representations will be intensively used. In a factor graph, a node can be a vector, a matrix or a tensor as illustrated in Fig. 2.1-(a), (b), (c). The edge encodes the dimension of the node. The order of the node is given by the number of edges. In Fig. 2.1-(d), the  $\times_{\frac{1}{3}}$  product of two 3-order tensors with a common dimension is illustrated.

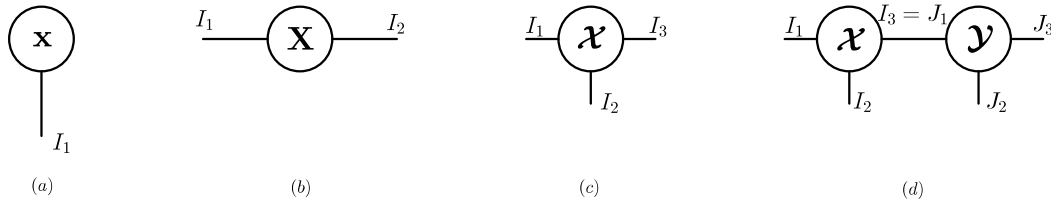


FIGURE 2.1: (a) Vector  $\mathbf{x}$  of size  $I_1 \times 1$ , (b) matrix  $\mathbf{X}$  of size  $I_1 \times I_2$ , (c) 3-order tensor  $\mathcal{X}$  of size  $I_1 \times I_2 \times I_3$ , (d)  $\times_{\frac{1}{3}}$  product of two 3-order tensors.

## 2.2 Standard tensor decompositions and low rank tensor approximation

In this section, we will recall the standard tensor decompositions, namely the Canonical polyadic decomposition (CPD), Tucker decomposition (TD) and high-order SVD (HOSVD), and discuss some their drawbacks regarding the computation and the storage cost before presenting the new tensor network paradigm in the next section.

### 2.2.1 CPD

In the literature [57, 79, 73, 130], two decompositions are popular, sharing a common goal, *i.e.*, to decompose a tensor  $\mathcal{X}$  into a  $D$ -order core tensor  $\mathcal{G}$  and  $D$  factor matrices. Due to the diagonality of the core tensor, the CPD [63, 61, 25] extends the definition of rank used in linear algebra to multilinear algebra. Indeed, the canonical rank of a tensor  $\mathcal{X}$  is defined as the smallest number  $R$  of rank-1 tensors necessary to yield  $\mathcal{X}$  in a linear combination. As a consequence, the number of free parameters (as well as the storage cost) of the CPD grows linearly with the tensor order  $D$ .



**Definition 9.** A  $Q$ -order tensor of size  $N_1 \times \dots \times N_Q$  belonging to the family of rank- $R$  CPD admits the following decomposition:

$$\mathcal{X} = \mathcal{I}_{Q,R} \times_1 \mathbf{P}_1 \times_2 \mathbf{P}_2 \times_3 \dots \times_Q \mathbf{P}_Q \quad (2.3)$$

where the  $q$ -th factor  $\mathbf{P}_q$  is of size  $N_q \times R$ ,  $1 \leq q \leq Q$ . The  $q$ -mode unfolded matrix  $\text{unfold}_q \mathcal{X}$ , of size  $N_q \times \frac{N_1 \dots N_Q}{N_q}$ , is given by:

$$\text{unfold}_q \mathcal{X} = \mathbf{P}_q \cdot (\mathbf{P}_Q \odot \dots \odot \mathbf{P}_{q+1} \odot \mathbf{P}_{q-1} \odot \dots \odot \mathbf{P}_1)^T.$$

Unfortunately, without any additional structural assumptions [93, 153, 114, 111, 19], the problems of the recovery of (i) the canonical rank is NP-hard [62] and (ii) the factor matrices is ill-posed [42, 78]. This means that (i) there is no guarantee on the existence of a low-rank approximation of a tensor by means of its rank- $R$  CPD and (ii) a stable and efficient algorithm to compute the factor matrices may not exist, or it performs poorly when dealing with big data tensors.

Below is an important lemma in relation to CPD that will be used in the sequel.

**Lemma 2.** For the CPD (2.3), the  $q$ -th reshaping, denoted by  $\mathbf{X}_{(q)}$ , of size  $(N_1 \dots N_q) \times (N_{q+1} \dots N_Q)$ , admits the following expression:

$$\begin{aligned} \mathbf{X}_{(q)} &= \text{reshape}(\mathcal{X}; N_1 \dots N_q, N_{q+1} \dots N_Q) \\ &= (\mathbf{P}_q \odot \dots \odot \mathbf{P}_1) \cdot (\mathbf{P}_Q \odot \dots \odot \mathbf{P}_{q+1})^T. \end{aligned}$$

## 2.2.2 Tucker decomposition

The Tucker decomposition was proposed in [142]. It decomposes a tensor into a core tensor of same order, multiplied by a factor matrix along each mode. It can be seen as a generalization of the CPD [63, 61, 25].

**Definition 10.** A  $Q$ -order tensor of size  $N_1 \times \dots \times N_Q$  that follows a Tucker decomposition can be written as:

$$\mathcal{X} = \mathcal{C} \times_1 \mathbf{F}_1 \times_2 \mathbf{F}_2 \times_3 \dots \times_Q \mathbf{F}_Q \quad (2.4)$$

where  $\mathbf{F}_q$  is of size  $N_q \times T_q$ ,  $1 \leq q \leq Q$ , and  $\mathcal{C}$  is the core tensor of size  $T_1 \times \dots \times T_Q$ . The multilinear rank [38] of the tensor  $\mathcal{X}$  is defined as the  $Q$ -uplet  $\{T_1, \dots, T_Q\}$ , such that  $T_1 \times \dots \times T_Q$  is the minimal possible size of the core tensor  $\mathcal{C}$ . The storage cost of a Tucker decomposition is  $O(QNT + T^Q)$ , where  $N = \max\{N_1, \dots, N_Q\}$ , and  $T = \max\{T_1, \dots, T_Q\}$ .

## 2.2.3 HOSVD

The TD/HOSVD [89, 5] is the second approach for decomposing a high-order tensor. In the case of the HOSVD, the  $D$  factor matrices are obtained from a low-rank approximation of the unfolding matrices of the tensor, which is possible by means of the SVD, under orthonormality constraint on the factor matrices. Unfortunately, this orthonormality constraint implies that the core tensor  $\mathcal{G}$  is generally not diagonal. Two remarks can be made at this point. First, unlike the SVD [53], the HOSVD is a multilinear rank-revealing decomposition [10] but does not reveal the canonical rank. Second, the storage cost associated with the computation of the core tensor grows exponentially with the order  $D$  of the data tensor. From this brief panorama,

we can conclude that the CPD and the HOSVD are not the appropriate solutions to deal with high-order big data tensors, and more efficient decompositions should be considered.

**Definition 11.** The HOSVD [38, 5] is a special case of the Tucker decomposition (2.4), where the factors  $\mathbf{F}_q$  are column-orthonormal. Generally, the HOSVD is not optimal in terms of low rank approximation, but it relies on the stable (i.e., non-iterative), SVD-based algorithm. In the real case, the core tensor can be expressed as

$$\mathcal{C} = \mathcal{X} \times_1 \mathbf{F}_1^T \times_2 \mathbf{F}_2^T \times_3 \dots \times_Q \mathbf{F}_Q^T$$

Note that the core tensor  $\mathcal{C}$  obeys to the “all-orthogonality” property [38].

**Lemma 3.** The  $q$ -th generalized unfolding, denoted by  $\mathbf{X}_{(q)}$ , of a tensor  $\mathcal{X}$  that follows (2.4), of size  $(N_1 \dots N_q) \times (N_{q+1} \dots N_Q)$ , admits the following expression:

$$\begin{aligned} \mathbf{X}_{(q)} &= \text{reshape}(\mathcal{X}; N_1 \dots N_q, N_{q+1} \dots N_Q) \\ &= (\mathbf{F}_q \otimes \dots \otimes \mathbf{F}_1) \cdot \mathbf{C}_{(q)} \cdot (\mathbf{F}_Q \otimes \dots \otimes \mathbf{F}_{q+1})^T, \end{aligned}$$

where  $\mathbf{C}_{(q)} = \text{reshape}(\mathcal{C}; T_1 \dots T_q, T_{q+1} \dots T_Q)$ .

### 2.2.4 Low rank tensor approximation

Low rank tensor approximation is one of the major challenges in the information scientific community, see for instance [130, 32]. It is useful to extract relevant information confined into a small dimensional subspace from a massive volume of data while reducing the computational cost. In the matrix case, the approximation of a full rank matrix by a (fixed) low rank matrix is a well posed problem. The famous Eckart-Young theorem [45] provides both theoretical guarantees on the existence of a solution and a convenient way to compute it. Specifically, the set of low rank matrices is closed and the best low rank approximation (in the sense of the Frobenius norm) is obtained via the truncated Singular Value Decomposition (SVD). This fundamental result is at the origin of the Principal Component Analysis (PCA) [53], for instance. PCA is based on the decomposition of a set of observations into a set of uncorrelated variables. When the measurements are naturally modeled according to more than two axes of variations, i.e., in the case of high-order tensors, the problem of obtaining a low rank approximation faces a number of practical and fundamental difficulties. Indeed, even if some aspects of tensor algebra can be considered as mature, several algebraic concepts such as decomposition uniqueness, rank determination, or the notions of singular values and eigenvalues remain challenging research topics [9, 39]. To illustrate these conceptual difficulties and without being too exhaustive, we will address the non-uniqueness of the rank through the description of the canonical rank and of the multilinear rank.

A natural generalization to high-order tensors of the usual concept of matrix rank leads to the canonical polyadic decomposition (CPD) [63, 61, 25, 44, 137]. The canonical rank of a  $Q$ -order tensor is equal to the minimal number, say  $R$ , of rank-one tensors that must be linearly combined to reach a perfect recovery of the initial tensor. A rank-one tensor of order  $Q$  is given by the outer product of  $Q$  vectors. In the context of massive data processing and analysis, this decomposition and its variants [28, 31] are attractive in terms of compactness thanks to the minimality constraint on  $R$ . In addition, the CPD has remarkable uniqueness properties [137] and involves only  $QNR$  free parameters for a  $Q$ -order rank- $R$  tensor of size  $N \times \dots \times N$ . Unfortunately,

unlike the matrix case, the set of tensors with low (tensorial) rank is not closed [62]. This singularity implies that the problem of computing the CPD is mathematically ill-posed. The consequence is that its numerical computation remains non trivial and is usually done using suboptimal iterative algorithms [92]. Note that this problem can sometimes be avoided by exploiting some natural hidden structures in the physical model [133], or by considering some constraints such as the coherence constraint as in [124].

The Tucker decomposition and the HOSVD (High-Order SVD) [38, 142] are two popular decompositions being an alternative to the CPD. In this case, the notion of canonical rank is no longer relevant and a new rank definition has to be introduced. Specifically, the multilinear rank of a tensor is defined as the set of  $Q$  positive integers :  $\{T_1, \dots, T_Q\}$  where each  $T_q$  is the usual (in the matrix sense) rank of the  $q$ -th mode unfolding of this tensor. Its practical construction is algebraic, non-iterative and optimal in the sense of the Eckart-Young theorem, applied to each matrix unfolding. This approach became popular because it can be computed in real time or adaptively, using very standard algebraic methods [17]. However, a low (multilinear) rank tensor based on the HOSVD is generally not optimal regarding the approximation in the Frobenius norm sense. In other words, there is no generalization of the Eckart-Young theorem for tensors of order strictly greater than two. This decomposition has a poor compactness property compared to the CPD. For a multilinear rank  $\{T, \dots, T\}$ , the number of free parameters is about  $QNT + T^Q$  for a  $Q$ -order tensor of size  $N \times \dots \times N$ , and therefore grows exponentially with the order  $Q$ . In the context of the massive data processing, this decomposition is irrelevant and cannot break the curse of dimensionality [106]. Indeed, a generally not sparse core tensor must be stored in the Tucker/HOSVD decomposition, leading to a term in  $T^Q$ .

## 2.3 Problem setup

### 2.3.1 The curse of dimensionality and tensor networks

One of the main challenges in high-order  $Q$  tensor processing is the *curse of dimensionality*. This problem has been identified by Oseledets and Tyrtshnikov in [106] and it refers to the fact that the storage cost and the complexity of processing required to solve high-order tensor problems both grow exponentially with the order  $Q$ . This is the case of the HOSVD where the core tensor is generally not sparse and requires a storage cost exponential in  $Q$ . However, this is not the case for the CPD which has a storage cost linear in  $Q$ , but its computing is mathematically ill-posed which results in a non trivial numerical computation, usually done by iterative algorithms. Unfortunately, these techniques may require several iterations to converge [118, 74], and convergence is increasingly difficult when the order of the tensor increases [21] and it is not even guaranteed [92]. This puts us in a dilemma regarding the use of state-of-the-art methods when we are dealing with high-order tensors. This dilemma/limitation can be presented by the following table.

	storage cost <i>w.r.t.</i> $Q$	stability
CPD	linear	<b>NO</b>
TD/HOSVD	<b>EXPONENTIAL</b>	yes

This means that, in one hand, the CPD is a very compact decomposition, but has some numerical computation issues, and in the other hand, the TD/HOSVD is a stable (*i.e.*, non-iterative) decomposition that relies on a SVD-based algorithm, but

its storage cost and computational complexity remains a big issue. To alleviate this problems, different new properties are desired:

- Stable (*i.e.*, non-iterative) algorithms.
- Low storage cost, *i.e.*, number of free parameters linear in  $Q$ .
- Formalism allowing a simple but rigorous representation of high-order tensors.

Recently, a new paradigm for dimensionality reduction has been proposed [28], therein referred to as tensor networks (TNs). The main idea of TNs [7, 31] is to split a high-order ( $D > 3$ ) tensor into a product set of lower-order tensors represented as a factor graph. Factor graphs allow visualizing the factorization of multi-variate or multi-linear functions (the nodes) and their dependencies (the edges) [90]. The graph formalism is useful in the big data context [125].

In this context, higher-order tensor networks give us the opportunity to develop more sophisticated models performing distributed computing and capturing multiple interactions and couplings, instead of standard pairwise interactions. In other words, to discover hidden components within multiway data the analysis tools should account for intrinsic multi-dimensional distributed patterns present in the data. Tensor network diagrams are very useful not only in visualizing tensor decompositions, but also in their different transformations/reshapings and graphical illustrations of mathematical (multilinear) operations. In general, tensor networks can be considered as a generalization and extension of standard tensor decompositions and are promising tools for the analysis of big data due to their extremely good compression abilities and distributed and parallel processing.

Particularly, the Tensor Train (TT) decomposition [105, 103, 57] is one the simplest tensor networks. It represents a  $D$ -order tensor by a product set of  $D$  3-order tensors. Each 3-order tensor is associated to a node of the graph and is connected to its left and right “neighbors” encoded in a one-to-one directional edge [102]. The storage cost with respect to the order  $D$  has the same linear behavior [105] for the CPD and the TT decomposition. Moreover, the TT decomposition has a stable (non-iterative) SVD-based algorithm [105], even for large tensors. Therefore, the TT decomposition helps to break the *curse of dimensionality* [106], as it can be seen as a special case of the hierarchical/tree Tucker (HT) decomposition [56, 57, 144, 132, 106]. Note that the HT and the TT decompositions allow to represent a  $D$ -order tensor of size  $I \times \dots \times I$  with a storage cost of  $O(DIR + DR^3)$  and  $O(DIR^2)$ , respectively [55, 28], where  $R$  is the rank of the considered decomposition. The use of the TT decomposition is motivated by its applicability in a number of interesting problems, such as super-compression [72], tensor completion [81], blind source separation [13], fast SVD of large scale matrices [91], for linear diffusion operators [71], and in machine learning [130], to mention a few.

### 2.3.2 Graph-based illustrations of TNs

Fig. 2.2-(a) gives the graph-based representation of a Tucker decomposition of a 6-order tensor. Its graph-based decompositions are given in Fig(s). 2.2-(b). The decomposition of the initial tensor is not unique since we have several possible graph-based configurations depending on the tensor order and the number of connected nodes. In the literature, the graph on the right of Fig. 2.2-(b) is viewed as a Tensor

Network (TN), also called Hierarchical Tucker (HT) decomposition [59]. This is due to the property that a TN can be alternatively represented as a tree (see Fig. 2.2-(c)) where the nodes are called leaves. The graph-based representation on the left of Fig. 2.2-(b) is viewed as a train of tensors with non-identity leaves. We call a *Tensor Train* (TT) decomposition [106, 105] if all the leaves are associated with an identity matrix except the first and last ones. The TT decomposition is one of the simplest and compact TN, and it is a special case of the HT decomposition [144, 56, 57]. Hereafter, we give the definition of the TT decomposition, and its graph-based representation is illustrated in Fig. 2.3.

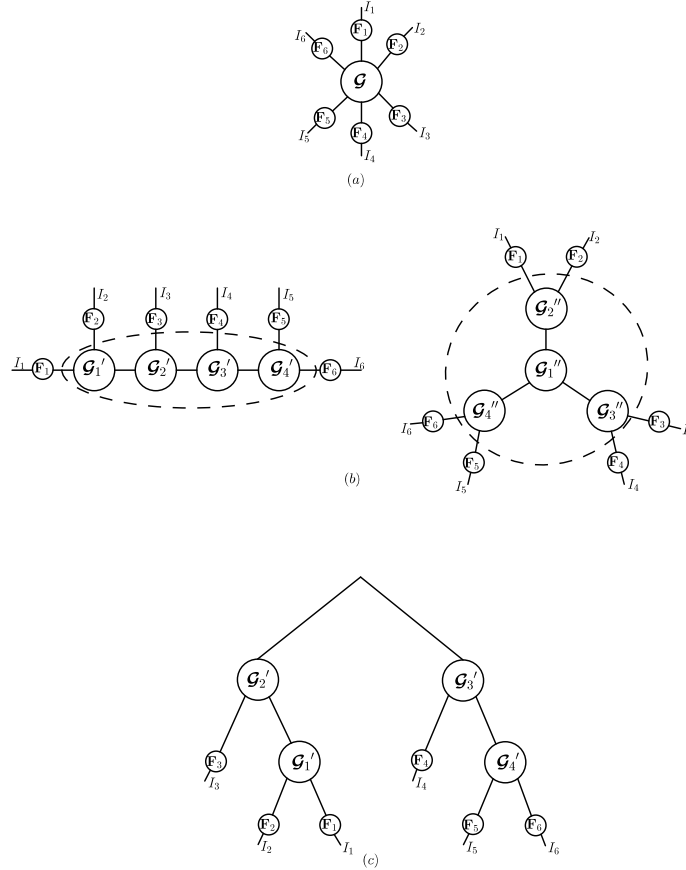


FIGURE 2.2: (a) Graph-based Tucker decomposition of a 6-order tensor, (b) TT-based (left) and TN-based (right) decompositions, (c) HT decomposition.

### 2.3.3 TT decomposition and tensor reshaping

**Definition 12.** The TT decomposition [105] with TT-ranks  $(R_1, \dots, R_{D-1})$  of a tensor  $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_D}$  into  $D$  3-order TT-core tensors denoted by  $\{\mathcal{G}_1, \dots, \mathcal{G}_D\}$  is given by

$$\begin{aligned} \mathcal{X}(i_1, i_2, \dots, i_D) &= \mathcal{G}_1(:, i_1, :) \mathcal{G}_2(:, i_2, :) \cdots \mathcal{G}_D(:, i_D, :) \\ &= \sum_{r_1, \dots, r_{D-1}=1}^{R_1, \dots, R_{D-1}} \mathbf{G}_1(i_1, r_1) \mathcal{G}_2(r_1, i_2, r_2) \cdots \\ &\quad \cdots \mathcal{G}_{D-1}(r_{D-2}, i_{D-1}, r_{D-1}) \mathbf{G}_D(r_{D-1}, i_D). \end{aligned}$$

where  $\mathcal{G}_d(:, i_d, :)$  is a  $R_{d-1} \times R_d$  matrix with the “boundary conditions”  $R_0 = R_D = 1$ . Finally, collecting all the entries<sup>1</sup>, we have

$$\mathcal{X} = \mathbf{G}_1 \times_2^1 \mathcal{G}_2 \times_3^1 \mathcal{G}_3 \times_4^1 \cdots \times_{D-1}^1 \mathcal{G}_{D-1} \times_D^1 \mathbf{G}_D. \quad (2.5)$$

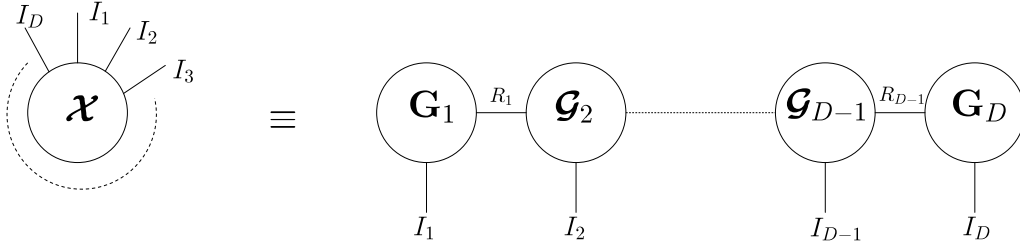


FIGURE 2.3: TT decomposition of a  $D$ -order tensor.

In tensor-based data processing, it is standard to unfold a tensor into matrices. We refer to Eq. (5) in [48], for a general matrix unfolding formula, also called tensor reshaping. The  $d$ -th reshaping  $\mathbf{X}_{(d)}$ , of size  $(\prod_{s=1}^d I_s) \times (\prod_{s=d+1}^D I_s)$ , of the tensor  $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_D}$  using the native reshape function of the software MATLAB [82], is defined by:

$$\mathbf{X}_{(d)} = \text{reshape} \left( \mathcal{X}, \prod_{l=1}^d I_l, \prod_{l=d+1}^D I_l \right). \quad (2.6)$$

**Definition 13.** The  $d$ -th reshaping  $\mathbf{X}_{(d)}$  of the tensor  $\mathcal{X}$  which follows the TT decomposition (2.5) admits the following expression:

$$\mathbf{X}_{(d)} = \sum_{r_1, \dots, r_{D-1}=1}^{R_1, \dots, R_{D-1}} \left( \mathbf{g}_d(r_{d-1}, r_d) \otimes \cdots \otimes \mathbf{g}_1(r_1) \right) \left( \mathbf{g}_D^T(r_{D-1}) \otimes \cdots \otimes \mathbf{g}_{d+1}^T(r_d, r_{d+1}) \right) \quad (2.7)$$

where  $\mathbf{g}_1(r_1) = \mathbf{G}_1(:, r_1)$ ,  $\mathbf{g}_D(r_{D-1}) = \mathbf{G}_D(r_{D-1}, :)^T$  and  $\mathbf{g}_d(r_{d-1}, r_d) = \mathcal{G}_d(r_{d-1}, :, r_d)$  are column-vectors of length  $I_1$ ,  $I_D$  and  $I_d$ , respectively. An alternative expression of (2.5) in terms of sum of outer products of vectors is given by:

$$\mathcal{X} = \sum_{r_1, \dots, r_{D-1}=1}^{R_1, \dots, R_{D-1}} \mathbf{g}_1(r_1) \circ \mathbf{g}_2(r_1, r_2) \circ \cdots \circ \mathbf{g}_{D-1}(r_{D-2}, r_{D-1}) \circ \mathbf{g}_D(r_{D-1}).$$

### 2.3.4 Why to use the TT decomposition ?

There are four main motivations for using a TT decomposition:

- The TT decomposition has a unique graph-based representation for a tensor of known order, i.e., any  $D$ -order tensor can be decomposed into the TT format as a product set of  $D$  TT-cores of order at most 3, with respect to the one configuration where all nodes of the underlying tensor network are aligned. This is

<sup>1</sup>Note that the product  $\times_n^m$  is used here in a different way as in [29].



not the case for the HT decomposition as illustrated by Fig. 2.2-(c). The number of possible configurations rapidly grows with the order  $D$ . For instance, a 10-order tensor admits 11 different HT decompositions [82].

- The ranks for HT are upper bounded by  $R^2$  if the TT-ranks are upper bounded by  $R$  [56]. If equal ranks is assumed and  $I$  is not too large, the number of free parameters in the TT format is smaller than that in the HT format due to the presence of leaves different from the identity matrix in the HT format.
- The TT decomposition storage grows linearly with  $D$ , whereas the HOSVD storage grows exponentially with  $D$ , with an important computational cost.
- The TT decomposition has a compact form, unlike the tree-like decompositions, such as Tree Tucker [106], that requires recursive algorithms based on the competition of Gram matrices, which is complicated to implement [105].

### 2.3.5 Description of the TT-SVD algorithm

In the state-of-the-art literature, the TT decomposition is performed thanks to the TT-SVD algorithm [106]. Fig. 2.4 illustrates this algorithm for a 4-order tensor  $\mathcal{X}$  of size  $I_1 \times I_2 \times I_3 \times I_4$ . The first step is based on the unfolding  $\mathbf{X}_{(1)}$  of size  $I_1 \times I_2 I_3 I_4$ , and consists in the computation of the  $R_1$ -truncated SVD giving  $\mathbf{X}_{(1)} = \mathbf{U}_1 \mathbf{V}_1$ . Note that the diagonal matrix constituted by the singular values is absorbed in  $\mathbf{V}_1$ . The first TT-core  $\mathbf{G}_1$  is directly obtained from  $\mathbf{U}_1$  which contains the left singular vectors associated with the  $R_1$  dominant singular values. The reshaping of  $\mathbf{V}_1$  leads to  $\mathbf{X}_{(2)}$  of size  $R_1 I_2 \times I_3 I_4$  whose a  $R_2$ -truncated SVD gives  $\mathbf{X}_{(2)} = \mathbf{U}_2 \mathbf{V}_2$ . The second TT-core  $\mathbf{G}_2$  is then obtained by a reshaping of  $\mathbf{U}_2$ . Next, the matrix  $\mathbf{X}_{(3)}$  is obtained from a reshaping of  $\mathbf{V}_2$ , and the same processing is repeated. The complete algorithm is described in Fig. 2.4. Note that this process effectively decomposes the original tensor to lower-order tensors of an order maximum 3.

From this simple example, one can conclude that the TT-SVD algorithm consists in sequentially truncating the SVD of a reshaped version of the matrices  $\mathbf{V}_d$ , for  $d = 1, \dots, D - 1$ . It is important to note that each step of the algorithm yields a TT-core, leading to a sequential algorithm which cannot be parallelized.

## 2.4 Conclusion

Several properties for tensor decompositions remain desired, such as stable scalable algorithms, and low storage costs. Unfortunately, the standard tensor decompositions do not provide these properties. Tensor networks, and particularly, the tensor train decomposition seems to be a good solution for this issue.

In the following chapters, we propose some new methodological results on the TTD on its relation with the standard tensor decompositions, and how it can be used to provide stable, flexible estimation scheme in the case of high-order tensors, while keeping low storage costs. Moreover, the TTD decomposition algorithm will be improved using some new reshaping strategies. Simulation results for realistic applications of these results will also be given.

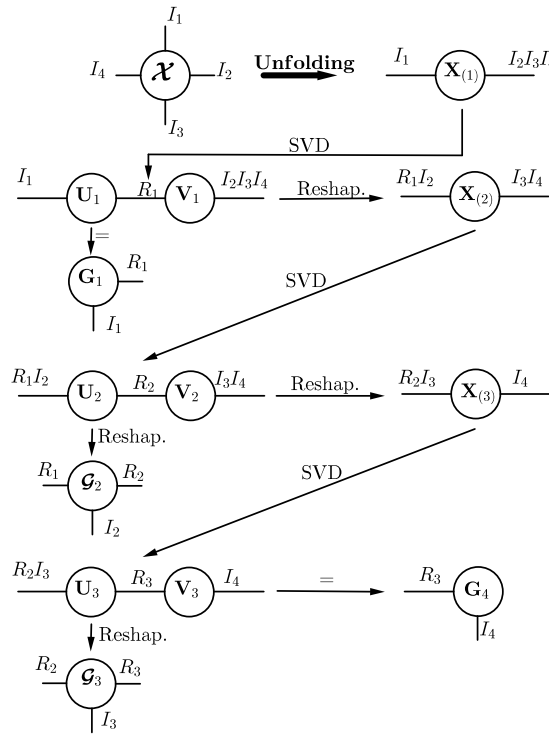


FIGURE 2.4: TT-SVD applied to a 4-order tensor.





## Chapter 3

# Joint dImensionality Reduction And Factor rEtrieval (JIRAFE)

The aim of this chapter is to present new equivalence results [156, 157, 161] between the usual tensor models, namely CPD, TD, PARALIND, and the Tensor Train decomposition. These results will be exploited in an estimation scheme, called JIRAFE for Joint dImensionality Reduction And Factor rEtrieval. Several JIRAFE-based algorithms [156, 157] will be presented and the uniqueness will also be studied in this chapter, before applying JIRAFE on two realistic applications. In the sequel, we will introduce the JIRAFE principle in Section 3.1. The uniqueness in the case of linear dependencies [161] is studied in Section 3.2. Two JIRAFE-based applications, namely the Multidimensional Harmonic Retrieval [158] and MIMO channel decomposition [160], are respectively detailed in sections 3.3 and 3.4.

### 3.1 JIRAFE principle

#### 3.1.1 Introduction

In this part, equivalence relations between a Tensor Train (TT) decomposition [104] and the Canonical Polyadic Decomposition (CPD)/Tucker Decomposition (TD) [63, 61, 25, 142] are investigated. It is shown that a  $Q$ -order tensor following a CPD/TD with  $Q > 3$  can be written using the graph-based formalism as a train of  $Q$  tensors of order at most 3 following the same decomposition as the initial  $Q$ -order tensor. This means that for any practical problem of interest involving the CPD/TD, it exists an equivalent TT-based formulation. As consequence, the TT decomposition deserves to become a fundamental and a more used estimation tool as the standard and popular CPD/TD. In addition, this equivalence allows us to overcome the curse of dimensionality [106] which is one of the main goals in the context of massive/big data processing. Indeed, our methodology has two main advantages. First, we show that the native difficult optimization problem in a  $Q$ -dimensional space can be efficiently solved according to flexible strategies involving  $Q - 2$  optimization problems in (low) 3-dimensional spaces. In the curse of dimensionality, the number of free parameters grows exponentially with  $Q$ . At contrary, our methodology involves a number of free parameters linear with  $Q$  and thus allows to mitigate this problem. Another contribution consists in the proposition of several robust and fast algorithms based on the TT decomposition to accomplish Joint dImensionality Reduction And Factors rEtrieval (JIRAFE) for the CPD/TD. In particular, based on the TT-SVD algorithm [104], we show how to exploit coupling properties existing between two successive TT-cores in the graph-based formalism. The advantages of the proposed approaches in terms of storage cost, computational complexity and factor estimation accuracy are pointed out.

In the sequel, we study the equivalence between the TT decomposition and both CPD and TD for an estimation purpose. Seminal works [104, 33] have shown some links between these decompositions. Our work is different, since we are not aiming to provide algebraic equivalences from a modeling point of view. Our methodology is mainly constructive and focused on the TT-ranks and TT-cores structure when the TT-SVD algorithm is applied to CPD and TD, depending on the rank conditions of the initial tensors, so they can be exploitable in an estimation scheme. This work brings the following original contributions:

- A different TT-based strategy for multilinear projection estimation applied to TD/HOSVD is proposed.
- We provide an algebraic analysis of the TT-SVD algorithm.
- We propose a new methodology called JIRAFE for Joint dImensionality Reduction And Factors rEtrieval to solve a sum of coupled least-square criteria for each TT-core
- New theorems on the analysis of the application of the TT-SVD algorithm to high-order CPD and TD, regarding the resulting TT-cores structures and the rank conditions, are given.
- Constructive proofs of the structure of the TT-cores are developped, while fast, stable and parallel algorithms exploiting this structure are proposed for both the CPD and TD.

All applications using high-order CPD/TD/HOSVD models can be processed using the proposed JIRAFE framework. For example, but not limited to, the JIRAFE framework can be applied to the problem of multilinear harmonic retrieval (MHR) [67], high-order CPD probability mass function tensors [70] in machine learning, and for channel estimation for massive MIMO systems [69] in wireless communications. Some of these applications will be studied in the next sections.

### 3.1.2 TD-Train model: Equivalence between a high-order TD and a train of low-order TDs

In this section, we present the first equivalence between TD and TT decompositions, showing that a high order TD is equivalent to a train of 3-order TDs. In Theorem 1, we present a new algebraic equivalence between the Tucker and TT decompositions. We show how the matrix factors and the core tensor of a TD can be recast into the TT format.

**Theorem 1.** Assume that tensor  $\mathcal{X}$  follows a  $Q$ -order Tucker model of multilinear rank- $(T_1, \dots, T_Q)$ , given by eq. (2.4). The TT decomposition (2.5) of  $\mathcal{X}$  is then given by

$$\begin{aligned}
 \mathbf{G}_1 &= \mathbf{F}_1, \\
 \mathbf{G}_q &= \mathcal{T}_q \times_2 \mathbf{F}_q, (1 < q < \bar{q}) \\
 &\quad \text{with } \mathcal{T}_q = \text{reshape}(\mathbf{I}_{R_q}; T_1 \cdots T_{q-1}, T_q, T_1 \cdots T_q) \\
 \mathbf{G}_{\bar{q}} &= \mathcal{C}_{\bar{q}} \times_2 \mathbf{F}_{\bar{q}}, \\
 &\quad \text{with } \mathcal{C}_{\bar{q}} = \text{reshape}(\mathcal{C}; R_{\bar{q}-1}, T_{\bar{q}}, R_{\bar{q}}) \\
 \mathbf{G}_q &= \tilde{\mathcal{T}}_q \times_2 \mathbf{F}_q, (\bar{q} < q < Q) \\
 &\quad \text{with } \tilde{\mathcal{T}}_q = \text{reshape}(\mathbf{I}_{R_{q-1}}; T_q \cdots T_Q, T_q, T_{q+1} \cdots T_Q) \\
 \mathbf{G}_Q &= \mathbf{F}_Q^T,
 \end{aligned}$$

where  $\mathcal{T}_q$  and  $\bar{\mathcal{T}}_q$  result from the tensorization as defined in (2.1) and (2.2), respectively, and

- $\bar{q}$  is the smallest  $q$  that verifies  $\prod_{i=1}^q T_i \geq \prod_{i=q+1}^Q T_i$ ,
- The TT-ranks verify:  $R_q = \min(\prod_{i=1}^q T_i, \prod_{i=q+1}^Q T_i)$ .

*Proof.* It is straightforward to verify that the TT decomposition of the  $Q$ -order Tucker core  $\mathcal{C}$  takes the following expression:

$$\mathcal{C} = \mathbf{I}_{R_1} \times_2^1 \mathcal{T}_2 \times_3^1 \cdots \times_{\bar{q}-1}^1 \mathcal{T}_{\bar{q}-1} \times_{\bar{q}}^1 \mathcal{C}_{\bar{q}} \times_{\bar{q}+1}^1 \bar{\mathcal{T}}_{\bar{q}+1} \times_{\bar{q}+2}^1 \cdots \times_{Q-1}^1 \bar{\mathcal{T}}_{Q-1} \times_Q^1 \mathbf{I}_{R_{Q-1}} \quad (3.1)$$

where tensors  $\mathcal{T}_q$  and  $\bar{\mathcal{T}}_q$  have been defined in the Theorem and thanks to the reshaping eq. (2.1) and eq. (2.2), respectively. Replacing the TT decomposition of  $\mathcal{C}$  in eq. (2.4), the  $q$ -th 3-order tensor in eq. (3.1) is multiplied in its second mode by its corresponding factor  $\mathbf{F}_q$ . By identifying the final TT-cores, the theorem is proved.  $\square$

Note that all the TT-cores follow a 3-order Tucker1 model, whose the second factor is the  $q$ -th factor of the original TD, hence the name TD-Train. More specifically, for  $q < \bar{q}$  and  $q > \bar{q}$ , the corresponding TT cores have Tucker1 structures whose core tensors have fixed 1's and 0's patterns. We recall that a Tucker1 model is a 3-order Tucker decomposition with two factor matrices equal to identity matrices. For  $q = \bar{q}$ , the associated TT-core has a Tucker1 structure with a core tensor  $\mathcal{C}_{\bar{q}}$  obtained from the core tensor of the original TD. To illustrate this result, consider the TT decomposition of the core tensor  $\mathcal{C}$  in eq. (2.4). Note that  $\bar{q}$  corresponds to the smallest  $q$  such that  $\mathbf{C}_{(q)}$  has at least as many rows as columns. For example, if  $\mathcal{C}$  is a 5-order tensor of size  $2 \times 3 \times 2 \times 4 \times 2$ , then  $\bar{q} = 3$ , since  $2 \times 3 \times 2 > 4 \times 2$ . Another example where  $\mathcal{C}$  is of size  $2 \times 2 \times 3 \times 4 \times 4$  corresponds to  $\bar{q} = 4$ .

The cores of the TD-Train can be recovered by using the TT-SVD algorithm [104] in practice. However, the application of this algorithm will recover the cores up to non-singular transformation matrices according to definition 8. In the following theorem, the structures of the TT-cores associated with a TD-Train are given.

**Theorem 2.** Applying the TT-SVD algorithm on the tensor (2.4), under the assumptions that

- $\mathbf{F}_q$  is a full column rank matrix of size  $N_q \times T_q$  ( $T_q < N_q, \forall q$ ),
- $\mathbf{C}_{(q)} = \text{reshape}(\mathcal{C}; \prod_{i=1}^q T_i, \prod_{i=q+1}^Q T_i)$  has full rank,

allows to recover the TT-cores according to

$$\begin{aligned} \mathbf{G}_1 &= \mathbf{F}_1 \mathbf{M}_1^{-1}, \\ \mathcal{G}_q &= \mathcal{T}_q \times_1 \mathbf{M}_{q-1} \times_2 \mathbf{F}_q \times_3 \mathbf{M}_q^{-T}, \quad (1 < q < \bar{q}) \\ \mathcal{G}_{\bar{q}} &= \mathcal{C}_{\bar{q}} \times_1 \mathbf{M}_{\bar{q}-1} \times_2 \mathbf{F}_{\bar{q}} \times_3 \mathbf{M}_{\bar{q}}^{-T}, \\ \mathcal{G}_q &= \bar{\mathcal{T}}_q \times_1 \mathbf{M}_{q-1} \times_2 \mathbf{F}_q \times_3 \mathbf{M}_q^{-T}, \quad (\bar{q} < q < Q) \\ \mathbf{G}_Q &= \mathbf{M}_{Q-1} \mathbf{F}_Q^T, \end{aligned} \quad (3.2)$$

where  $\mathbf{M}_q$  is a  $R_q \times R_q$  nonsingular change-of-basis matrix, and the quantities  $\mathcal{T}_q$ ,  $\bar{\mathcal{T}}_q$ ,  $\mathcal{C}_{\bar{q}}$  and  $R_q$  are defined in Theorem 1.

*Proof.* See Appendix A □

Note that the TT-cores follow Tucker models with nonsingular transformation matrices along their first and third modes. These matrices compensate each other due to the train format. The proof relies on the TT-SVD algorithm applied to a  $Q$ -order Tucker decomposition.

**Remark.** One may note that, in Theorem 1, no assumptions on the rank of the Tucker core nor the factors are made, thus there is no guarantee that the considered TT-ranks are minimal, i.e., applying a decomposition algorithm such as TT-SVD may estimate different (lower) TT-ranks and provide different TT-cores' structure. In Theorem 2, additional rank assumptions are made, especially on the Tucker core  $\mathcal{C}$ , allowing to guarantee the minimality of the given TT-ranks and to provide constructive/exploitable results from an estimation point of view.

### 3.1.3 CPD-Train model: Equivalence between a high-order CPD and a train of low-order CPD(s)

Following the same methodology as in the previous section, we will draw results about the relation between the CPD and the TT decomposition. The CPD-Train equivalence turns out to be very useful and of broad interest, due to the various applications of the CPD. Note that the idea of rewriting a CPD into the TT format was briefly introduced in seminal works [104, 33]. In [33], Section 4.3 presents a similar result as Theorem 3 without discussing the TT-cores structure after the application of a decomposition algorithm such as the TT-SVD algorithm. In this work, we exploit this idea to propose a new CPD factor retrieval algorithm. Indeed, in Theorem 4, which is the main result to derive the structure of all the proposed algorithms, we discuss and expose the TT-cores structure after the application of the TT-SVD algorithm on a CPD tensor when the factor have full column rank. Moreover, we give a rigorous demonstration for the relation between TT and CPD in an algebraic point of view, and more important, we expose, for the first time, as a constructive proof an useful relation between TT and CPD in the context of the TT-SVD algorithm. Specifically, the coupling factors between consecutive TT-cores takes its explication in the TT-SVD. This important property totally inspires the optimization strategy of the proposed algorithms. To the best of the authors' knowledge, this is the first CPD algorithm that exploits such a link. In the following theorem, we provide an algebraic equivalence between the two decompositions.

**Theorem 3.** *If the tensor  $\mathcal{X}$  follows a  $Q$ -order CPD of rank- $R$  according to (2.3), then the TT decomposition (2.5) is given by [104, 33]:*

$$\begin{aligned} \mathbf{G}_1 &= \mathbf{P}_1, \\ \mathbf{G}_q &= \mathcal{I}_{3,R} \times_2 \mathbf{P}_q \quad (3\text{-order CPD}), \text{ where } 2 \leq q \leq Q-1, \\ \mathbf{G}_Q &= \mathbf{P}_Q^T, \end{aligned}$$

and the TT-ranks are all identical and equal to the canonical rank  $R$ .

*Proof.* The TT decomposition of the  $Q$ -order identity tensor  $\mathcal{I}_{Q,R}$  of size  $R \times \cdots \times R$  involved in the CPD given by (2.3) is

$$\begin{aligned} \mathcal{I}_{Q,R}(i_1, i_2, \dots, i_Q) &= \sum_{r_1, \dots, r_{Q-1}=1}^R \mathbf{I}_R(i_1, r_1) \mathcal{I}_{3,R}(r_1, i_2, r_2) \mathcal{I}_{3,R}(r_2, i_3, r_3) \cdots \\ &\quad \cdots \mathcal{I}_{3,R}(r_{Q-2}, i_{Q-1}, r_{Q-1}) \mathbf{I}_R(r_{Q-1}, i_Q), \end{aligned}$$

with  $1 < i_1, \dots, i_Q < R$ , which can be rewritten in a train format using identity matrices and tensors as

$$\mathcal{I}_{Q,R} = \mathbf{I}_R \times_2^1 \mathcal{I}_{3,R} \times_3^1 \cdots \times_{Q-1}^1 \mathcal{I}_{3,R} \times_Q^1 \mathbf{I}_R.$$

Replacing the above TT decomposition in (2.3), and identifying the TT-cores, we can deduce the result of Theorem 3.  $\square$

It is worth noting that both decompositions have the same number of free parameters,  $O(RNQ)$  for the CPD, and  $O(R^2N(Q-2))$  for the TTD. All the TT-cores follow a 3-order rank- $R$  CPD structure, whose the second factor is the  $q$ -th factor of the original CPD, the other factors being equal to the identity matrix  $\mathbf{I}_R$ . The result of Theorem 3 means that computing a CPD via its associated TT decomposition maintains the model structure while reducing its complexity. In addition, we know the whole structure of the CPD-Train cores, as well as the ranks of the associated 3-order CPDs.

In the same perspective, the application of the TT-SVD algorithm to (2.3) allows to recover the TT-cores up to nonsingular matrices following Definition 8, as established in Theorem 4. It is very important to note that the non-uniqueness property of the TT decomposition suggest that there exists some indeterminate invertible matrices, but in the following theorem, we rigorously, show that these matrices in the context of the TT-SVD algorithm are in fact a set of change-of-basis rank- $R$  matrices essentially linked to the fundamental problem of singular subspace estimation via the SVD. Moreover, it is also important to note that the structure of the TT-cores presented in Theorem 4 is essentially related to the assumption that the factors have full column rank. For instance, in the case of full row rank factors, the structure of the TT-cores will completely change, and we may have 2 factors that are absorbed in the same TT-core, unlike the case we consider and demonstrate in Theorem 4, which shows that the structure of the resultant TT-cores is not trivial and that each separate case needs a new analysis.

**Theorem 4.** *Applying the TT-SVD algorithm to the  $Q$ -order CPD (2.3) allows to recover the TT-cores of (2.3) when the factors  $\mathbf{P}_q$  have full column rank, yielding*

$$\begin{aligned} \mathbf{G}_1 &= \mathbf{P}_1 \mathbf{M}_1^{-1}, \\ \mathcal{G}_q &= \mathcal{I}_{3,R} \times_1 \mathbf{M}_{q-1} \times_2 \mathbf{P}_q \times_3 \mathbf{M}_q^{-T}, \text{ where } 2 \leq q \leq Q-1 \\ \mathbf{G}_Q &= \mathbf{M}_{Q-1} \mathbf{P}_Q^T \end{aligned}$$

where  $\mathbf{M}_q$  is a nonsingular  $R \times R$  change-of-basis matrix.

*Proof.* See Appendix B  $\square$

From Theorem 4, we can see that all TT-cores have a 3-order CPD structure, whose two matrix factors are nonsingular transformation matrices. Note that the CPD does not belong to the cases of Theorem 2, since the generalized unfolding  $\mathcal{I}_{(q)}$  of the CPD core tensor  $\mathcal{I}$  is not a full rank matrix (as supposed in Theorem 2), due to the sparsity of the core tensor. In other words, Theorem 4 is not a special case of Theorem 2. Figure 3.1 depicts the CPD of the TT-core  $\mathcal{G}_q$ .

**Remark.** *Once again, we should mention that the algebraic equivalence of Theorem 3 does not guarantee the minimality of the given TT-ranks, i.e., the algebraic equivalence of Theorem 3 remains true even if some factors are rank deficient ( $\text{rank} \mathbf{P}_q < R$ ), but the estimated TT-ranks and the TT-cores' structure will not be the same when the TT-SVD is applied in that*

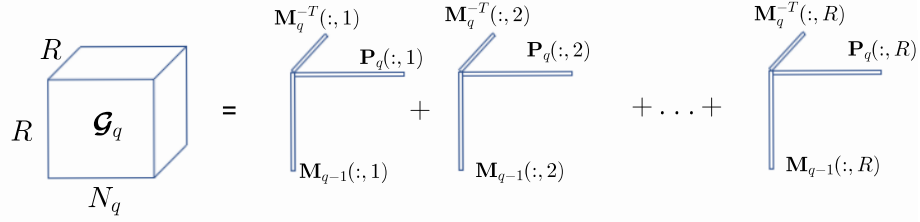


FIGURE 3.1: CPD of the  $q$ -th 3-order core of the associated  $Q$ -order CPD-Train.

case. Thus, additional rank assumptions on the factors are made in Theorem 4 guaranteeing the minimality of the given TT-ranks.

### 3.1.3.1 Symmetric CPD

A special case of the CPD is the fully symmetric CPD tensor, which has found several applications, e.g. in signal processing using high-order statistics [37].

**Definition 14.** A  $Q$ -order symmetric tensor of size  $N \times \dots \times N$  belonging to the family of rank- $R$  CPD admits the following decomposition:

$$\mathcal{X} = \mathcal{I}_{Q,R} \times_1 \mathbf{P} \times_2 \mathbf{P} \times_3 \dots \times_Q \mathbf{P}$$

where  $\mathbf{P}$  is of size  $N \times R$ .

**Theorem 5.** If a  $Q$ -order symmetric tensor admits a rank- $R$  CPD, and  $\mathbf{P}$  has full column rank, we have:

$$\begin{aligned} \mathcal{X} &\stackrel{\text{CPD}}{=} \mathcal{I}_{Q,R} \times_1 \mathbf{P} \times_2 \mathbf{P} \times_3 \dots \times_Q \mathbf{P} \\ &\stackrel{\text{TT}}{=} \mathbf{G}_1 \times_2^1 \mathcal{G}_2 \times_3^1 \mathcal{G}_3 \times_4^1 \dots \times_{Q-1}^1 \mathcal{G}_{Q-1} \times_Q^1 \mathbf{G}_Q, \end{aligned} \quad (3.4)$$

where

$$\mathbf{G}_1 = \mathbf{P} \mathbf{M}_1^{-1} \quad (3.5)$$

$$\mathcal{G}_q = \mathcal{I}_{3,R} \times_1 \mathbf{M}_{q-1} \times_2 \mathbf{P} \times_3 \mathbf{M}_q^{-T} \quad \text{for } 2 \leq q \leq Q-1 \quad (3.6)$$

$$\mathbf{G}_Q = \mathbf{M}_{Q-1} \mathbf{P}^T, \quad (3.7)$$

and  $\mathbf{M}_q$  is a  $R \times R$  matrix that follows Definition 8 ( $1 \leq q \leq Q-1$ ). We can then conclude that:

$$\langle \mathbf{P} \rangle = \langle \mathbf{G}_1 \rangle = \langle \mathcal{G}_q(:, :, i) \rangle = \langle \mathcal{G}_{q'}(j, :, :) \rangle = \langle \mathbf{G}_Q^T \rangle \quad (3.8)$$

for  $2 \leq q \leq Q-1, 2 \leq q' \leq Q-1, 1 \leq i \leq R, 1 \leq j \leq R$ , and we have:

$$\text{rank}(\mathbf{P}) = \text{rank}(\mathbf{G}_1) = \text{rank}(\mathcal{G}_q(:, :, i)) = \text{rank}(\mathcal{G}_{q'}(j, :, :)) = \text{rank}(\mathbf{G}_Q^T) \quad (3.9)$$

which means that, in the symmetric case, the horizontal and frontal slices of the TT-cores span the same subspaces.

*Proof.* See Appendix C. □

### 3.1.3.2 Permutation and scaling ambiguities

It is known that matrix decompositions are not unique unless structural constraints are imposed on the factor matrices. The usual approach is to assume orthonormality by means of the SVD. However, the CPD enjoys essential uniqueness under mild conditions. Specifically, the factors of the 3-order CPD can be identified in a unique manner up to trivial (column permutation and scaling) ambiguities [137].

**Theorem 6.** *The factors of the  $Q$  CPDs associated with a CPD-Train are unique up to the following ambiguities:*

1. common column permutation matrix denoted by  $\Pi$ ;
2. diagonal scaling matrices satisfying the following relation:

$$\Lambda_1 \Lambda_2 \cdots \Lambda_{Q-1} \Lambda_Q = \mathbf{I}_R, \quad (3.10)$$

where  $\Lambda_k$  is the scaling ambiguity for the  $k$ -th factor  $\mathbf{P}_k$ .

*Proof.* 1. It is straightforward to check that if a *change-of-basis* matrix  $\mathbf{M}_k^{-T}$  can be determined up to a column permutation matrix  $\Pi$  and a diagonal scaling matrix  $\Gamma_k$ , then  $\mathbf{M}_k$  can also be determined up to the same permutation matrix and inverse scaling  $\Gamma_k^{-1}$ . We can deduce that the column permutation matrix is unique and common to all the TT-cores due to the recursion property. This proves the first point of the above theorem.

2. We have

$$\begin{aligned} \mathbf{G}_1 &= \mathbf{P}_1 \Pi \Lambda_1 (\mathbf{M}_1 \Pi \Gamma_1^{-1})^{-1} \\ \mathcal{G}_2 &= \mathcal{I}_{3,R} \times_1 \mathbf{M}_1 \Pi \Gamma_1^{-1} \times_2 \mathbf{P}_2 \Pi \Lambda_2 \times_3 \mathbf{M}_2^{-T} \Pi \Gamma_2 \end{aligned} \quad (3.11)$$

$$\mathcal{G}_3 = \mathcal{I}_{3,R} \times_1 \mathbf{M}_2 \Pi \Gamma_2^{-1} \times_2 \mathbf{P}_3 \Pi \Lambda_3 \times_3 \mathbf{M}_3^{-T} \Pi \Gamma_3 \quad (3.12)$$

$\vdots$

$$\mathcal{G}_{Q-1} = \mathcal{I}_{3,R} \times_1 \mathbf{M}_{Q-2} \Pi \Gamma_{Q-2}^{-1} \times_2 \mathbf{P}_{Q-1} \Pi \Lambda_{Q-1} \times_3 \mathbf{M}_{Q-1}^{-T} \Pi \Gamma_{Q-1}$$

$$\mathbf{G}_Q = \mathbf{M}_{Q-1} \Pi \Gamma_{Q-1}^{-1} (\mathbf{P}_Q \Pi \Lambda_Q)^T.$$

Based on the above expressions, we have  $\Gamma_1^{-1} \Lambda_2 \Gamma_2 = \mathbf{I}_R$  from (3.11) and  $\Gamma_2 = \Lambda_3 \Gamma_3$  from (3.12). From these relations, we deduce  $\Gamma_1^{-1} \Lambda_2 \Lambda_3 \Gamma_3 = \mathbf{I}_R$ . Following the same reasoning for the  $k$ -th step we have

$$\begin{aligned} \Gamma_1^{-1} \Lambda_2 \cdots \Lambda_k \Gamma_k &= \mathbf{I}_R, \\ \Gamma_k^{-1} \Lambda_{k+1} \Gamma_{k+1} &= \mathbf{I}_R. \end{aligned}$$

As  $\Gamma_1^{-1} = \Lambda_1$ , and  $\Gamma_{Q-1} = \Lambda_Q$ , combining the above relations for a  $Q$ -order tensor allows to obtain (3.10). □

The result of Theorem 6 is important from an estimation viewpoint. It means that the CPD-Train offers a way to retrieve the factors of a CPD under the same uniqueness properties but at lower complexity, thanks to dimensionality reduction and the *change-of-basis* matrices  $\mathbf{M}_q$ . This is particularly important for high-order tensors, where the direct computation of the factor matrices by means of traditional algorithms such as ALS may be impractical due to processing and storage limitations.



Otherwise, not considering the matrices  $\mathbf{M}_q$  in the estimation will not guarantee the same permutation for the estimated factors, nor verify the equation (3.10) between the scaling matrices. Moreover, these matrices will be used to reduce the complexity by assuming the knowledge/pre-estimation of  $\mathbf{M}_{q-1}$  when decomposing  $\mathcal{G}_q$  in an estimation scheme. Hence the importance of these matrices.

### 3.1.4 Estimation algorithms for the CPD-and TD-Trains

In this section, several algorithms for Joint dImensionality Reduction And Factors rEtrieval (JIRAFE) are presented. These solutions are based on the results of Theorem 4 for the CPD-Train and of Theorem 2 for the TD-Train. So, the JIRAFE methodology can be described by the following two steps procedure.

1. Reduce the dimensionality of the original factor retrieval problem by breaking the difficult multidimensional optimization problem into a collection of simpler optimization problems on small-order tensors. This step is carried out using the TT-SVD algorithm or the TT-HSVD algorithm (see chapter 4).
2. Design a factor retrieval strategy by exploiting (or not) the coupled structure existing in the 1st and 3rd factors for two consecutive TT-cores. Here, the goal is to minimize a sum of coupled least-squares criteria.

#### 3.1.4.1 Fast Multilinear Projection (FMP) based on TD-Train

In many important applications, it is crucial to extract the dominant singular subspaces associated to the factors while the computation of the core tensor is not a primary goal. This scenario is usually known as a multilinear projection as described in [97] for instance. Indeed, in the context of multilinear analysis for facial recognition [146], the physical informations, *i.e.* people  $\times$  expression  $\times$  view  $\times$  illumination are encoded in the factors. Specifically, assume that we dispose of  $Q$  matrices  $\hat{\mathbf{F}}_q$  ( $1 \leq q \leq Q$ ), with  $\hat{\mathbf{F}}_q^T \hat{\mathbf{F}}_q = \mathbf{I}$ . The tensor-to-tensor multilinear projection is formulated according to

$$\mathcal{X}_{\text{proj}} = \mathcal{X}_{\text{data}} \times \hat{\mathbf{F}}_1^T \times_1 \cdots \times_Q \hat{\mathbf{F}}_Q^T.$$

**3.1.4.1.1 Algorithmic description** A fast computation of the orthonormal factors  $\hat{\mathbf{F}}_1, \dots, \hat{\mathbf{F}}_Q$  is presented in this section. Our scheme is based on an alternative interpretation of equations (3.2)-(3.3) such as

$$\begin{aligned} \text{unfold}_2 \mathcal{G}_q &= \mathbf{F}_q \text{unfold}_2 \mathcal{T}'_q, \quad (1 < q < \bar{q}), \\ \text{unfold}_2 \mathcal{G}_{\bar{q}} &= \mathbf{F}_{\bar{q}} \text{unfold}_2 \mathcal{C}'_{\bar{q}}, \\ \text{unfold}_2 \mathcal{G}_q &= \mathbf{F}_q \text{unfold}_2 \bar{\mathcal{T}}'_q, \quad (\bar{q} < q < Q) \end{aligned}$$

where

$$\begin{aligned} \mathcal{T}'_q &= \mathcal{T}_q \times_1 \mathbf{M}_{q-1} \times_2 \mathbf{I} \times_3 \mathbf{M}_q^{-T}, \quad (1 < q < \bar{q}), \\ \mathcal{C}'_{\bar{q}} &= \mathcal{C}_{\bar{q}} \times_1 \mathbf{M}_{\bar{q}-1} \times_2 \mathbf{I} \times_3 \mathbf{M}_{\bar{q}}^{-T}, \quad (q = \bar{q}), \\ \bar{\mathcal{T}}'_q &= \bar{\mathcal{T}}_q \times_1 \mathbf{M}_{q-1} \times_2 \mathbf{I} \times_3 \mathbf{M}_q^{-T}, \quad (\bar{q} < q < Q). \end{aligned}$$

In other words, after the dimensionality reduction based on the TT-SVD algorithm, each projector matrix can be recovered thanks to a truncated SVD of the 2nd unfolding of the corresponding TT-core. A pseudo-code of this new approach is given in Algorithm 2.

---

**Algorithm 2** Fast Multilinear Projection algorithm based on JIRAFE
 

---

**Input:**  $Q$ -order tensor  $\mathcal{X}$

**Output:** Estimated orthonormal factors:  $\hat{\mathbf{F}}_1, \dots, \hat{\mathbf{F}}_Q$ .

1: Dimensionality reduction:

$$[\hat{\mathbf{G}}_1, \hat{\mathbf{G}}_2, \dots, \hat{\mathbf{G}}_{Q-1}, \hat{\mathbf{G}}_Q] = \text{TT-SVD}(\mathcal{X}).$$

Orthonormal Factors retrieval:

2: **for**  $q = 2 \dots Q - 1$  **do**

3:    $\hat{\mathbf{F}}_q = \text{Matrix of the left singular vectors of SVD}(\text{unfold}_2 \hat{\mathbf{G}}_q)$  {This step can be done in a parallel way}

4: **end for**

---

**3.1.4.1.2 On the difference with the HOSVD** It is important to note that using Definition 11, the core tensor  $\hat{\mathcal{C}}$  generated by the above algorithm is given by  $\hat{\mathcal{C}} = \mathcal{X} \times_1 \hat{\mathbf{F}}_1^T \times_2 \dots \times_Q \hat{\mathbf{F}}_Q^T$ . Let be the HOSVD of a tensor  $\mathcal{X} = \mathcal{S} \times_1 \mathbf{U}_1 \times_2 \dots \times_Q \mathbf{U}_Q$  with  $\mathbf{U}_q^T \mathbf{U}_q = \mathbf{I}$ . Using Definition 11, the core tensor for the HOSVD is given by  $\mathcal{S} = \mathcal{X} \times_1 \mathbf{U}_1^T \times_2 \dots \times_Q \mathbf{U}_Q^T$ . It is well-known that the core tensor,  $\mathcal{S}$ , of the HOSVD satisfies the all-orthogonality and pseudo-diagonality properties [38]. On the contrary, we have no guaranty that  $\hat{\mathcal{C}}$  satisfies these properties. Indeed, the TT-SVD algorithm provides an orthonormal matrix  $\hat{\mathbf{F}}_q$  which span the same subspace as the factor  $\mathbf{U}_q$ . This means that there exists a change-of-basis matrix (see Definition 2.1), denoted by  $\mathbf{J}_q$  for instance, such as  $\mathbf{J}_q = \mathbf{U}_q^T \hat{\mathbf{F}}_q$ . The link between the two core tensors is given by

$$\mathcal{S} = \mathcal{X} \times_1 \mathbf{U}_1^T \times_2 \dots \times_Q \mathbf{U}_Q^T = \hat{\mathcal{C}} \times_1 \mathbf{J}_1 \times_2 \dots \times_Q \mathbf{J}_Q.$$

**3.1.4.1.3 Analysis of the computational cost and execution time** The computation of the native HOSVD for a  $Q$ -order Tucker tensor  $\mathcal{X}$  of size  $N \times \dots \times N$  involves the computation of  $Q$  dominant left singular basis thanks to truncated-SVDs on the unfolding matrices of size  $N \times N^{Q-1}$ , other methods, such as ST-HOSVD [145], may use different truncation strategies, while the proposed strategy consists of computing  $Q$  left and right dominant singular basis in the TT-SVD algorithm and  $Q - 2$  left dominant singular basis of the TT-cores. We recall that based on the QR-orthogonal iteration (QR-OI) [53], the  $r$ -truncated SVD computational cost for a single iteration and for the left dominant singular basis for a  $n \times m$  matrix is evaluated to  $2rmn + 2r^2 \min(n, m)$  flops. The computation of the left and the right dominant singular basis is evaluated to  $2rmn + 2r^2(n + m)$  flops.

As an illustrative example, the computational cost of the HOSVD factors of  $(T, \dots, T)$ -multilinear rank for a 4-order tensor is  $8TN^4 + 8T^2N$ . For the TT-SVD, the TT-ranks are  $(T, T^2, T)$  and  $\bar{q} = 2$ . Note that the TT-ranks may be large with respect to the multilinear rank so as shown on Fig. 3.2, it makes sense to consider the case where

$N \gg T$ . Note that this is a standard assumption when using the HOSVD. The complexity cost for the TT-SVD is  $2TN^4 + 2T^2(N + N^3)$  for the first SVD of a  $N \times N^3$  matrix of rank  $T$ . The complexity of the second SVD of a  $(TN) \times N^2$  matrix of rank  $T^2$  is  $2T^3N^3 + 2T^4(TN + N^2)$ . Finally, the last SVD of a  $(T^2N) \times N$  matrix of rank  $T$  is  $2T^3N^2 + 2T^2(T^2N + N)$ . Finally, we have to perform two SVDs of the 2nd unfolding matrices of  $\mathcal{G}_2$  and  $\mathcal{G}_3$  of size  $T \times N \times T^2$  and  $T^2 \times N \times T$ , respectively for the extraction of the left dominant basis. The total cost is  $4NT^4 + 4T^2 \min(N, T^3)$ .

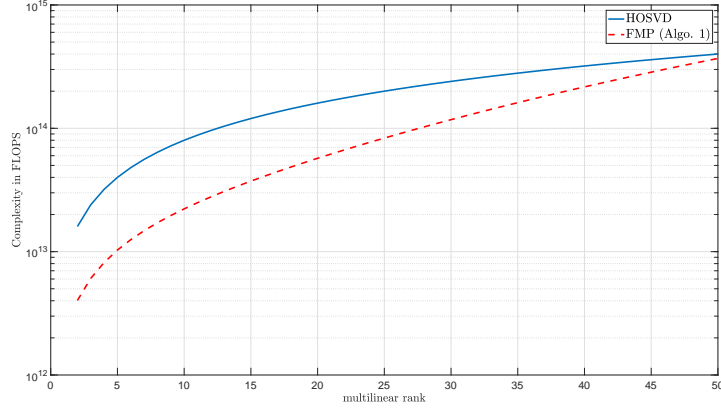


FIGURE 3.2: Number of flops vs the multilinear rank 4-order tensor with  $N = 500$ .

If we generalize this analysis to a  $Q$ -order tensor, we find  $2QTN^Q$  flops for the factors computation of the HOSVD and  $2TN^Q$  for the proposed algorithm. So, the proposed algorithm computational cost is reduced by a factor  $Q$ . In other words, the cost of the computation of the  $Q$  factors by Algo.1 is comparable to the cost of a single factor computation for the HOSVD.

Although the proposed method remains less complex than the HOSVD. The computation of the TT-SVD algorithm still requires the computation of the SVD of a  $N \times N^{Q-1}$  matrix, with a complexity  $O(TN^Q)$ , which can be expensive for large  $Q$ . That is to say that the computation of the TT decomposition is an interesting problem that needs to be more investigated.

In Table 3.1, an hypercubic tensor with  $N = 10$  was generated, with a multilinear rank given by  $T_1 = \dots = T_Q = T = 2$ . Its entries are randomly drawn from a Gaussian distribution with zero mean and unit variance for several tensor orders in a noiseless scenario. The average execution time was evaluated for 20 Monte Carlo realisations. Note that in these simulations, we compute the thin SVD [53].

TABLE 3.1: Computation times for  $(T = 2, N = 10)$

Tensor order	Fast Multilinear Projection	$Q$ -th order HOSVD	Gain
$Q = 6$	0.14 (s)	0.66 (s)	4.71
$Q = 7$	1.31 (s)	7.34 (s)	5.6
$Q = 8$	15.25 (s)	101.87 (s)	6.68

Note that the gain increases when the order increases and is of the order of  $Q$  as remarked before. It is also worth noting that the TT-ranks can actually be large,

but, as demonstrated by the complexity cost analysis, Algorithm 2 is less complex to a native implementation of the HOSVD and faster as demonstrated in the simulations.

The robustness to an i.i.d. Gaussian noise is evaluated thanks to the Normalized MSE defined according to  $\text{NMSE} = \frac{\|\hat{\mathcal{X}} - \mathcal{X}\|_F^2}{\|\mathcal{X}\|_F^2}$ , where  $\hat{\mathcal{X}}$  refers to the estimated tensor.

In Fig. 3.3, the NMSE measurements averaged over 1000 noise realisations are given for the proposed Fast Multilinear Projection and the native HOSVD algorithms for a 6-order hypercubic tensor, with  $N = 4$ , and  $T = 2$ . Note that for a computational gain approximately equal to the tensor order  $Q$ , the same robustness to noise is observed for both algorithms. This result is important since the HOSVD is intensively exploited in numerous applications.

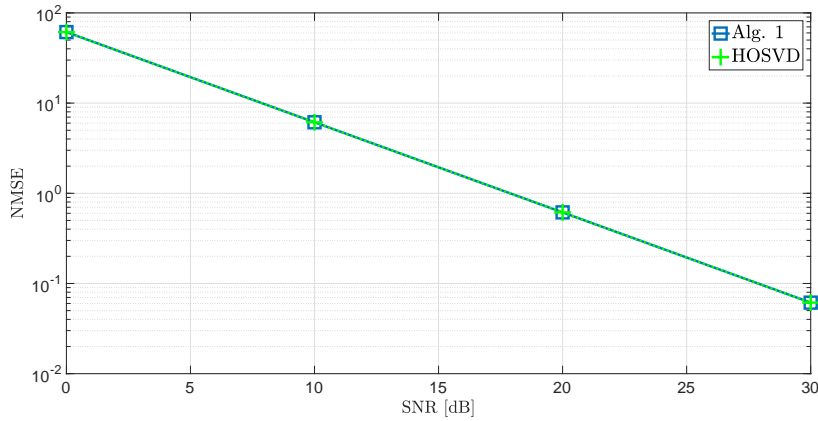


FIGURE 3.3: NMSE vs SNR in dB with Fast Multilinear Projection for a 6-order Tucker with  $N = 4$ ,  $T = 2$ , 1000 runs

#### 3.1.4.2 Fast CPD with CPD-Train

In this section, we provide several algorithms adapted to the CPD-Train. Before presenting the new algorithms, we give hereafter a list of advantages of the proposed CPD-Train equivalence:

1. The CPD-Train has the same number of free parameters  $O(RNQ)$  as the CPD for large  $Q$ .
2. The TT-cores follow a 3-order CPD with canonical rank equal to the rank of the initial  $Q$ -order CPD.
3. Based on the TT-SVD algorithm, a dimensionality reduction is carried out thanks to a non-iterative SVD-based algorithm instead of the standard iterative ALS algorithm.
4. Due to the CPD structure of the TT-cores, the optimization in a high dimensional ( $Q$ ) space is replaced by a collection of much simpler optimization problems, *i.e.*, we have to solve  $(Q - 3)$  optimizations in bi-dimensional spaces and a single optimization in a tri-dimensional space.

**3.1.4.2.1 CPD-Train based on low-order ALS algorithm** The goal of the JIRAFE approach is to optimize the following criterion:

$$\min_{\mathbf{M}, \mathbf{P}} \left\{ \|\mathbf{G}_1 - \mathbf{P}_1 \mathbf{M}_1^{-1}\| + \|\mathbf{G}_Q - \mathbf{M}_{Q-1} \mathbf{P}_Q\| + \sum_{q=2}^{Q-2} \|\mathbf{G}_q - \mathcal{I}_{3,R} \times_1 \mathbf{M}_{q-1} \times_2 \mathbf{P}_q \times_3 \mathbf{M}_q^{-T}\| \right\}$$

where  $\mathbf{M} = \{\mathbf{M}_1, \dots, \mathbf{M}_Q\}$  and  $\mathbf{P} = \{\mathbf{P}_1, \dots, \mathbf{P}_Q\}$ .

Our first proposition is based on the popular ALS-CPD algorithm [25, 61] applied in a sequential way on the CPD-Train cores to jointly retrieve the CPD factors  $\mathbf{P}$  and the *change-of-basis* matrices  $\mathbf{M}$ . A pseudo-code is presented in Algorithm 3, where Tri-ALS stands for the ALS algorithm applied to a 3-order tensor, while Bi-ALS denotes the ALS algorithm applied to a 3-order tensor using *a priori* knowledge of one factor. Note that the ALS approach fixes all but one factor to estimate this latter solving the following least squares problem.

$$\min_{\mathbf{P}_q} \left\| \text{unfold}_q \mathcal{X} - \mathbf{P}_q \cdot (\mathbf{P}_Q \odot \dots \odot \mathbf{P}_{q+1} \odot \mathbf{P}_{q-1} \odot \dots \odot \mathbf{P}_1)^T \right\|^2$$

This procedure is repeated several times until a convergence criterion is satisfied. When one of the 3-order tensor's factors is known, the ALS algorithm only has two steps, which we referred to with the Bi-ALS.

---

**Algorithm 3** JIRAFE based on CPD-Train

---

**Input:**  $Q$ -order rank- $R$  CPD tensor  $\mathcal{X}$

**Output:** Estimated factors:  $\hat{\mathbf{P}}_1, \dots, \hat{\mathbf{P}}_Q$ .

1: Dimensionality reduction:

$$[\hat{\mathbf{G}}_1, \hat{\mathbf{G}}_2, \dots, \hat{\mathbf{G}}_{Q-1}, \hat{\mathbf{G}}_Q] = \text{TT-SVD}(\mathcal{X}, R).$$

2: Factor retrieval:

$$[\hat{\mathbf{M}}_1, \hat{\mathbf{P}}_2, \hat{\mathbf{M}}_2^{-T}] = \text{Tri-ALS}(\hat{\mathbf{G}}_2, R).$$

3: **for**  $k = 3 \dots Q - 1$  **do**

4:    $[\hat{\mathbf{P}}_k, \hat{\mathbf{M}}_k^{-T}] = \text{Bi-ALS}(\hat{\mathbf{G}}_k, \hat{\mathbf{M}}_{k-1}, R)$

5: **end for**

6:  $\hat{\mathbf{P}}_1 = \hat{\mathbf{G}}_1 \hat{\mathbf{M}}_1$ , and  $\hat{\mathbf{P}}_Q = \hat{\mathbf{G}}_Q^T \hat{\mathbf{M}}_{Q-1}^{-T}$

---

The usual strategy is based on the brute force exploitation of a  $Q$ -order ALS-CPD on the original tensor. We note that  $\text{TT-SVD}(\mathcal{X}, R)$  refers to the conventional TT-SVD algorithm with all TT-ranks set to  $R$ . For the complexity analysis, it is assumed that the pseudo-inverse and the SVD have the same complexity order. Note that a single iteration of the  $Q$ -order ALS-CPD requires  $Q$  SVDs of rank- $R$  matrices of size  $N \times N^{Q-1}$ . The complexity is evaluated as  $Q \cdot O(R^2 \cdot N^{Q-1})$ , whereas the TT-SVD applied to the original  $Q$ -order tensor ( $Q \gg 1$ ) has a complexity of  $O(R^2 \cdot N^{Q-1})$ . In addition, the  $R$ -truncated SVD [53] using the orthogonal iteration algorithm is faster to compute than the full rank SVD. The complexity of a  $R$ -truncated SVD for

a  $m \times n$  matrix is  $O(R^2 \max(m, n))$ . Thus, we have

$$\begin{aligned} \kappa(1 \cdot \text{iteration of } Q\text{-order ALS}) &\gg \kappa(Q\text{-order TT-SVD}) = O(R^2 \cdot N^{Q-1}) \\ &\gg \kappa(1 \cdot \text{iteration of 3-order ALS}) = O(3 \cdot R^3 \cdot N). \end{aligned}$$

This means that the proposed strategy is approximately ( $Q$ .number of iterations)-times less complex than the  $Q$ -order ALS-CPD. In Table 3.2, the computation times of the two methods are given for  $N = 6$  and for different tensor orders. The computation time takes into account the algorithmic complexity and the storage costs including the memory acces. To manage the convergence of the ALS-CPD, the stopping criterion is  $\frac{|f(\hat{\mathbf{x}}^{(t)}) - f(\hat{\mathbf{x}}^{(t+1)})|}{f(\hat{\mathbf{x}}^{(t)})} < \epsilon$ , where  $f(\hat{\mathbf{x}}^{(t)}) = \|\mathbf{x} - \hat{\mathbf{x}}^{(t)}\|_F$ , in which  $\hat{\mathbf{x}}^{(t)}$  denotes the estimated tensor at the  $t$ -th iteration and  $\epsilon$  is the convergence threshold. In addition, the number of iterations is limited to 1000. It is clear that the gain strongly increases when the order of the initial tensor grows.

TABLE 3.2: Computation times for ( $R = 3, N = 6$ )

Tensor order	Alg. 3	$Q$ -order ALS-CPD	Gain
$Q = 6$	0,63 (s)	7,48 (s)	11,87
$Q = 7$	1,07 (s)	62,57 (s)	58,47
$Q = 8$	1,31 (s)	431,95 (s)	329,73

In Table 3.3, we fix the order at  $Q = 8$  and vary the tensor dimensions. Here again, the gain of Alg. 3 over ALS-CPD in terms of computation time is measured.

TABLE 3.3: Computation times for ( $R = 3, Q = 8$ )

Tensor dimension	Alg. 3	$Q$ -order ALS-CPD	Gain
$N = 4$	0,79 (s)	19,26 (s)	24,37
$N = 5$	0,91 (s)	114,16 (s)	125,45
$N = 6$	1,31 (s)	431,95 (s)	329,73

To evaluate the noise robustness, an 8-order, rank-2 CPD tensor of size  $N \times \dots \times N$  is generated with  $N = 3$ . In Fig. 3.4, we plot the NMSE obtained with Algorithm 3, the  $Q$ -order ALS algorithm and the popular HOSVD-based preprocessed ALS for 8-order hypercubic CPDs, with  $N = 3$  and  $R = 2$ . The preprocessing step is done thanks to the HOSVD with multilinear ranks  $T = 2$ . The association of a preprocessing (HOSVD) to the ALS algorithm generally improves its robustness to noise while increasing the convergence speed. The NMSEs were averaged over 1000 i.i.d. realisations of a Gaussian noise. To eliminate ill-convergence experiments and outliers, 5% of the worst and 5% of the best NMSE values are discarded.

According to the NMSE results, the  $Q$ -order ALS algorithm is the least robust scheme for SNR lower than 20 dB for a high computational cost. The ALS scheme with the preprocessing step shows an improved noise robustness for a very high computational cost. Finally, the proposed algorithm has the highest accuracy, or equivalently, the best noise robustness with the lowest computational cost.

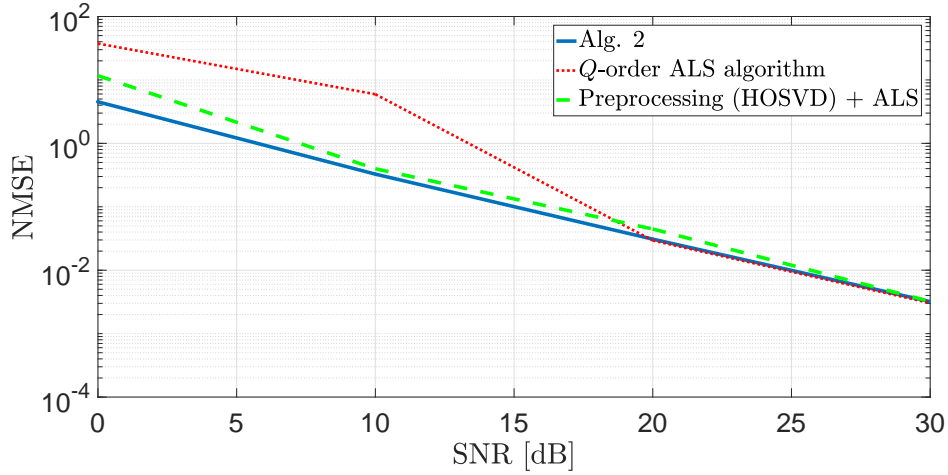


FIGURE 3.4: NMSE vs SNR in dB with JIRAFE for an 8-order CPD

**3.1.4.2.2 Improved CPD-Train** In this section, an improved CPD-Train based factor retrieval algorithm is proposed. In Algorithm 3, by exploiting the repetition of the matrices  $\{\hat{\mathbf{M}}_1, \dots, \hat{\mathbf{M}}_{Q-1}\}$  in the TT-cores, it is possible to replace the iterative Bi-ALS algorithm by the non-iterative Khatri-Rao factorization (KRF) algorithm proposed in [74]. The KRF algorithm is a closed-form algorithm that recovers 3-order CPD factors assuming that one factor is known and has a full-column rank. It computes  $R$  SVDs of rank-one matrices to recover the remaining two other factors. These assumptions match exactly to our case. Note that the matrix  $\hat{\mathbf{M}}_q$  by definition is invertible and thus satisfies the full-column rank assumption. A pseudo-code of the proposed strategy is given in Algorithm 4. Note that Algorithm 4 is thus less sensitive to potential ill-convergence problems than Algorithm 3. In addition, it is worth noting that the complexity of the KRF corresponds to  $\kappa(\text{KRF}) = O(R \cdot N) \ll \kappa(1 \cdot \text{iteration of Bi-ALS}) = O(R^3 \cdot N)$ , which means that the non-iterative KRF is less complex than the iterative Bi-ALS algorithm.

---

**Algorithm 4** Improved CPD-Train algorithm

---

**Input:**  $Q$ -order rank- $R$  CPD tensor  $\mathcal{X}$ ,

**Output:** Estimated factors:  $\hat{\mathbf{P}}_1, \dots, \hat{\mathbf{P}}_Q$ .

1: Dimensionality reduction:

$$[\hat{\mathbf{G}}_1, \hat{\mathcal{G}}_2, \dots, \hat{\mathcal{G}}_{Q-1}, \hat{\mathbf{G}}_Q] = \text{TT-SVD}(\mathcal{X}, R).$$

2: Factor retrieval:

$$[\hat{\mathbf{M}}_1, \hat{\mathbf{P}}_2, \hat{\mathbf{M}}_2^{-T}] = \text{Tri-ALS}(\hat{\mathcal{G}}_2, R).$$

3: **for**  $q = 3 \dots Q - 1$  **do**

4:    $[\hat{\mathbf{P}}_q, \hat{\mathbf{M}}_q^{-T}] = \text{KRF}(\hat{\mathcal{G}}_q, \hat{\mathbf{M}}_{q-1}, R)$

5: **end for**

6:  $\hat{\mathbf{P}}_1 = \hat{\mathbf{G}}_1 \hat{\mathbf{M}}_1$ , and  $\hat{\mathbf{P}}_Q = \hat{\mathbf{G}}_Q^T \hat{\mathbf{M}}_{Q-1}^{-T}$

---

In Fig. 3.5, we plot the NMSE of Algorithm 4 for an 8-order hypercubic CPD with



$N = 3$  and  $R = 2$ . The NMSEs were averaged in the same way as in Fig. 3.4. It is worth noting that Algorithm 4 has an equivalent robustness to noise as Algorithm 3, using a closed-form, non sensitive to ill-convergence problems, KRF method instead of the iterative Bi-ALS algorithm.

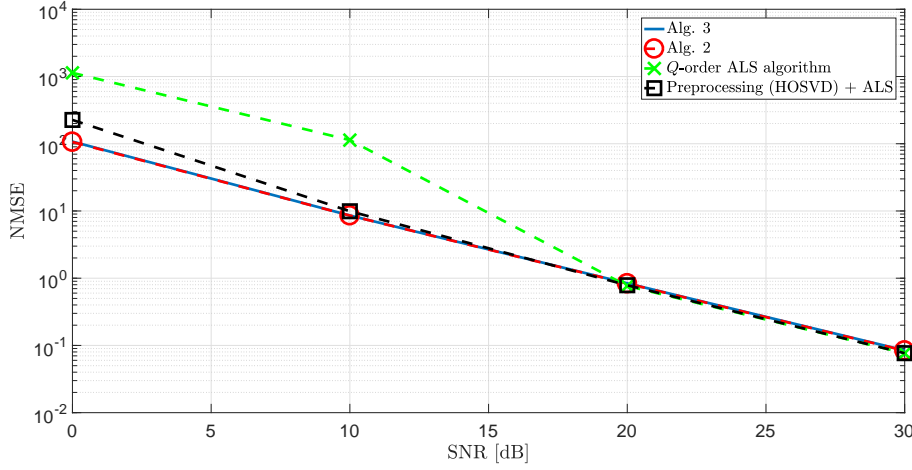


FIGURE 3.5: NMSE vs SNR in dB with improved CPD-Train for an 8-order CPD

**3.1.4.2.3 Non-iterative CPD-Train algorithm in the case of a known factor** In some applications, a factor among  $Q$  can be supposed to be known [50, 151]. In this case, a fully non-iterative algorithm can be derived. Indeed, assume without loss of generality<sup>1</sup> that the first factor,  $\mathbf{P}_1$ , is known. The recovery of  $\mathbf{M}_1$  is straightforward thanks to the pseudo-inverse of the first TT-core. The rest of the method remains identical as for Algorithm 4. A pseudo-code is given in Algorithm 5.

---

**Algorithm 5** Non-iterative CPD-Train algorithm in case of a known factor

---

**Input:**  $Q$ -order rank- $R$  CPD tensor  $\mathcal{X}$ ,  $\mathbf{P}_1$ .

**Output:** Estimated factors:  $\hat{\mathbf{P}}_2, \dots, \hat{\mathbf{P}}_Q$ .

1: Dimensionality reduction:

$$[\hat{\mathbf{G}}_1, \hat{\mathcal{G}}_2, \dots, \hat{\mathcal{G}}_{Q-1}, \hat{\mathbf{G}}_Q] = \text{TT-SVD}(\mathcal{X}, R).$$

2: Factor retrieval:

$$\hat{\mathbf{M}}_1 = \hat{\mathbf{G}}_1^\dagger \mathbf{P}_1$$

3: **for**  $q = 2 \dots Q - 1$  **do**

4:  $[\hat{\mathbf{P}}_q, \hat{\mathbf{M}}_q^{-T}] = \text{KRF}(\hat{\mathcal{G}}_q, \hat{\mathbf{M}}_{q-1}, R)$

5: **end for**

6:  $\hat{\mathbf{P}}_Q = \hat{\mathbf{G}}_Q^T \hat{\mathbf{M}}_{Q-1}^{-T}$

---

<sup>1</sup>Remark that the choice of the index of the factors is totally arbitrary and is meaningless relatively to the model.



**3.1.4.2.4 Parallel and non-iterative CPD-Train algorithm in case of Toeplitz factors** For structured tensors, several algorithms have been proposed by exploiting the structure of the factors in [40, 5]. In some applications [75, 134], the factors of the CPD are Toeplitz. Recall that for a vector  $\mathbf{a} = [a_0 \cdots a_{L-1}]^T$  of length  $L$ , we denote by  $\mathbf{T}(\mathbf{a})$  the  $(L + R - 1) \times R$  Toeplitz matrix that verifies  $\mathbf{T}(\mathbf{a})_{i,j} = a_{i-j}$ , where by convention  $a_{i-j} = 0$  if  $i - j < 0$  or  $i - j > L - 1$ . The choice of this convention is related to applications like Wiener-Hammerstein systems [74], the generalization to the general case is straightforward. Thanks to Theorem 5.2, the TT-cores inherit from this structure. Thus, it makes sense to exploit the TOMFAC algorithm, for TOeplitz Matrix FAcator Computation, proposed in [74]. This leads to the proposition of Algorithm 6. TOMFAC is an algorithm that allows to recover a CPD factor in a closed-form way, considering that it has a Toeplitz structure. As a consequence, Algorithm 6 is fully non-iterative and thus the convergence problem due to the use of an alternated estimation scheme is completely avoided. In addition, the estimation of the Toeplitz factors can be done via a *parallel* processing.

---

**Algorithm 6** Parallel and non-iterative CPD-Train algorithm in case of Toeplitz factors

---

**Input:**  $Q$ -order rank- $R$  CPD tensor  $\mathcal{X}$

**Output:** Estimated factors:  $\hat{\mathbf{P}}_1, \dots, \hat{\mathbf{P}}_Q$ .

1: Dimensionality reduction:

$$[\hat{\mathbf{G}}_1, \hat{\mathbf{G}}_2, \dots, \hat{\mathbf{G}}_{Q-1}, \hat{\mathbf{G}}_Q] = \text{TT-SVD}(\mathcal{X}, R).$$

2: **for**  $q = 1 \cdots Q$  **do**

3:    $\hat{\mathbf{P}}_q = \text{TOMFAC}(\hat{\mathbf{G}}_q, R)$  {This step can be done in a parallel way}

4: **end for**

---

In Table 3.4, the computation times of the native TOMFAC method applied to a  $Q$ -order CPD tensor with Toeplitz factors and of Algorithm 6 are compared. Note that the complexity of the  $Q$ -order TOMFAC is of the same order as the complexity of a  $Q$ -order HOSVD. Here again, interesting gains in terms of computation times are obtained.

TABLE 3.4: Computation times for ( $R = 3, N = 4$ )

Tensor order	CPD-Train-TOMFAC	TOMFAC	Gain
$Q = 6$	0.0296 (s)	0.1206 (s)	4.0743
$Q = 7$	0.2906 (s)	1.6447 (s)	5.6593
$Q = 8$	5.1253 (s)	39.7964 (s)	7.7647

Figure 3.6 depicts the NMSE performance for the  $Q$ -order TOMFAC and Algorithm 6, referred to here as CPD-Train-TOMFAC, for an 8-order hypercubic CPD with  $N = 6$  and  $R = 3$ . The NMSE curves are plotted with respect to the SNR in dB. Note that thanks to the dimensionality reduction step, the CPD-Train-TOMFAC algorithm shows a comparable robustness, for severe SNRs, to the native TOMFAC for a much smaller computational complexity cost.

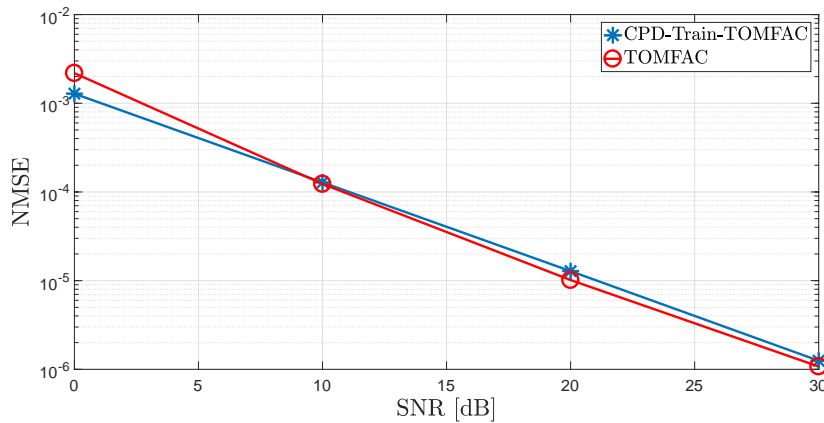


FIGURE 3.6: NMSE vs SNR in dB with CPD-Train-TOMFAC for an 8-order CPD with Toeplitz factors,  $N = 6$ , and  $R = 3$

### 3.1.5 Discussion

We have discussed the joint dimensionality reduction and factor retrieval problem for high-order tensors. We have shown that a  $Q$ -order tensor following a CPD/TD ( $Q > 3$ ) can be written as a train of  $Q$  3-order tensors with a coupled structure. Exploiting this model equivalence property, we have introduced new tensor models, namely the CPD-Train and the Tucker Decomposition (TD)-Train. A two-step JIRAFE methodology has been proposed to overcome the “curse of dimensionality” for high-order tensors. The initial step, called “dimensionality reduction”, consists of splitting the high-order tensor into a collection of graph-connected core tensors of lower orders, at most equal to three. The second step consists of factors retrieval. Several algorithms specialized for the CPD-Train/TD-Train have been proposed, which have a low computational cost and a storage cost that grows linearly with respect to the order of the data tensor. The advantages in terms of storage cost, computational complexity, algorithmic stability, and factor estimation robustness to noise have been demonstrated. JIRAFE can then be seen as a new concept of a more general framework for joint dimensionality reduction and factor retrieval, where different solutions for the factor retrieval step can be considered, such as Gauss-Newton [140] for fitting the CPD, gradient descent [36], or improved versions of ALS such as those based on enhanced line search [115], for instance. Perspective for future works include the application of JIRAFE to big data tensors for multilinear harmonic retrieval problem, massive MIMO systems and coupled matrix-tensor factorizations, besides the investigation of the TT-cores structures when the rank exceeds the dimensions.

## 3.2 Uniqueness of JIRAFE with Linear Dependencies

### 3.2.1 Context of study

In the last section, it has been shown that a canonical rank- $R$  CPD / PARAFAC model can always be represented exactly by a TT model whose cores are canonical rank- $R$  CPD/PARAFAC. This model is called TT-CPD. We generalize this equivalence to the PARALIND model in order to take into account potential linear dependencies in factors. We derive and discuss here uniqueness conditions for the case of the TT-PARALIND model.

Here, we focus on the case where linear dependencies are present between the columns on the factor matrices leading to high-order PARALIND (PARAllel profiles with LINear Dependencies) model [23]. PARALIND is a variant of the CPD with constrained factor matrices, that models a linearly dependent factor  $\mathbf{P}$  as a product of a full column rank matrix  $\tilde{\mathbf{P}}$  and an interaction matrix  $\Phi$ . Matrix  $\Phi$  introduces the linear dependency and rank deficiency in  $\mathbf{P}$ . Linear dependencies in factor matrices are of great interest in real scenarios and can be encountered in chemometrics applications [23] or in array signal processing [154], to mention a few. In this part, some new equivalence results between the TTD and PARALIND are presented. The TT-cores structure is exposed when the  $Q$ -order PARALIND has only two full column rank factor matrices. Partial and full uniqueness conditions for the new TT-PARALIND model are also studied.

### 3.2.2 Equivalence between the PARALIND and the TTD

#### 3.2.2.1 Lack of uniqueness of the TTD

Let  $\mathcal{X}$  be a  $Q$ -order tensor of size  $N_1 \times \dots \times N_Q$  that follows a Tensor Train decomposition (TTD) [104] of TT-ranks  $\{R_1, \dots, R_{Q-1}\}$ :

$$\mathcal{X} = \mathbf{G}_1 \times_2^1 \mathcal{G}_2 \times_3^1 \mathcal{G}_3 \times_4^1 \dots \times_{Q-1}^1 \mathcal{G}_{Q-1} \times_Q^1 \mathbf{G}_Q, \quad (3.13)$$

where the TT-cores  $\mathbf{G}_1, \mathcal{G}_q$ , and  $\mathbf{G}_Q$  are, respectively, of dimensions  $N_1 \times R_1, R_{q-1} \times N_q \times R_q$ , and  $R_{Q-1} \times N_Q$ , for  $2 \leq q \leq Q-1$ , and we have  $\text{rank}\{\mathbf{G}_1\} = R_1$ ,  $\text{rank}\{\mathbf{G}_Q\} = R_{Q-1}$ ,  $\text{rank}\{\text{unfold}_1 \mathcal{G}_q\} = R_{q-1}$ , and  $\text{rank}\{\text{unfold}_3 \mathcal{G}_q\} = R_q$ .

It is straightforward to see that the TTD of  $\mathcal{X}$  in eq. (3.13) is not unique since

$$\mathcal{X} = \mathbf{A}_1 \times_2^1 \mathcal{A}_2 \times_3^1 \mathcal{A}_3 \times_4^1 \dots \times_{Q-1}^1 \mathcal{A}_{Q-1} \times_Q^1 \mathbf{A}_Q,$$

where

$$\begin{aligned} \mathbf{A}_1 &= \mathbf{G}_1 \mathbf{U}_1^{-1}, \\ \mathbf{A}_Q &= \mathbf{U}_{Q-1} \mathbf{G}_Q, \\ \mathcal{A}_q &= \mathbf{U}_{q-1} \times_2^1 \mathcal{G}_q \times_3^1 \mathbf{U}_q^{-1}. \end{aligned}$$

For  $1 \leq q \leq Q-1$ ,  $\mathbf{U}_q$  are square nonsingular matrices of dimension  $R_q \times R_q$ . In practice, the TTD is performed thanks to the state-of-art TT-SVD algorithm [104]. It is a sequential algorithm that recovers the TT-cores  $\mathcal{G}_q$  based on  $(Q-1)$  SVDs applied to several matrix-based reshapings using the original tensor  $\mathcal{X}$ . This algorithm allows to recover the true TT-cores up to a post and pre-multiplication by transformation (*change-of-basis*) matrices due to the extraction of dominant subspaces when

using the SVD. In the next section, we will derive the structure of the estimated TT-cores when the original tensor  $\mathcal{X}$  follows a CPD with linear dependencies between the columns of the factor matrices.

### 3.2.2.2 PARALIND-TTD equivalence

Consider  $Q$ -order tensor  $\mathcal{X}$  of size  $N_1 \times \dots \times N_Q$  that follows a rank- $R$  CPD:

$$\mathcal{X} = \mathcal{I}_{Q,R} \times_1 \mathbf{P}_1 \times_2 \mathbf{P}_2 \dots \times_Q \mathbf{P}_Q, \quad (3.14)$$

where the factor matrices  $\mathbf{P}_q$  are of size  $N_q \times R$ . It was shown in [156] that if the factor matrices  $\mathbf{P}_q$  are full-column rank for  $1 \leq q \leq Q$ , then they can be recovered from the TT-cores by 3-order CPD decompositions.

In this section we study the case where linear dependencies are present between the columns of the factor matrices of (3.14). Thus, a factor matrix  $\mathbf{P}_q$  can be expressed as:

$$\mathbf{P}_q = \tilde{\mathbf{P}}_q \Phi_q, \quad (3.15)$$

where  $\tilde{\mathbf{P}}_q$  is full column rank of size  $N_q \times R_q$  ( $R_q \leq R$ ) and  $\Phi_q$  is a rank deficient matrix of size  $R_q \times R$  containing the dependency pattern between the columns of  $\tilde{\mathbf{P}}_q$ . This CPD model with linear dependencies is also known as *PARALIND* (*PARAllel profiles with LINear Dependencies*) [23].

**Theorem 7** (PARALIND - TTD equivalence). *Decomposing tensor  $\mathcal{X}$  in (3.14) into a TT format, where  $\mathbf{P}_1$  and  $\mathbf{P}_Q$  are full column rank matrices, and  $\mathbf{P}_q$  ( $2 \leq q \leq Q-1$ ) follow (3.15), recovers the estimated TT-cores such that*

$$\begin{aligned} \mathbf{G}_1 &= \mathbf{P}_1 \mathbf{U}_1^{-1}, \\ \mathbf{G}_q &= \mathcal{I}_{3,R} \times_1 \mathbf{U}_{q-1} \times_2 (\tilde{\mathbf{P}}_q \Phi_q) \times_3 \mathbf{U}_q^{-T}, \quad 2 \leq q \leq Q-1 \\ \mathbf{G}_Q &= \mathbf{U}_{Q-1} \mathbf{P}_Q^T, \end{aligned}$$

where, for  $1 \leq q \leq Q-1$ ,  $\mathbf{U}_q$  is a square  $R \times R$  nonsingular matrix. The TT-cores  $\mathbf{G}_1, \mathbf{G}_q$ , and  $\mathbf{G}_Q$  are, respectively, of dimensions  $N_1 \times R, R \times N_q \times R$ , and  $R \times N_Q$ , given TT-ranks all equal to  $R$ .

*Proof.* Note that tensor  $\mathcal{I}_{Q,R}$  in eq. (3.14) can be expressed as

$$\mathcal{I}_{Q,R} = \mathbf{I}_R \times_2^1 \mathcal{I}_{3,R} \times_3^1 \dots \times_{Q-1}^1 \mathcal{I}_{3,R} \times_Q^1 \mathbf{I}_R, \quad (3.16)$$

replacing eq. (3.16) into eq. (3.14), we get

$$\begin{aligned} \mathcal{X} &= (\mathbf{I}_R \times_2^1 \mathcal{I}_{3,R} \times_3^1 \dots \times_Q^1 \mathbf{I}_R) \times_1 \mathbf{P}_1 \times_2 \mathbf{P}_2 \times_3 \dots \times_Q \mathbf{P}_Q, \\ &= (\mathbf{I}_R \times_2^1 \mathcal{I}_{3,R} \times_3^1 \dots \times_Q^1 \mathbf{I}_R) \times_1 \mathbf{P}_1 \times_2 \tilde{\mathbf{P}}_2 \Phi_2 \times_3 \dots \times_Q \mathbf{P}_Q. \end{aligned}$$

Before introducing the ambiguity matrices  $\mathbf{U}_q$ , tensor  $\mathcal{X}$  can then be expressed into a TT format as

$$\begin{aligned} \mathcal{X} &= \underbrace{\mathbf{P}_1}_{\mathbf{A}_1} \times_2^1 \underbrace{(\mathcal{I}_{3,R} \times_2 \tilde{\mathbf{P}}_2 \Phi_2)}_{\mathcal{A}_2} \times_3^1 \dots \times_{Q-2}^1 \underbrace{(\mathcal{I}_{3,R} \times_2 \tilde{\mathbf{P}}_{Q-2} \Phi_{Q-2})}_{\mathcal{A}_{Q-2}} \\ &\quad \times_{Q-1}^1 \underbrace{(\mathcal{I}_{3,R} \times_2 \tilde{\mathbf{P}}_{Q-1} \Phi_{Q-1})}_{\mathcal{A}_{Q-1}} \times_Q^1 \underbrace{\mathbf{P}_Q^T}_{\mathbf{A}_Q}. \end{aligned} \quad (3.17)$$

One may note that for  $2 \leq q \leq Q - 1$ , the considered TT-cores  $\mathbf{A}_1$ ,  $\mathcal{A}_q$  and  $\mathbf{A}_Q$  verify the definition of the TTD with  $\text{rank}\{\mathbf{A}_1\} = \text{rank}\{\mathbf{A}_Q\} = \text{rank}\{\text{unfold}_1 \mathcal{A}_q\} = \text{rank}\{\text{unfold}_3 \mathcal{A}_q\} = R$ , which justify that matrices  $\mathbf{P}_1$  and  $\mathbf{P}_Q$  must be of full column rank. By identifying the TT-cores  $\mathcal{A}_q$  in eq. (3.17), introducing the pre- and post-multiplication ambiguity matrices  $\mathbf{U}_q$  presented in 3.2.2.1, and using the following equivalence

$$\mathcal{G}_q = \mathbf{U}_{q-1} \times_2^1 \mathcal{A}_q \times_3^1 \mathbf{U}_q^{-1} = \mathcal{A}_q \times_1 \mathbf{U}_{q-1} \times_3 \mathbf{U}_q^{-T},$$

theorem 7 is proven.  $\square$

### 3.2.3 Uniqueness of the PARALIND-TTD

One of the most popular condition for the uniqueness of the CPD decomposition is the Kruskal's condition [86] relying on the concept of "Kruskal-rank", or simply krank. The krank of an  $N \times R$  matrix  $\mathbf{P}$ , denoted by  $\text{krank}\{\mathbf{P}\}$ , is the maximum value of  $\ell \in \mathbb{N}$  such that every  $\ell$  columns of  $\mathbf{P}$  are linearly independent. By definition, the krank of a matrix is less than or equal to its rank. Kruskal proved [86] that the condition

$$\text{krank}\{\mathbf{P}_1\} + \text{krank}\{\mathbf{P}_2\} + \text{krank}\{\mathbf{P}_3\} \geq 2R + 2 \quad (3.18)$$

is sufficient for uniqueness of the CPD decomposition in (3.14), with  $Q = 3$ . Furthermore, it becomes a necessary and sufficient condition in the cases  $R = 2$  or 3 (see [8]). Herein, by uniqueness, we understand "essential uniqueness", meaning that if another set of matrices  $\bar{\mathbf{P}}_1$ ,  $\bar{\mathbf{P}}_2$  and  $\bar{\mathbf{P}}_3$  verify (3.18), then there exists a permutation matrix  $\mathbf{\Pi}$  and three invertible diagonal scaling matrices  $(\Delta_1, \Delta_2, \Delta_3)$  satisfying  $\Delta_1 \Delta_2 \Delta_3 = \mathbf{I}_R$ , where  $\mathbf{I}_R$  is the  $R$ -th-order identity matrix, such that

$$\bar{\mathbf{P}}_1 = \mathbf{P}_1 \mathbf{\Pi} \Delta_1, \quad \bar{\mathbf{P}}_2 = \mathbf{P}_2 \mathbf{\Pi} \Delta_2, \quad \bar{\mathbf{P}}_3 = \mathbf{P}_3 \mathbf{\Pi} \Delta_3.$$

The uniqueness condition (3.18) has been generalised to  $Q$ -order CPDs in [131]. It states that the factor matrices  $\mathbf{P}_q$  ( $q = 1, \dots, Q$ ) in (3.14) can be uniquely estimated from  $\mathcal{X}$  if

$$\sum_{q=1}^Q \text{krank}\{\mathbf{P}_q\} \geq 2R + (Q - 1). \quad (3.19)$$

This condition is sufficient but not necessary for the uniqueness of the CPD decomposition.

Based on Kruskal's uniqueness condition as well as on the results derived in [58], we formulate in the following a partial and a full uniqueness condition for the PARALIND-TTD of a  $Q$ -order tensor.

**Theorem 8** (Partial uniqueness of TT-PARALIND). *The factor matrix  $\mathbf{P}_q$  can be uniquely recovered from the estimated TT decomposition of  $\mathcal{X}$  if there exist  $q_1$  and  $q_2$  ( $q_1 \neq q_2 \neq q$ ), such that:*

$$\begin{cases} \text{rank}\{\mathbf{P}_{q_1}\} = \text{rank}\{\mathbf{P}_{q_2}\} = R, \\ \text{rank}\{\mathbf{P}_q\} \geq 2. \end{cases}$$

*Proof.* In the CPD (3.14) the order of the factor matrices is arbitrary and can be changed by a simple index permutation. Thus, in the following we will suppose, without loss of generality, that  $q_1 = 1$  and  $q_2 = Q$ . The fact that  $\text{rank}\{\mathbf{P}_1\} = \text{rank}\{\mathbf{P}_Q\} = R$  implies that the square matrices  $\mathbf{U}_q$  in theorem 7 are all full rank  $R$ . Therefore, the  $\mathcal{G}_q$  tensor can be uniquely recovered from  $\mathcal{X}$  by the TT-SVD algorithm.

According to theorem 7, the tensor  $\mathcal{G}_q$  can be expressed as:

$$\mathcal{G}_q = \mathcal{I}_{3,R} \times_1 \mathbf{U}_{q-1} \times_2 \mathbf{P}_q \times_3 \mathbf{U}_q^{-\top}. \quad (3.20)$$

Following Kruskal's uniqueness condition (3.18), the factor matrices in (3.20) can be recovered from  $\mathcal{G}_q$  if

$$\text{krank}\{\mathbf{U}_{q-1}\} + \text{krank}\{\mathbf{P}_q\} + \text{krank}\{\mathbf{U}_q^{-\top}\} \geq 2R + 2. \quad (3.21)$$

However, in our case we are only interested in recovering  $\mathbf{P}_q$ , which allows to relax Kruskal's condition. It was proven in [58] that the matrix  $\mathbf{P}_q$  can be uniquely estimated from  $\mathcal{G}_q$  if

$$\text{krank}\{\mathbf{U}_{q-1}\} + \text{rank}\{\mathbf{P}_q\} + \text{krank}\{\mathbf{U}_q^{-\top}\} \geq 2R + 2. \quad (3.22)$$

As  $\mathbf{U}_{q-1}$  and  $\mathbf{U}_q$  are full rank square matrices, and  $\text{rank}\{\mathbf{P}_q\} \geq 2$ , (3.21) is verified, which completes the proof.  $\square$

**Theorem 9** (Full TT-PARALIND uniqueness). *The factor matrices  $\mathbf{P}_1, \dots, \mathbf{P}_Q$  can be uniquely recovered from the estimated TT-cores  $\mathbf{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_{Q-1}, \mathbf{G}_Q$  if:*

$$\begin{cases} \text{rank}\{\mathbf{P}_1\} = \text{rank}\{\mathbf{P}_Q\} = R \\ \text{rank}\{\mathbf{P}_q\} \geq 2, \quad 2 < q < Q - 1 \\ \text{krank}\{\mathbf{P}_2\}, \text{krank}\{\mathbf{P}_{Q-1}\} \geq 2. \end{cases}$$

*Proof.* This result is a consequence of theorem 8. The uniqueness of factor matrices  $\mathbf{P}_2, \dots, \mathbf{P}_{Q-1}$  can be proven by repeatedly applying theorem 8 to the different TT-cores  $\mathcal{G}_q$ ,  $2 \leq q \leq Q - 1$ . Meanwhile, condition (3.22) does not guarantee uniqueness of the change-of-basis matrices  $\mathbf{U}_{q-1}$  and  $\mathbf{U}_q$ . In order to guarantee this, Kruskal's condition (3.21) must be verified.

Thus, the condition  $\text{krank}\{\mathbf{P}_2\}, \text{krank}\{\mathbf{P}_{Q-1}\} \geq 2$  implies uniqueness of the CPD decomposition of TT-cores  $\mathcal{G}_2$  and  $\mathcal{G}_{Q-1}$  and consequently, the uniqueness of the  $R \times R$  non-singular matrices  $\mathbf{U}_1$  and  $\mathbf{U}_{Q-1}$ . From theorem 7 we get:

$$\mathbf{P}_1 = \mathbf{G}_1 \mathbf{U}_1 \text{ and } \mathbf{P}_Q = \mathbf{G}_Q^{\top} \mathbf{U}_{Q-1}^{-\top}.$$

Thus, the unique recovery of  $\mathbf{G}_1$  and  $\mathbf{G}_Q$  from  $\mathcal{X}$  along with uniqueness of  $\mathbf{U}_1$  and  $\mathbf{U}_{Q-1}$  implies uniqueness of factor matrices  $\mathbf{P}_1$  and  $\mathbf{P}_Q$ , which completes the proof.  $\square$

### 3.2.4 Discussion

#### 3.2.4.1 More restrictive conditions

Compared to Kruskal's condition (3.19) for  $Q$ -order CPD, the uniqueness condition of theorem 9 is more restrictive. For example, in the case of a fourth-order tensor ( $Q = 4$ ), the condition of theorem 9 implies  $\sum_{q=1}^4 \text{krank}\{\mathbf{P}_q\} \geq 2R + 4$ , while Kruskal's condition requires  $\sum_{q=1}^4 \text{krank}\{\mathbf{P}_q\} \geq 2R + 3$ . This is a direct consequence of imposing simultaneous (partial) uniqueness on all the 3-order TT-cores. More restrictive uniqueness conditions is the price to pay for having a numerically efficient algorithm, that guarantees recovery of the factor matrices for a wide variety of scenarios.

### 3.2.4.2 Estimation scheme architecture

It is worth noting that, from an algorithmic point of view, the estimation of the factor matrices  $\mathbf{P}_q$  can be done either in parallel or sequentially. For a parallel estimation scheme, the conditions of theorem 9 are sufficient. In [156], a sequential scheme was proposed, based on a sequential retrieval of both matrices  $\mathbf{P}_q$  and  $\mathbf{U}_q$ . It requires at each step the knowledge of  $\mathbf{U}_{q-1}$  for decomposing  $\mathcal{G}_q$ . To use a similar sequential scheme for the TT-PARALIND model, it is necessary to also ensure the uniqueness of matrices  $\mathbf{U}_q$ . This can be done by replacing condition  $\text{rank}\{\mathbf{P}_q\} \geq 2$  ( $2 < q < Q - 1$ ) in theorem 9 by a stronger one,  $\text{krank}\{\mathbf{P}_q\} \geq 2$  ( $2 < q < Q - 1$ ).

### 3.2.5 Conclusion and perspectives

The factorisation of a high-order tensor into a collection of low-order tensors, called cores, is an important research topic. Indeed, this family of methods called tensor Networks is an efficient way to mitigate the well-known “curse of dimensionality” problem. In this work, we prove that a  $Q$ -order PARALIND of rank  $R$  can be reformulated as a  $Q - 2$  train of tensors possibly column-deficiency and two full column rank matrices. The condition of partial and full uniqueness are exposed and discussed.

Perspectives of this work are as follows.

1. The condition  $\text{rank}\{\mathbf{P}_1\} = \text{rank}\{\mathbf{P}_Q\}$  in theorem 7 requires the knowledge of the indices of full-rank modes of tensor  $\mathcal{X}$ , which are then arbitrarily fixed to 1 and  $Q$ ; once these two modes are fixed, the order in which the remaining modes are processed is arbitrary. It is certainly possible to obtain a condition involving only one full rank matrix, but in this case the order in which the other modes are processed must be carefully chosen to guarantee the required rank conditions for the TT-SVD algorithm. This aspect is currently under investigation.
2. A very promising application domain of these results is the low-rank approximation of high-dimensional probability mass functions. In this case, these uniqueness results are of utmost importance as the linear dependencies in the model could account for the random variables correlations. A potential application is represented by the flow cytometry data analysis, as shown in [20].



### 3.3 JIRAFE-based Multidimensional Harmonic Retrieval

In this section, we will exploit the proposed JIRAFE scheme in the context of the Multidimensional Harmonic Retrieval. We will show that our methodology [158] has several advantages in terms of flexibility, robustness to noise, computational cost and automatic pairing of the parameters of interest with respect to the tensor order and storage costs.

#### 3.3.1 Context and data model

Multidimensional Harmonic Retrieval (MHR) [66, 126, 14] is a classical signal processing problem that has found several applications in spectroscopy [93], wireless communications [99], sensor array processing [127, 162], to mention a few. The Multidimensional Harmonic (MH) model can be viewed as the tensor-based generalization of the one-dimensional harmonic one, resulting from the sampling process [16] over a multidimensional regular grid. As a consequence, the MH model can be expressed in tensor form as a constrained Canonical Polyadic Decomposition (CPD) [61, 25, 63] with structured Vandermonde factor matrices. Unlike the sum of  $M$  one-dimensional harmonics, which is parametrized by  $M$  angular-frequencies only, the  $P$ -dimensional harmonic model needs the estimation of a large number ( $PM$ ) of paired angular-frequencies of interest. We can easily note that the number of unknown parameters and the order of the associated data tensor grow with  $P$ . For instance, in the problem of dual-polarized MIMO (Multiple-Input Multiple-Output) channel estimation, the data tensor order is five [112]. Moreover, it is likely that the joint exploitation of multi-diversity/modality sensing technologies for data fusion [1, 47, 87] further increases the data tensor order. This trend is usually called the “curse of dimensionality” [28, 147] and the challenge here is to reformulate a high-order tensor as a set of low-order tensors. In this context, we observe an increasing interest for the tensor network theory (see [28] and references therein). Tensor network provides a useful and elegant graphical representation of a high-order tensor into a factor graph where the nodes are low-order tensors, called cores, and the edges encode their dependencies, *i.e.*, their common dimensions, often called “rank”. In addition, tensor network allows to perform scalable/distributed computations over the cores [28]. In the tensor network framework, Hierarchical/tree Tucker [59, 106] and Tensor Train (TT) [142] are two popular representations of a high-order tensor into a graph-connected low-order (at most 3) tensors. In this work, we focus our effort on the TT formalism for its simplicity and compactness in terms of storage cost. Unlike the hierarchical Tucker model, TT is exploited in many practical and important contexts as, for instance, tensor completion [81], blind source separation [12], and machine learning [130], to mention a few. In the context of the MHR problem, this strategy has at least two advantages. First, it is well-known that the convergence of the Alternating Least Squares (ALS) algorithm becomes more and more difficult when the order increases [118, 21, 92]. To deal with this problem, applying ALS on lower-order tensors is preferable. The second argument is to exploit some latent coupling properties between the cores [33, 130] to propose new efficient estimators.

The Maximum Likelihood estimator [84, 34] is the optimal choice from an estimation point of view, since it is statistically efficient, *i.e.*, its Mean Squared Error (MSE) reaches the Cramér-Rao Bound (CRB) in the presence of noise. The main drawback of the maximum likelihood estimator is its prohibitive complexity cost. This limitation is particularly severe in the context of a high-order data tensor. To overcome



this problem, several low-complexity methods can be found in the literature. These methods may not reach, sometimes, the CRB, but they provide a significant gain in terms of the computational cost compared to the maximum likelihood estimator. There are essentially two main families of methods. The first one is based on the factorization of the data to estimate the well-known signal/noise subspace such as the Estimation of Signal Parameters *via* Rotational Invariance Techniques (ESPRIT) [122], the ND-ESPRIT [123], the Improved Multidimensional Folding technique [95], and the CP-VDM [135]. The second one is based on the uniqueness property of the CPD. Indeed, factorizing the data tensor thanks to the ALS algorithm [22] allows a direct identification of the unknown parameters by inspection of the factor matrices. However, the ALS algorithm [21], as well as most of its variants, do not take into account the Vandermonde structure of the factor matrices. Potentially, discarding this physical *a priori* knowledge on the MH model may degrade the estimation performance of the unknown parameters [54]. So, recently, a new family of estimation methods has been introduced in [15]. This approach, called RecALS for Rectified ALS, modifies the ALS method by integrating a rectification/reinforcement strategy on the Vandermonde structure of the factor matrices.

In this work, we propose to use the JIRAFE framework. JIRAFE is composed of two main steps. The first one is the dimensionality reduction of a high-order harmonic model thanks to a Tensor Train decomposition (TTD) [142]. We show how a model equivalence between the CPD and TTD [156] can be exploited to design an efficient optimization strategy exploiting a sum of coupled Least Squares (LS) problems<sup>2</sup>. The second step is dedicated to the estimation of the unknown parameters using a polynomial rooting procedure. This scheme belongs to the RecALS family and can be viewed as an alternative to the one proposed in [15]. Our main contributions can be summarized as follows:

- **Vandermonde based Tensor Train decomposition (VTDD).** The classic MHR problem is reformulated under a Vandermonde based TT format, instead of the usual CPD formulation.
- **Optimization strategy for VTDD.** The structure of the TT-cores is provided and in particular the latent coupling property existing between the TT-cores is explained and exploited in the context of a sequential optimization strategy of a sum of coupled LS problems. This is the first step of JIRAFE, *i.e.*, the dimensionality reduction step.
- **New rectification strategy for Vandermonde factors.** A new rectification strategy for Vandermonde factors is presented and used in the second step of JIRAFE, *i.e.*, the retrieval step.

The MH model assumes that the measurements can be modeled as the superposition of  $M$  undamped exponentials sampled on a  $P$ -dimensional grid according to [66]

$$[\mathcal{X}]_{n_1 \dots n_P} = \sum_{m=1}^M \alpha_m \prod_{p=1}^P z_{m,p}^{n_p-1}, \quad 1 \leq n_p \leq N_p \quad (3.23)$$

in which the  $m$ -th complex amplitude is denoted by  $\alpha_m$  and the pole is defined by  $z_{m,p} = e^{i\omega_{m,p}}$  where  $\omega_{m,p}$  is the  $m$ -th angular-frequency along the  $p$ -th dimension,

<sup>2</sup>Note that the idea of rewriting a CPD into the TT format was briefly mentioned in [142, 33], without discussing the structure of the TT-cores resulting from a decomposition algorithm.

and we have  $\mathbf{z}_p = [z_{1,p} \ z_{2,p} \ \dots \ z_{M,p}]^T$ . Note that the tensor  $\mathcal{X}$  is expressed as the linear combination of  $M$  rank-1 tensors, each of size  $N_1 \times \dots \times N_P$  (the size of the grid), and follows a generalized Vandermonde CPD [109]:

$$\mathcal{X} = \mathcal{A} \times_1 \mathbf{V}_1 \times_2 \dots \times_P \mathbf{V}_P \quad (3.24)$$

where  $\mathcal{A}$  is a  $M \times \dots \times M$  diagonal tensor with  $[\mathcal{A}]_{m,\dots,m} = \alpha_m$  and

$$\mathbf{V}_p = [\mathbf{v}(z_{1,p}) \ \dots \ \mathbf{v}(z_{M,p})]$$

is a  $N_p \times M$  rank- $M$  Vandermonde matrix, where

$$\mathbf{v}(z_{m,p}) = [1 \ z_{m,p} \ z_{m,p}^2 \ \dots \ z_{m,p}^{N_p-1}]^T.$$

We define a noisy MH tensor model of order  $P$  as

$$\mathcal{Y} = \mathcal{X} + \sigma \mathcal{E}$$

where  $\sigma \mathcal{E}$  is the noise tensor,  $\sigma$  is a positive real scalar, and each entry  $[\mathcal{E}]_{n_1 \dots n_P}$  follows an *i.i.d.* circular Gaussian distribution  $\mathcal{CN}(0,1)$ , and  $\mathcal{X}$  has a canonical rank equal to  $M$ .

**Remark.** The addressed MHR problem aims to estimate the PM angular-frequencies correctly paired in terms of “source/dimension” given by the complex argument of the entries of the vectors  $\{\mathbf{z}_1, \dots, \mathbf{z}_P\}$ , from the noisy observation tensor  $\mathcal{Y}$ , when the order of  $\mathcal{X}$  is strictly greater than three.

It is important to stress that the paring and estimation operations have to be jointly performed. Indeed, searching the tensor generated from a given scheduling of PM parameters of interest that are the closest to the observed tensor suffers from the well-known combinatorial explosion, especially for large values of  $P$ .

### 3.3.2 Dimensionality reduction based on a train of low-order tensors

#### 3.3.2.1 Vandermonde based TT decomposition

Due to the simple graph-based formalism of the TT, it is straightforward to rewrite the diagonal tensor  $\mathcal{A}$  introduced in eq. (3.24) as the following TTD

$$\mathcal{A} = \mathbf{A} \times_2^1 \mathcal{I}_{3,M} \times_3^1 \dots \times_{P-1}^1 \mathcal{I}_{3,M} \times_P^1 \mathbf{I}_M$$

where  $\mathbf{A}$  is a  $M \times M$  diagonal matrix with  $[\mathbf{A}]_{m,m} = \alpha_m$ . A graph-based visualization is given on Fig. 3.7.

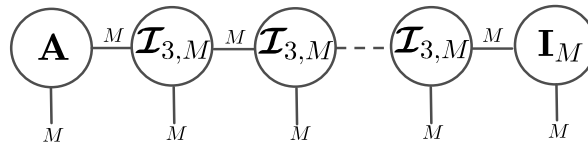


FIGURE 3.7: A possible TTD of tensor  $\mathcal{A}$ .

Plugging the TTD of tensor  $\mathcal{A}$  into eq. (3.24), leads to

$$\begin{aligned}\mathcal{X} &= \mathcal{A} \times_1 \mathbf{V}_1 \times_2 \dots \times_P \mathbf{V}_P \\ &= \left( \mathbf{A} \times_2^1 \mathcal{I}_{3,M} \times_3^1 \dots \times_{P-1}^1 \mathcal{I}_{3,M} \times_P^1 \mathbf{I}_M \right) \times_1 \mathbf{V}_1 \times_2 \dots \times_P \mathbf{V}_P \\ &= (\mathbf{V}_1 \mathbf{A}) \times_2^1 \mathcal{V}_2 \times_3^1 \dots \times_{P-1}^1 \mathcal{V}_{P-1} \times_P^1 \mathbf{V}_P^T\end{aligned}\quad (3.25)$$

where  $\mathcal{V}_p = \mathcal{I}_{3,M} \times_2 \mathbf{V}_p$  is of size  $M \times N_p \times M$ . A representation of  $\mathcal{V}_p$  and  $\mathcal{X}$  are given on Figs 3.8 and 3.9, respectively.

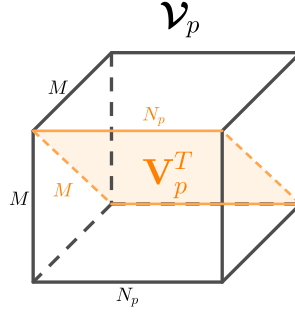


FIGURE 3.8: Representation of the 3-order  $M \times N_p \times M$  tensor  $\mathcal{V}_p$ .

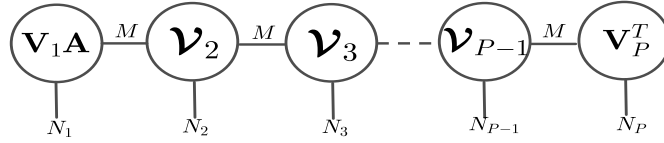


FIGURE 3.9: VTTD of tensor  $\mathcal{X}$  corresponding to eq. (3.25).

According to eq. (3.25), the generalized Vandermonde CPD of tensor  $\mathcal{X}$  is equivalent to a train of  $(P - 2)$  3-order tensors that follow a constrained rank- $M$  CPD with a Vandermonde factor on its 2nd mode.

In the following theorem, we give the retrieved TT-cores structure when applying the TT-SVD algorithm to the multidimensional harmonic model in eq. (3.24).

**Theorem 10.** *Applying the TT-SVD algorithm to eq. (3.25) with full column rank factors, we have the following result:*

$$\begin{aligned}\mathbf{G}_1 &= \mathbf{V}_1 \mathbf{A} \mathbf{M}_1^{-1} \\ \mathcal{G}_p &= \mathcal{I}_{3,M} \times_1 \mathbf{M}_{p-1} \times_2 \mathbf{V}_p \times_3 \mathbf{M}_p^{-T}, \quad \text{where } 2 \leq p \leq P-1, \\ \mathbf{G}_P &= \mathbf{M}_{P-1} \mathbf{V}_P^T\end{aligned}\quad (3.26)$$

where  $\mathbf{M}_p \in \mathbb{C}^{M \times M}$  is a nonsingular transformation matrix.

*Proof.* Here, we prove the result given in the preliminary article [156]. Our methodology is based on a constructive proof. The aim is to apply the TT-SVD algorithm to the model of interest presented in eq. (3.23) and to provide the algebraic structure of the TT-cores resulting from the decomposition. Let  $\mathcal{X}$  be a  $P$ -order rank- $M$  constrained CPD tensor following eq. (3.23) of size  $N_1 \times \dots \times N_P$  with full column rank factors. Applying the TT-SVD algorithm to  $\mathcal{X}$  consists of applying sequentially the

SVD to extract the dominant subspaces at each step. In the following, we give the expression of the matrix unfoldings and the SVD factors at each step.

- The first unfolding  $\mathbf{X}_{(1)}$  of size  $N_1 \times (N_2 \cdots N_P)$  is given by:

$$\begin{aligned}\mathbf{X}_{(1)} &= \text{reshape}(\mathcal{X}; N_1, \prod_{s=2}^P N_s) \\ &\stackrel{\Delta}{=} \mathbf{V}_1 \mathbf{A} (\mathbf{V}_P \odot \mathbf{V}_{P-1} \odot \cdots \odot \mathbf{V}_2)^T \\ &\stackrel{\text{SVD}}{=} \mathbf{U}^{(1)} \mathbf{V}^{(1)}\end{aligned}$$

where  $\mathbf{U}^{(1)}$  and  $\mathbf{V}^{(1)}$  are the left and right singular vectors matrices, respectively. We recall that the diagonal singular values matrix is absorbed in  $\mathbf{V}^{(1)}$ . These matrices can be expressed as

$$\mathbf{U}^{(1)} \mathbf{M}_1 = \mathbf{V}_1 \mathbf{A}, \quad (3.27)$$

$$\mathbf{V}^{(1)} = \mathbf{M}_1 (\mathbf{V}_P \odot \mathbf{V}_{P-1} \odot \cdots \odot \mathbf{V}_2)^T \quad (3.28)$$

where  $\mathbf{M}_1$  is a  $M \times M$  transformation matrix and  $\mathbf{A}$  is defined in Section 3.3.2.1. Note that in terms of rank we have the equality

$$\text{rank} \mathbf{X}_{(1)} = \text{rank}(\mathbf{V}_1 \mathbf{A}) = \text{rank} \mathbf{G}_1 = M.$$

It is worth noting that the Khatri-Rao product of matrices  $\mathbf{V}_p$  does not decrease the rank [9], *i.e.*, the rank of the Khatri-Rao product matrix also equals  $M$ . From eq. (3.27), the expression of the first TT-core is

$$\mathbf{G}_1 = \mathbf{U}^{(1)} = \mathbf{V}_1 \mathbf{A} \mathbf{M}_1^{-1}. \quad (3.29)$$

- From eq. (3.28), reshaping matrix  $\mathbf{V}^{(1)}$  provides

$$\begin{aligned}\mathbf{V}_{(2)}^{(1)} &= \text{reshape}(\mathbf{V}^{(1)}; M N_2, \prod_{s=3}^P N_s) \\ &\stackrel{\Delta}{=} (\mathbf{V}_2 \odot \mathbf{M}_1) (\mathbf{V}_P \odot \mathbf{V}_{P-1} \odot \cdots \odot \mathbf{V}_3)^T \\ &\stackrel{\text{SVD}}{=} \mathbf{U}^{(2)} \mathbf{V}^{(2)}\end{aligned}$$

where  $\mathbf{U}^{(2)}$  and  $\mathbf{V}^{(2)}$  are of rank  $M$ , and can be expressed as

$$\begin{aligned}\mathbf{U}^{(2)} \mathbf{M}_2 &= \mathbf{V}_2 \odot \mathbf{M}_1, \\ \mathbf{V}^{(2)} &= \mathbf{M}_2 (\mathbf{V}_P \odot \mathbf{V}_{P-1} \odot \cdots \odot \mathbf{V}_3)^T.\end{aligned}$$

The second TT-core can then be expressed as:

$$\begin{aligned}\mathcal{G}_2 &= \text{reshape}(\mathbf{U}^{(2)}; M, N_2, M) \\ &= \mathcal{I}_{3,M} \times_1 \mathbf{M}_1 \times_2 \mathbf{V}_2 \times_3 \mathbf{M}_2^{-T}.\end{aligned} \quad (3.30)$$

- Following the same reasoning, the expression of  $\mathbf{V}_{(2)}^{(p-1)}$  at the  $p$ -th step is

$$\begin{aligned}\mathbf{V}_{(2)}^{(p-1)} &= \text{reshape}(\mathbf{V}^{(p-1)}; MN_p, \prod_{s=3}^P N_s) \\ &\stackrel{\Delta}{=} (\mathbf{V}_p \odot \mathbf{M}_{p-1})(\mathbf{V}_P \odot \mathbf{V}_{P-1} \odot \cdots \odot \mathbf{V}_{p+1})^T \\ &\stackrel{\text{SVD}}{=} \mathbf{U}^{(p)} \mathbf{V}^{(p)}\end{aligned}$$

where

$$\begin{aligned}\mathbf{U}^{(p)} \mathbf{M}_p &= \mathbf{V}_p \odot \mathbf{M}_{p-1}, \\ \mathbf{V}^{(p)} &= \mathbf{M}_p (\mathbf{V}_P \odot \mathbf{V}_{P-1} \odot \cdots \odot \mathbf{V}_{p+1})^T,\end{aligned}\tag{3.31}$$

where  $\mathbf{M}_p$  is a  $M \times M$  transformation matrix and the TT-core expression is given by

$$\begin{aligned}\mathcal{G}_p &= \text{reshape}(\mathbf{U}^{(p)}; M, N_p, M) \\ &= \mathcal{I}_{3,M} \times_1 \mathbf{M}_{p-1} \times_2 \mathbf{V}_p \times_3 \mathbf{M}_p^{-T}.\end{aligned}\tag{3.32}$$

- At the last step, and from eq. (3.31), the SVD factor  $\mathbf{V}^{(P-1)}$  and the last TT-core expressions are given by

$$\mathbf{G}_P = \mathbf{V}^{(P-1)} = \mathbf{M}_{P-1} \mathbf{V}_P^T.\tag{3.33}$$

Theorem 10 can be proved using (3.29), (3.30), (3.32) and (3.33).  $\square$

The meaning of the above result is that applying the TT-SVD algorithm to eq. (3.24) generates 3-order TT-cores that follow a CPD, with TT-ranks all equal to the canonical rank  $M$  (see Fig. 3.10). Furthermore, each CPD-train core  $\mathcal{G}_p$  has a common coupled factor with the two consecutive TT-cores  $\mathcal{G}_{p-1}$  and  $\mathcal{G}_{p+1}$ . Theorem 10 shows that applying the TT-SVD algorithm to eq. (3.24) allows to retrieve the exact TT-cores defined in eq. (3.25), up to transformation matrices  $\mathbf{M}_p$ . These matrices  $\mathbf{M}_p$  are *change-of-basis* matrices that are related to the estimation of the dominant subspace using the SVD and TT-SVD, as shown in the previous proof.

**Remark.** One should be cautious about reducing the above theorem to the well-known TT model ambiguities. Indeed, each entry of a tensor following a TT model is the product of  $P$  matrices, each of which obtained as a reshaping of a TT-core [142]. In fact, the transformation matrices involved in the TT-SVD algorithm coincide with the TT model ambiguities only if the factors are full column rank<sup>3</sup>. Finally, the above theorem shows that the TT-SVD involves latent but crucial information in the transformation matrices that must be estimated.

**Remark.** Under mild conditions, it is well-known that the factors of a  $P$ -order CPD are unique up to trivial ambiguities (common column permutation and scaling [136]). In [156], it is proved that the factors based on eq. (3.26) can also be estimated with the same trivial ambiguities.

<sup>3</sup>For a full row rank factor as for instance in wireless communications [112], the TT-cores computed by the TT-SVD algorithm have less intuitive and more complicated expressions. This is the subject of Section 4.2.

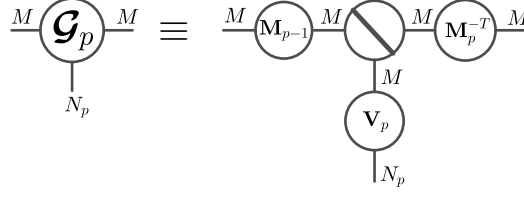


FIGURE 3.10: 3-order CPD of the  $p$ -th TT-core. Matrices  $\mathbf{M}_{p-1}$  and  $\mathbf{M}_p$  are latent quantities.

### 3.3.3 Factor retrieval scheme

#### 3.3.3.1 JIRAFE with Vandermonde-based rectification

The proposed estimator is based on the JIRAFE principle. JIRAFE, meaning Joint dimensionality Reduction And Factors rEtieval, is composed of two main steps.

1. The first one is the computation of the TTD of the initial tensor. By doing this, the initial  $P$ -order tensor is broken down into  $P$  graph-connected third-order tensors, called TT-cores as represented by Fig. 3.9. This dimensionality reduction is an efficient way to mitigate the “curse of dimensionality”. To reach this goal, the TT-SVD [142] presented above is used as a first step.
2. The second step is dedicated to the factorization of the TT-cores. Recall the main result given by Theorem 10, *i.e.*, if the initial tensor follows a  $P$ -order CPD of rank  $M$ , then the TT-cores for  $2 \leq p \leq P-1$  follow coupled 3-order CPD of rank  $M$ . Consequently, the JIRAFE minimizes the following criterion over the physical quantities  $\{\mathbf{V}_1, \dots, \mathbf{V}_P\}$  and over the latent quantities  $\{\mathbf{M}_1, \dots, \mathbf{M}_{P-1}\}$ :

$$\begin{aligned} \mathcal{C} = & \|\mathbf{G}_1 - \mathbf{V}_1 \mathbf{A} \mathbf{M}_1^{-1}\|_F^2 + \|\mathbf{G}_P - \mathbf{M}_{P-1} \mathbf{V}_P^T\|_F^2 \\ & + \sum_{p=2}^{P-1} \|\mathbf{G}_p - \mathcal{I}_{3,M} \times_1 \mathbf{M}_{p-1} \times_2 \mathbf{V}_p \times_3 \mathbf{M}_p^{-T}\|_F^2. \end{aligned} \quad (3.34)$$

The above cost function is the sum of coupled LS criterions. The aim of JIRAFE is to recover the original tensor factors using only the 3-order tensors  $\mathbf{G}_p$  based on the result given in eq. (3.26). eq. (3.34) is expressed as a sum of dependent positive terms due to the coupling properties existing through the matrices  $\{\mathbf{M}_1, \dots, \mathbf{M}_{P-1}\}$  as demonstrated in Theorem 10. Note that the coupling properties between tensors are usually due to physical constraints [2, 1]. In our case, these properties are a result of the sequential structure of the TT-SVD algorithm. Matrices  $\{\mathbf{M}_1, \dots, \mathbf{M}_{P-1}\}$  can be considered as latent matrices, *i.e.*, they do not have a physical meaning, but are essential from an estimation point of view. These remarkable coupling properties take place between a given TT-core and the two other TT-cores connected to it in the graph-based representation (see Fig. 3.10). Minimizing independently all the positive terms in eq. (3.34) is a simple procedure but this also means that the structure of the problem of interest is completely eluded. On the other hand, finding jointly  $\{\mathbf{V}_1, \dots, \mathbf{V}_P\}$  and  $\{\mathbf{M}_1, \dots, \mathbf{M}_{P-1}\}$  is not trivial and highly time consuming [35, 1]. Consequently, the JIRAFE approach adopts a straightforward sequential methodology, described in Fig. 3.11, to minimize the cost function  $\mathcal{C}$ . Any 3-order algorithm existing in the literature dedicated to the computation of a

3-order CPD can be exploited as for instance the popular ALS algorithm [22]. However, in the context of the MHR problem, the RecALS method introduced in [15] is used and extended. The idea of the RecALS is to associate the ALS algorithm with a Vandermonde rectification strategy. In [15], a link between a Vandermonde vector and the rank-1 factorization of a Toeplitz matrix is proposed. As a cheaper alternative, the Shift Invariant Property (SIP) is proposed and described in the next section.

Based on the new Vandermonde constrained Tensor Train modelisation in eq. (3.25), and the CPD structure of the TT-cores in Theorem 10, the idea of the proposed scheme is to replace the estimation of the high  $P$ -order tensor by a sequential estimation procedure that operates on 3-order tensors only. We recall that the RecALS algorithm is an ALS-based solution that is efficient for 3-order tensors but becomes a delicate estimator for high-order tensors. In addition, the ALS-based techniques may require several iterations to converge, and convergence is increasingly difficult when the order of the tensor increases, and it is not even guaranteed. In the proposed solution, the use of iterative algorithms, such as ALS, becomes easier after applying a dimensionality reduction to the original tensor using a CPD-train model, since they are applied to (smaller) 3-order tensors. The new proposed solution is called VTT-RecALS algorithm, and its pseudo-code is presented in Algorithm 7. The VTT-

---

**Algorithm 7** VTT-RecALS
 

---

**Input:**  $\mathcal{Y}, M, \text{CritStop}$

**Output:** Estimated parameters:  $\{\mathbf{z}_1, \dots, \mathbf{z}_P\}$ .

1: Dimensionality reduction:

$$[\mathbf{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_{P-1}, \mathbf{G}_P] = \text{TT-SVD}(\mathcal{X}, M).$$

2: Factor retrieval:

3: For  $p = 2$ ,

$$[\hat{\mathbf{M}}_1, \hat{\mathbf{V}}_2, \hat{\mathbf{M}}_2, \mathbf{z}_2] = \text{RecALS}_3(\mathcal{G}_2, M, \text{CritStop}).$$

4: **for**  $p = 3 \dots P - 1$  **do**

5:    $[\hat{\mathbf{V}}_p, \hat{\mathbf{M}}_p, \mathbf{z}_p] = \text{RecALS}_2(\mathcal{G}_p, \hat{\mathbf{M}}_{p-1}, M, \text{CritStop})$

6: **end for**

7:  $\hat{\mathbf{V}}_1 \hat{\mathbf{A}} = \mathbf{G}_1 \hat{\mathbf{M}}_1$ , and  $\hat{\mathbf{V}}_p = \mathbf{G}_p^T \hat{\mathbf{M}}_{p-1}^{-T}$

---

RecALS algorithm is actually divided into two parts. The first part is dedicated to dimensionality reduction, *i.e.*, breaking the dimensionality of the high  $P$ -order tensor into a set of 3-order tensors using Theorem 10. The second part is dedicated to the factors retrieval from the TT-cores using the RecALS algorithm presented in the next section. It is worth noting that the factors  $\hat{\mathbf{V}}_p$  are estimated up to a trivial (common) permutation ambiguity [136]. As noted in [156], since all the factors are estimated up to a unique column permutation matrix, the estimated angular-frequencies are automatically paired.

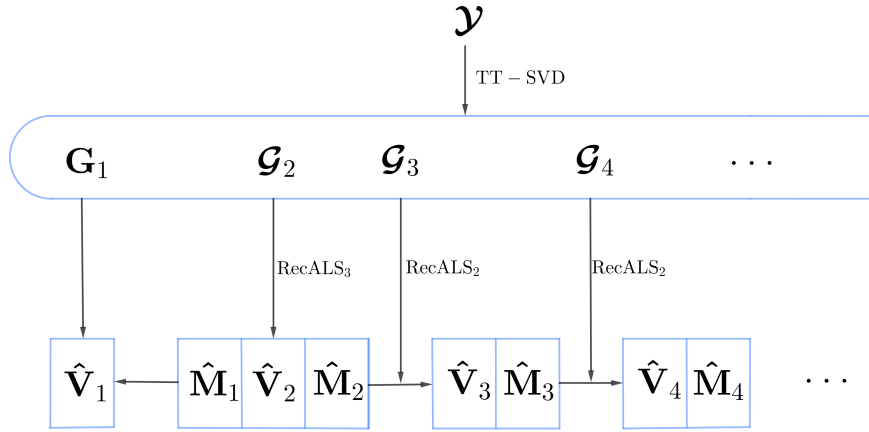


FIGURE 3.11: VTT-RecALS representation

### 3.3.3.2 Shift Invariance Principle (SIP)

In this section, we propose a new rectification strategy for Vandermonde factors, which is an alternative to Toeplitz Rank-1 Approximation (TR<sub>1</sub>A) proposed in [15]. The rectification strategy is called shift invariance principle (SIP), which is inspired from the notion of pencil of matrices (see [64] for instance). It is a rectification strategy that copes with the multidimensionality of the MH model, and is integrated into the VTT-RecALS algorithm to rectify the Vandermonde factors.

**3.3.3.2.1 The SIP criterion** Note that in a noiseless scenario, each Vandermonde factor matrix  $\mathbf{V}_p$  in (3.24) satisfies the following equality [6, 108]

$$\bar{\mathbf{V}}_p = \underline{\mathbf{V}}_p \text{diag}(\mathbf{z}_p),$$

where  $\mathbf{V}_p$  is the  $p$ -th factor in eq. (3.24). Let us now consider the following cost function:

$$\min_{\mathbf{z}_p} \mathcal{C}(\mathbf{z}_p) \quad \text{where} \quad \mathcal{C}(\mathbf{z}_p) = \|\bar{\mathbf{V}}_p - \underline{\mathbf{V}}_p \text{diag}(\mathbf{z}_p)\|_F^2 = \sum_{m=1}^M P(z_{m,p}) \quad (3.35)$$

in which  $P(z_{m,p}) = \|\bar{\mathbf{v}}(z_{m,p}) - \underline{\mathbf{v}}(z_{m,p})z_{m,p}\|^2$ . Minimizing eq. (3.35) with respect to  $\mathbf{z}_p$  is equivalent to minimizing each positive term  $P(z_{m,p})$  in the sum. In addition,  $P(z_{m,p})$  is not a function of  $z_{m',p}$  for  $m' \neq m$ . Thus, minimizing eq. (3.35) with respect to  $\mathbf{z}_p$  is equivalent to solve  $M$  independent problems of the following form:

$$\min_{z_{m,p}} P(z_{m,p}) \quad \text{where} \quad P(z_{m,p}) = -z_{m,p}^* \cdot Q(z_{m,p})$$

and

$$Q(z_{m,p}) = a_{m,p} z_{m,p}^2 - b_{m,p} z_{m,p} + a_{m,p}^*$$



in which  $a_{m,p} = \bar{\mathbf{v}}(z_{m,p})^H \underline{\mathbf{v}}(z_{m,p})$ ,  $b_{m,p} = \|\bar{\mathbf{v}}(z_{m,p})\|^2 + \|\underline{\mathbf{v}}(z_{m,p})\|^2$ , and  $Q(z_{m,p})$  is a second degree polynomial where the argument of the two roots is

$$\hat{\omega}_{m,p} = \angle \hat{z}_{m,p} = \angle \left( \frac{b_{m,p} \pm \sqrt{b_{m,p}^2 - 4|a_{m,p}|^2}}{2a_{m,p}} \right) = \angle \left( \frac{1}{a_{m,p}} \right), \quad (3.36)$$

where  $\hat{\omega}_{m,p}$  is the estimate of the  $m$ -th angular-frequency along the  $p$ -th dimension. The result in eq. (3.36) is integrated in the RecALS<sub>3</sub> algorithm. We denote by RecALS<sub>3</sub>, the RecALS applied to a 3-order tensor, while RecALS<sub>2</sub> denotes the RecALS applied to a 3-order tensor using the knowledge of one factor. The RecALS<sub>3</sub> algorithm used in Algorithm 7 is summarized in Algorithm 8. RecALS<sub>2</sub> has a similar algorithmic description as Algorithm 8, removing step 3, since  $\mathbf{M}_{p-1}$  becomes an a priori known input. These algorithms are applied to the resultant 3-order TT-cores to recover the 3 factors with a Vandermonde 2nd mode factor.

---

**Algorithm 8** Rectified Tri-ALS (RecALS<sub>3</sub>)

---

**Input:**  $\mathcal{G}_p, M, \text{CritStop}$

**Output:** Estimated parameters:  $\{\omega_{1,p}, \dots, \omega_{M,p}\}$ .

```

1: Initialize:  $\hat{\mathbf{V}}_p, \hat{\mathbf{M}}_p^{-T}$ 
2: while CritStop is false do
3:    $\hat{\mathbf{M}}_{p-1} = \text{unfold}_1 \mathcal{G}_p \cdot ((\hat{\mathbf{M}}_p^{-T} \odot \hat{\mathbf{V}}_p)^T)^\dagger$ 


---


4:    $\hat{\mathbf{V}}_p = \text{unfold}_2 \mathcal{G}_p \cdot ((\hat{\mathbf{M}}_p^{-T} \odot \hat{\mathbf{M}}_{p-1})^T)^\dagger$ 
5:   for  $m = 1 \dots M$  do
6:      $a_{m,p} = \bar{\mathbf{v}}(z_{m,p})^H \underline{\mathbf{v}}(z_{m,p})$ 
7:      $\omega_{m,p} = \angle \left( \frac{1}{a_{m,p}} \right)$ 
8:      $z_{m,p} = e^{i\omega_{m,p}}$ 
9:   end for
10:   $\hat{\mathbf{V}}_p := [\mathbf{v}(z_{1,p}) \dots \mathbf{v}(z_{M,p})]$ 


---


11:   $\hat{\mathbf{M}}_p^{-T} = \text{unfold}_3 \mathcal{G}_p \cdot ((\hat{\mathbf{V}}_p \odot \hat{\mathbf{M}}_{p-1})^T)^\dagger$ 
12: end while

```

---

**3.3.3.2.2 Comparison with other RecALS scheme** It is worth noting that, for each parameter  $\omega_{m,p}$ , the TR<sub>1</sub>A method, proposed in [15], for a  $P$ -order rank- $M$  tensor of size  $N \times \dots \times N$ , is based on (i) the rank-1 diagonalization to obtain the dominant eigen-vector  $\mathbf{u}$  of a rank-1  $N \times N$  Toeplitz matrix and (ii) the computation of the angle of the product  $[\mathbf{u}]_1 [\mathbf{u}]_2^*$ . This cost is evaluated to  $O(N + 1)$ . So, for the entire set of parameters of interest, the final computation cost for the TR<sub>1</sub>A method is evaluated to  $O(N \cdot P \cdot M + P \cdot M)$ . In the SIP methodology, only  $MP$  inner products have to be computed. Each inner product implies  $N - 1$  sums and multiplications, thus the complexity is evaluated to  $O(N - 1)$ . The overall cost of the SIP method is thus  $O(N \cdot P \cdot M - P \cdot M)$ . We can see that the additional term  $O(PM)$  is involved in the TR<sub>1</sub>A method with respect to the SIP one. This quantity may be large for high values of  $P$ .

### 3.3.4 Simulation results

This section is organized as follows.

- In section 3.3.4.1, the interest of the VTT approach is illustrated and studied.
  - (a) Paragraph 3.3.4.1.1 compares the VTT-RecALS-SIP algorithm with the CPD-RecALS-SIP algorithm to evaluate the interest of using the VTT instead of the CPD.
  - (b) Paragraph 3.3.4.1.2 studies the impact of the parameters  $P$  and  $N$  on the estimation accuracy of the VTT-RecALS-SIP algorithm.
- Section 3.3.4.2 is dedicated to the comparison of the VTT-RecALS-SIP with the state-of-art estimators.
  - (a) In terms of robustness to noise in paragraph 3.3.4.2.1.
  - (b) In terms of computational time in paragraph 3.3.4.2.2.

It is worth noting that the study of the impact of the parameters dimension, rank and order, is important, since they can be related to physical quantities in realistic applications. For instance, in the problem of dual-polarized MIMO channel estimation [112], the dimension, rank and order represent respectively, the number of sensors/antennas at the transmission and reception, the number of dominant paths and the number of spatial diversities at the reception and transmission.

The simulations were performed on a PC equipped with Matlab2016b, an i7, 2.10GHz processor and 8Gb RAM.

Note that the  $N \times M$  Vandermonde factors are generated based on a single realization of  $\omega_{m,p}$  following a uniform distribution in  $]0, \pi[$ . Let  $f(\hat{\mathcal{X}}^{(t)}) = \|\mathcal{X} - \hat{\mathcal{X}}^{(t)}\|_F$ , where  $\hat{\mathcal{X}}^{(t)}$  denotes the estimated tensor at the  $t$ -th iteration. The convergence test, noted as CritStop, for RecALS algorithms is chosen such that  $\frac{|f(\hat{\mathcal{X}}^{(t)}) - f(\hat{\mathcal{X}}^{(t+1)})|}{f(\hat{\mathcal{X}}^{(t)})} < \epsilon$ , or when the number of iterations exceeds 1000. The signal to noise ratio (SNR) is defined as

$$\text{SNR [dB]} = 10 \log \left( \frac{\|\mathcal{X}\|_F^2}{\|\sigma \mathcal{E}\|_F^2} \right),$$

with  $\mathcal{E}$  drawn from a complex circular *i.i.d.* Gaussian distribution with zero mean and unit variance. The plotted CRB is the one calculated in [14]. The rank  $M$  is supposed to be perfectly known in the simulations.

The MSE is defined as  $\text{MSE} = \frac{1}{MC} \sum_{k=1}^{MC} \sum_{p=1}^P \sum_{m=1}^M (\omega_{m,p} - \hat{\omega}_{m,p}^{(k)})^2$ , where  $\hat{\omega}_{m,p}^{(k)}$  is the estimation of  $\omega_{m,p}$  at the  $k$ -th run and  $MC$  is the number of Monte-Carlo runs. The depicted MSE is the error over the angular frequencies, and is obtained by averaging the results over 1000 independent Monte Carlo runs, truncated from 5% worst and 5% best MSEs to eliminate ill-convergence experiments and outliers.

#### 3.3.4.1 Advantages of the TT approach

**3.3.4.1.1 VTT over CPD** In this section, we show the interest of using the VTT model over the CPD in terms of noise robustness. The aim of this section is to compare VTT and CPD, through the comparison of VTT-RecALS-SIP and CPD-RecALS-SIP, using the proposed solution SIP presented in Section 3.3.3.2.

In Fig. 3.12, the algorithms VTT-RecALS-SIP and CPD-RecALS-SIP are applied to

a 6-order tensor. The rank is fixed at  $M = 3$  and the dimension  $N = 6$ . We can remark that both methods are efficient for a wide range of SNR, with a better robustness for the VTT-RecALS-SIP for low SNR. Similar behavior was found for a rank  $M = 2$ . We also noticed that for  $N = 8$ ,  $M = 2$  and  $P = 6$ , *i.e.*, for a lower number

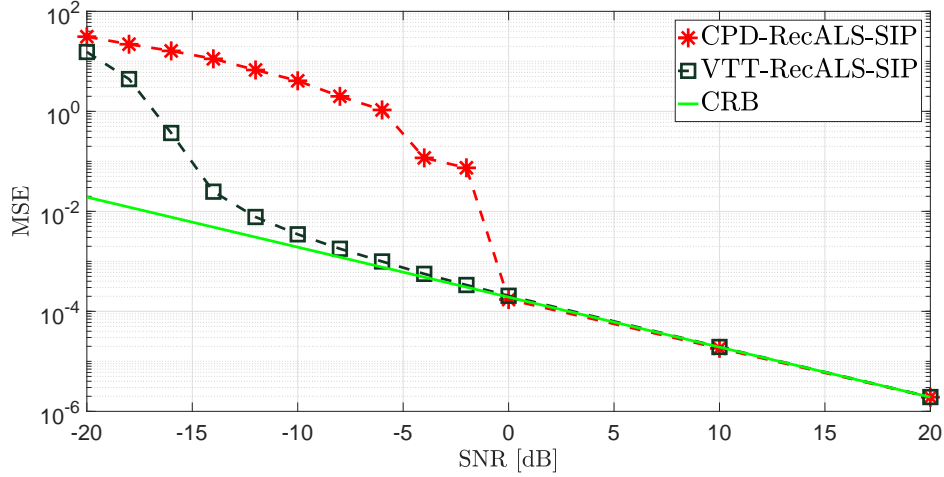


FIGURE 3.12: MSE vs SNR in dB for  $P = 6$  with  $M = 3$ ,  $N = 6$ .

( $PM$ ) of parameters to be estimated, and a higher number of samples compared to the last experiment, the MSE of the VTT-RecALS-SIP gets closer to the CRB for low SNR, meanwhile the CPD-RecALS-SIP keeps the same behavior. It is worth noting that the effectiveness of VTT-RecALS-SIP over the CPD-RecALS-SIP in the low SNR range can be justified by the noise reduction property of the truncated SVD when the TT-SVD is applied.

Note that the well-known threshold effect in the MSE curves indicates the limit SNR between the two regimes, *i.e.*, when the estimator fails and succeeds to estimate the parameters of interest [116, 76, 27]. So, this is a key quantity to assess the quality of an estimator in a practical context (see for instance [141, 117]). Thanks to the dimensionality reduction step performed here with VTT, this SNR threshold is improved at least by 10 dB compared to the CPD solution in Fig. 3.12.

**3.3.4.1.2 Impact of parameters  $P$  and  $N$  on the VTT-RecALS-SIP** The purpose of next simulations to evaluate the impact of parameters  $P$  and  $N$  on the behavior of the VTT-RecALS-SIP algorithm. First, in Fig. 3.13, we fix  $P = 6$ ,  $M = 2$ ,  $\text{SNR} = 5\text{dB}$  and vary the dimension  $N$ . Note that the MSE continuously decreases when the dimension is increased, which is predictable, since the number ( $MP$ ) of parameters is fixed and the number  $N^P$  of samples grows with dimension  $N$ . In Fig. 3.14, we fix  $N = 6$ ,  $M = 2$ ,  $\text{SNR} = 5\text{dB}$  and vary the tensor order  $P$ . Here, both, the number ( $MP$ ) of parameters and  $N^P$  of samples grow, but since this latter grows faster with  $P$ , the MSE of VTT-RecALS-SIP linearly decreases, which shows that the proposed method becomes more efficient when the order of the MH tensor increases.

### 3.3.4.2 VTT-RecALS-SIP versus the state-of-art estimators

After showing the interest of the VTT-RecALS-SIP over the CPD-RecALS-SIP, in this section, we compare the proposed VTT-RecALS-SIP algorithm with different state-of-art schemes such as the CPD-RecALS-TR<sub>1</sub>A [15], the ND-ESPRIT [123] and the

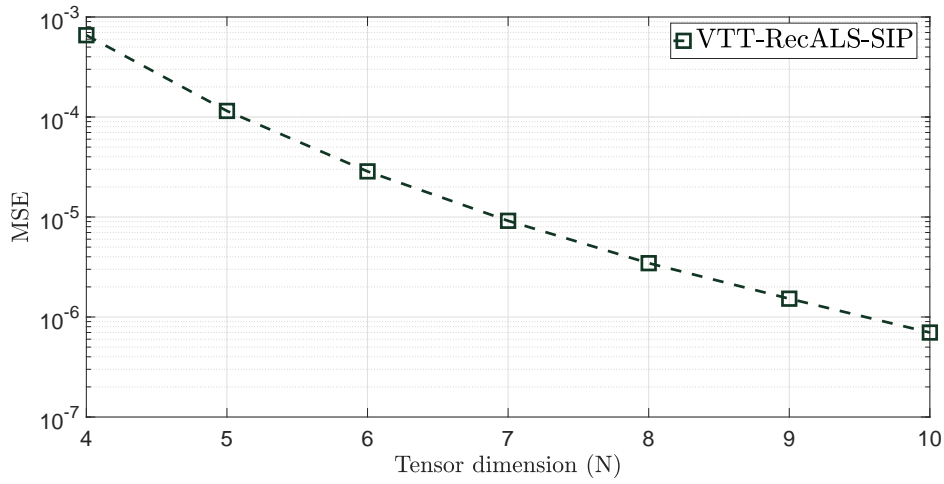


FIGURE 3.13: MSE vs tensor dimension for  $P = 6$  with  $M = 2$ , SNR = 5dB (impact of  $N$ ).

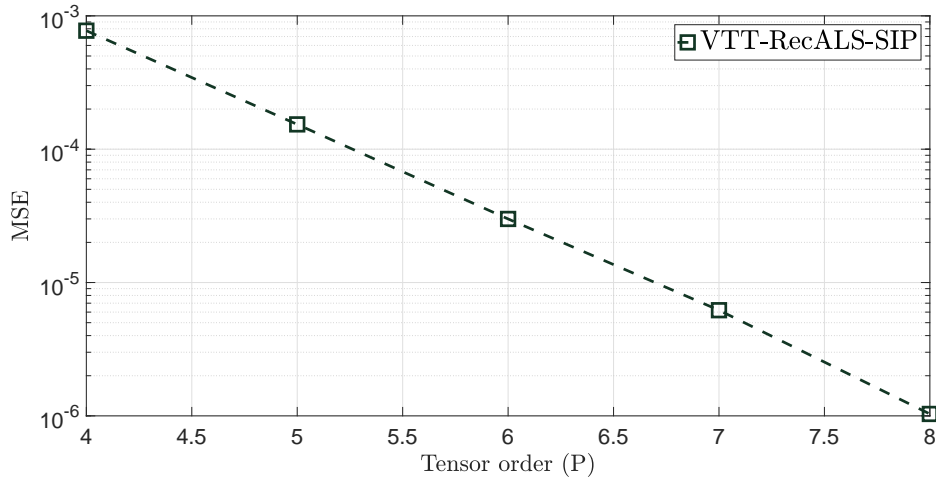


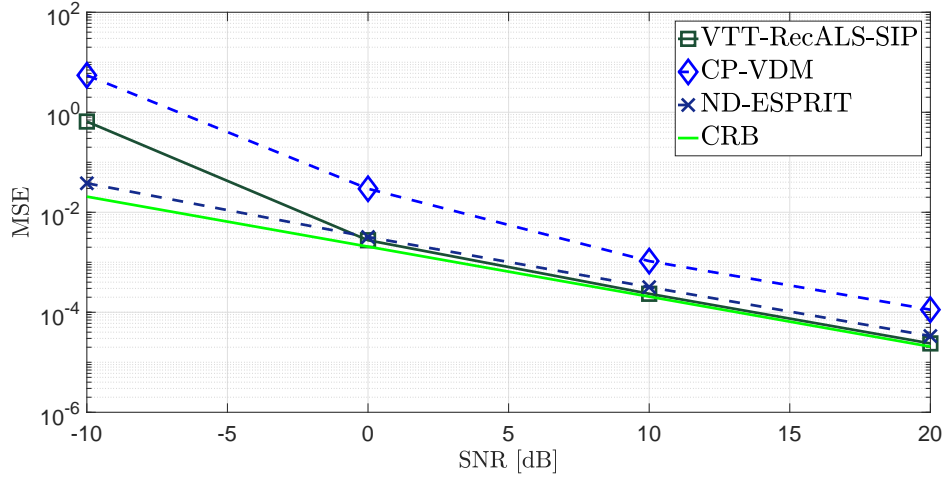
FIGURE 3.14: MSE vs tensor order for  $N = 6$  with  $M = 2$ , SNR = 5dB (impact of  $P$ ).

CP-VDM [84]. Like for the first experiments, different values of parameters are considered. In Tab. 3.5, we give the chosen values in each figure. Note that only one parameter is changed from an experiment to another.

**3.3.4.2.1 Robustness to noise** In Fig. 3.15, we fix  $P = 4$ ,  $M = 2$  and  $N = 6$ . We can remark that the VTT-RecALS-SIP has the MSE closest to the CRB for positive SNR, keeping in mind that  $P = 4$  is not a very high order, which is thus not the most interesting case. Increasing the rank  $M = 3$ , in Fig. 3.16 compared to Fig. 3.15, and for a relatively “small” order  $P = 4$ , gives a comparable behavior between the VTT-RecALS-SIP and the ND-ESPRIT algorithms, the difference of the computational time of both methods is evaluated in the next section. Meanwhile the gap with CP-VDM becomes more pronounced. More interesting cases with a higher order  $P$  are considered in the next experiments. In the following figures, we choose  $P = 6$ . In Fig. 3.17, we fix  $M = 2$  and  $N = 6$ . We remark that, as in the previous section, the VTT-RecALS-SIP becomes more robust when the order increases, and is efficient for

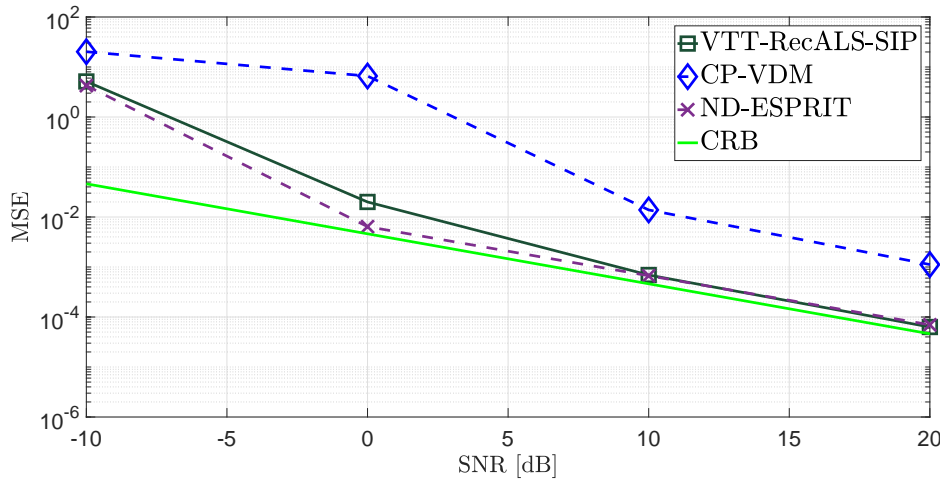
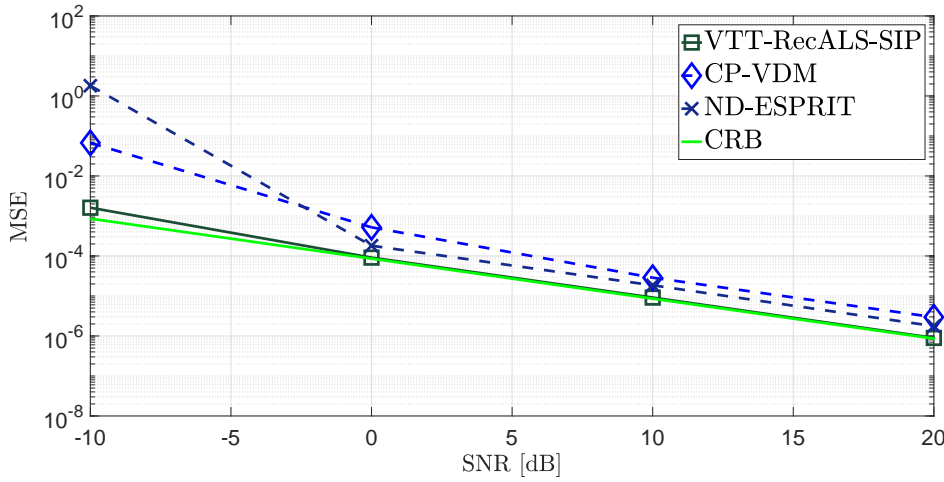
TABLE 3.5: Summary of chosen parameters in Section 3.3.4.2

	Dimension $N$	Rank $M$	Order $P$
Fig. 3.15	6	2	4
Fig. 3.16	6	3	4
Fig. 3.17	6	2	6
Fig. 3.18	8	2	6

FIGURE 3.15: MSE vs SNR in dB for  $P = 4$  with  $M = 2$ ,  $N = 6$ .

a wide range of SNR. This justifies that VTT-RecALS-SIP is well designed for high-order tensors. A last scenario for robustness simulations is with  $P = 6$ ,  $N = 8$  and  $M = 2$ , considered in Fig. 3.18. The ND-ESPRIT is not depicted here due to its too high computational complexity for high-order tensors. In this figure, we compare the VTT-RecALS-SIP algorithm with CPD-RecALS and CPD-RecALS-TR<sub>1</sub>A. We recall that the CPD-RecALS is the RecALS algorithm applied to a  $P$ -order tensor using a naive rectification, by dividing each column by its first entry, to refine the Vandermonde structure, whereas CPD-RecALS-TR<sub>1</sub>A uses the TR<sub>1</sub>A rectification. Note that the CPD-RecALS-TR<sub>1</sub>A is efficient for positive SNR, but is computationally intense, which means that VTT-RecALS-SIP is the best tradeoff between noise robustness and computational complexity in this case.

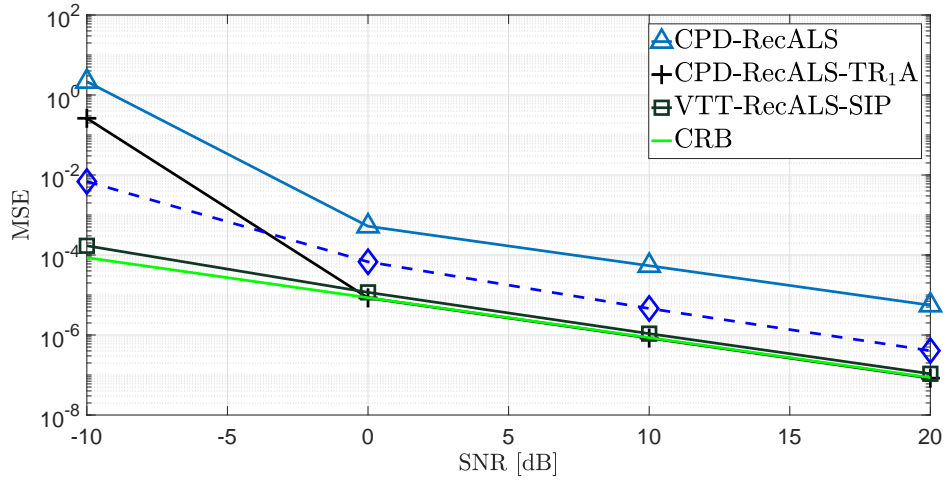
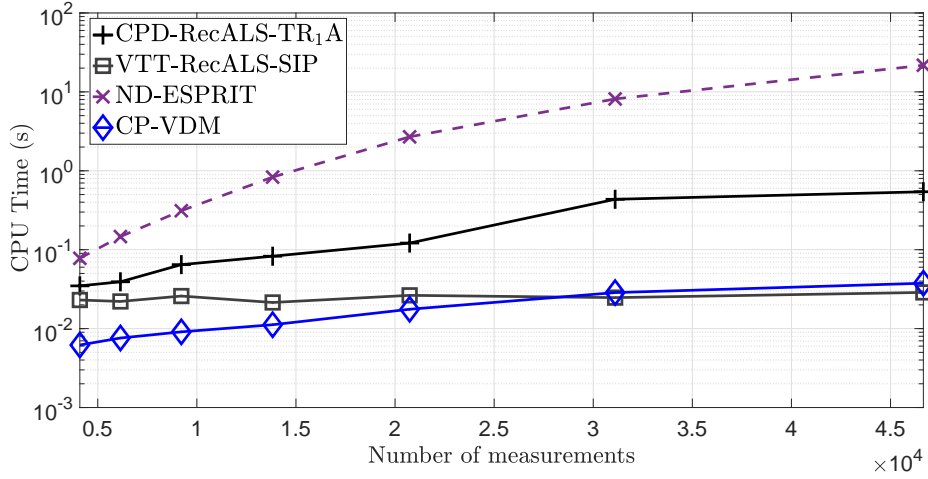
**3.3.4.2.2 Computational times** In this section, the computational time is evaluated using the native “Tic-Toc” functions of MatLab. In the following figures, we generate a 6-order rank-2 tensor of size  $N_1 \times \dots \times N_6$ , we fix  $\text{SNR} = 5\text{dB}$ , and we vary the number of the measurements as the product of the dimensions  $N_1 N_2 \dots N_6$ . In Fig. 3.19, we start with a tensor of size  $4 \times \dots \times 4$ , having  $4^6$  measurements, and we increase each  $N_i$  to  $N_i = 6$ , having at the end  $6^6$  measurements. The same methodology is used in Fig. 3.20, changing each dimension from  $N_i = 6$  to  $N_i = 8$ . Note that in Fig. 3.19 the computational time of ND-ESPRIT grows faster than the other algorithms. For larger number of measurements in Fig. 3.20, the ND-ESPRIT is removed due to its too high computational time. Note that, like all the ALS-based estimators, CPD-RecALS-TR<sub>1</sub>A has also an intense computational time since it is

FIGURE 3.16: MSE vs SNR in dB for  $P = 4$  with  $M = 3$ ,  $N = 6$ .FIGURE 3.17: MSE vs SNR in dB for  $P = 6$  with  $M = 2$ ,  $N = 6$ .

an iterative algorithm applied to high-order tensors. For example, for a 6-order tensor of size  $8 \times \dots \times 8$  which corresponds to  $8^6$  measurements, we have an interesting computational gain of more than 50 for VTT-RecALS-SIP compared to CPD-RecALS-TR<sub>1</sub>A for the same noise robustness (see Fig. 3.18). On the other side, we have a comparable computational time for VTT-RecALS-SIP and CP-VDM which is a non-iterative algorithm, unlike VTT-RecALS-SIP. This can be justified by the dimensionality reduction step, and by the fact that the computational time of VTT-RecALS-SIP is approximately the one of TT-SVD algorithm when  $P \gg 1$ . As a conclusion, we can say that VTT-RecALS-SIP offers the best tradeoff between noise robustness and computational complexity.

### 3.3.5 Discussion

Multidimensional Harmonic Retrieval (MHR) is at the heart of many important signal-based applications. The MHR problem admits a natural formulation into the tensor (*a.k.a.* multi-way array) framework, usually called generalized Vandermonde CPD. Joint exploitation of multi-diversity/modality sensing technologies for data

FIGURE 3.18: MSE vs SNR in dB for  $P = 6$  with  $M = 2$ ,  $N = 8$ .FIGURE 3.19: CPU time versus the number of measurements for  $P = 6$ ,  $M = 2$ ,  $N_i = (4, 6)$ , and  $\text{SNR} = 5\text{dB}$ .

fusion increases inexorably the tensor order/dimensionality. Thus, efficient estimation schemes have to face to the well-known “curse of dimensionality”. The challenge here is to reformulate a high-order tensor as a set of low-order tensors, called cores or nodes into the graph-based formalism. Splitting the initial multidimensional optimization problem into a sum of low dimensionality optimization problems for each node of the graph has at least two advantages. Firstly, ill-converging problems for high dimensional optimization are considerably mitigated. Secondly, thanks to the graph-based formalism, some latent coupling properties between the nodes of the graph can be revealed. As a consequence, new optimization strategies taking the coupling relations into account can be designed. In this work, a new scheme, called VTT-RecALS-SIP, belonging to the JIRAFE (Joint dImensionality Reduction And Factors rEtrieval) family, is proposed for the MHR problem. The aim of the first step of the VTT-RecALS-SIP scheme is to reduce the dimension of the initial Least-Square optimization problem or equivalently to recover the nodes in the popular and simple graph called Tensor Train. We show that due to the MHR problem structure, each node in the TT is associated to a partially structured VTT-core



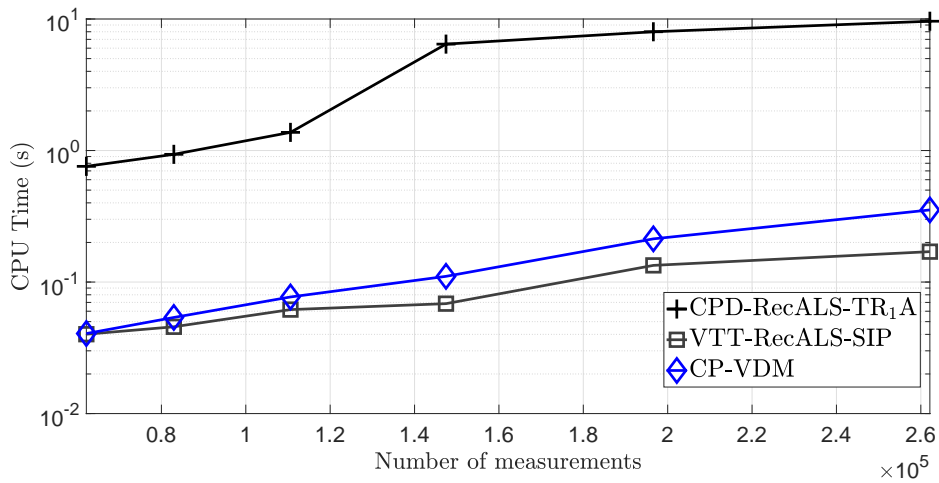


FIGURE 3.20: CPU time versus the number of measurements for  $P = 6$ ,  $M = 2$ ,  $N_i = (6, 8)$ , and  $\text{SNR} = 5\text{dB}$ .

coupled with its two neighbors (in a graph-based sense) VTT-cores. In other words, the initial difficult multidimensional LS optimization problem is now reformulated as a more tractable and flexible equivalent optimization problem, *i.e.*, as the sum of coupled low dimensional LS optimization problems. The second step is dedicated to the Vandermonde-based factor retrieval, *i.e.*, the estimation of the parameters of interest automatically paired. To reach this goal, a new rectified ALS algorithm is proposed and adapted to the exploitation of the coupling properties between the VTT-cores. Specifically, the Vandermonde rectification exploits the Shift Invariance Property (SIP). Numerical simulations show the effectiveness of the proposed VTT-RecALS-SIP method in terms of noise robustness and computational cost compared to other state-of-art methods.



### 3.4 JIRAFE-based MIMO channel decomposition

In this part, we will propose a JIRAFE-based method for a new 5-order MIMO channel tensor model. The proposed method finds the transmit and receive spatial signatures as well as the complex path gains (which also capture the polarization effects).

#### 3.4.1 Context of study

MIMO systems have been subject of intense research due to their great potential to provide substantial energy efficiency and data rate gains [68]. Hence, for MIMO channel modeling and estimation, it is important to accurately estimate path directions in azimuth and elevation, along with the polarization and amplitude parameters at both sides of the link. In multiuser MIMO systems, the knowledge of the channel parameters at the base station (angles of arrival, angles of departure, path gains, and polarization parameters) can be efficiently exploited to realize beamforming designs and deal with multiuser interference. In this context, adopting a parametric approach to model/estimate the MIMO channel enables the use of limited feedback in frequency division duplexing (FDD) systems to provide the base station with the downlink channel parameters for subsequent transmit signal design. In a recent paper [112], a tensor-based approach for dual-polarized MIMO channel estimation has been proposed by recasting the MIMO channel as a fourth-order tensor. The authors assumed a MIMO system with a uniform rectangular array (URA) at the transmitter (e.g. base station) and a uniform linear array (ULA) at the receiver (e.g. user equipment). The identifiability of the channel parameters is thoroughly discussed and a channel estimation algorithm is proposed, the core of which relies on the Alternating Least Squares (ALS) algorithm [60]. Despite being an attractive solution, its computational complexity may still be high, especially when downlink channel estimation is carried out at the user equipment with limited processing capabilities. In this work, URA is considered also at the reception. Note that this particular array geometry is highly relevant, not only for wireless communications, but also for modern radio-interferometry-based telescopes [101]. Exploiting azimuth and elevation diversities at both ends of the link increases the tensor order to five. Even if the ALS-based method proposed in [112] remains a possible solution, our approach is inspired by the tensor network theory [28]. Indeed, the ALS algorithm turns out to be often inefficient for high-order tensors. The principal drawbacks as illustrated in the simulation part of this work are ill-converging problems [92] and a high computational complexity cost. To mitigate this dimensionality problem, the decomposition of the channel tensor is carried out using a Joint dImensionality Reduction And Factor rEtrieval (JIRAFE) principle [156]. The acronym JIRAFE encompasses a flexible and generic family of algorithms which has already been successfully applied in the context of multidimensional harmonic retrieval [158]. More precisely, the fifth-order channel tensor is first decomposed as a graph-based connected lower-order tensors, called cores [104]. The coupled structure of these cores is described in two scenarios of interest, *i.e.*, when only few ( $< 4$ ) propagation paths are dominant and the case where the multi-path propagation condition becomes more severe ( $\geq 4$ ). While the first case is based on some preliminary results given in [156], the core structure for the more challenging situations is described in this work. The second step of the JIRAFE method is dedicated to the factor retrieval for which we exploit the Vandermonde rectification strategy proposed in [15]. Our detailed contributions can be summarized as follows:

1. From a fundamental perspective, the equivalence between CPD and TTD has been presented in [156] for full column rank factors. This equivalence is deeply reformulated in the sense that the structure of the TT-cores changes if the full column rank factor assumption is violated. This is precisely the case of the MIMO channel tensor considered in this work.
2. Comparatively with the channel model considered in [112], the one proposed in this work exploits an URA at the reception, inducing an increase of spatial diversity. To the best of our knowledge, this is the first time that such a tensor model is introduced.
3. The MIMO channel is represented under a TT format, instead of the usual CPD representation, and the structure of the TT-cores is highlighted separately under the assumptions of full column rank and full row rank for the matrix factors.
4. The TT structure characterized by properties of coupling between two adjacent cores containing the same latent matrices, is exploited for dimensionality reduction and channel parameters estimation using the JIRAFE (Joint dImensionality Reduction And Factor rEtieval) scheme.

The proposed JIRAFE-based method allows to find the transmit and receive spatial signatures as well as the complex path gains (which also capture the polarization effects). Monte Carlo simulations show that our proposed TT-based representation of the channel is more robust to noise and computationally more attractive than available competing tensor-based methods, for physical parameters estimation.

### 3.4.2 Tensor-based channel modeling

#### 3.4.2.1 Canonical Polyadic Decomposition (CPD) of the channel tensor

**3.4.2.1.1 Expression of the channel tensor** The steering vectors for the  $k$ -th path for an URA in transmission of size  $M_{T_x} \times M_{T_y}$  and in reception of size  $M_{R_x} \times M_{R_y}$  are respectively,  $\mathbf{a}_T(k) = \mathbf{a}_{T_x}(k) \otimes \mathbf{a}_{T_y}(k)$ ,  $\mathbf{a}_R(k) = \mathbf{a}_{R_x}(k) \otimes \mathbf{a}_{R_y}(k)$  where

$$\mathbf{a}_X(k) = [1, \exp(i\omega_X(k)), \dots, \exp(i\omega_X(k)(M_X - 1))]^T$$

with  $X \in \{T_x, T_y, R_x, R_y\}$ . Note that  $\otimes$  and  $\odot$  denote respectively Kronecker and Khatri-Rao products. The steering matrices for  $K$  paths in transmission and in reception are respectively,  $\mathbf{A}_T = \mathbf{A}_{T_x} \odot \mathbf{A}_{T_y}$  and  $\mathbf{A}_R = \mathbf{A}_{R_x} \odot \mathbf{A}_{R_y}$  in which

$$\mathbf{A}_X = [\mathbf{a}_X(1) \quad \dots \quad \mathbf{a}_X(K)].$$

Now, define  $\beta_k^{(p,q)}$  as the  $k$ -th entry of the vector  $\boldsymbol{\beta}^{(p,q)}$ , with  $1 \leq k \leq K$ , where  $\beta_k^{(p,q)}$  is the generalized (complex) path-loss parameter for the  $k$ -th path and for the  $(p, q)$ -th subchannel. Note that  $p \in \{V_r, H_r\}$  refers to the vertical (V) polarized and horizontal (H) polarized receive antennas, and  $q \in \{V_t, H_t\}$  refers to the V-polarized and H-polarized transmit antennas. In the noise-free scenario, the channel matrix is given by

$$\mathbf{H} = (\mathbf{A}_{T_x}^* \odot \mathbf{A}_{T_y}^* \odot \mathbf{A}_{R_x} \odot \mathbf{A}_{R_y}) \mathbf{B}^T$$

with  $\mathbf{B} = [\boldsymbol{\beta}^{(V_r, V_t)} \boldsymbol{\beta}^{(V_r, H_t)} \boldsymbol{\beta}^{(H_r, V_t)} \boldsymbol{\beta}^{(H_r, H_t)}]^T \in \mathbb{C}^{4 \times K}$ . From this matrix unfolding of the channel tensor  $\mathcal{H}$ , we can conclude that it follows a fifth-order CPD of canonical

rank  $K$ , and size  $M_{T_x} \times M_{T_y} \times M_{R_x} \times M_{R_y} \times 4$ , given by

$$\mathcal{H} = \mathcal{I}_{5,K} \times_1 \mathbf{A}_{T_x}^* \times_2 \mathbf{A}_{T_y}^* \times_3 \mathbf{A}_{R_x} \times_4 \mathbf{A}_{R_y} \times_5 \mathbf{B} + \mathcal{N}. \quad (3.37)$$

The additive term  $\mathcal{N}$  encompasses the background noise and the estimation error due to the pre-estimation of the unstructured channel obtained by sending known pilot sequences from the transmit antennas. If orthogonal pilot sequences are used then the noise tensor  $\mathcal{N}$  can be modeled as zero-mean circularly complex Gaussian random variables. Note that the considered channel tensor model is an extension of the one presented in [112] due to the URA assumption at both the transmitter and the receiver. We draw attention to the fact that the focus of our work is on the extraction of the MIMO channel parameters based on the unstructured noisy channel tensor.

**3.4.2.1.2 Model assumptions** For identifiability concerns, we assume the following constraints.

1. The steering matrices are all of full column rank, which implies

$$K \leq \min\{M_{T_x}, M_{T_y}, M_{R_x}, M_{R_y}\}.$$

2. Two scenarios in terms of the number of propagation paths are of interest:
  - (a)  $K < 4$  (few dominant paths), then  $\mathbf{B}$  is a full column rank matrix.
  - (b)  $K \geq 4$ , then  $\mathbf{B}$  is a full row rank factor matrix.

### 3.4.2.2 Tensor train decomposition (TTD) of the channel tensor

The idea of the TTD [104] is to break the dimensionality/order of  $\mathcal{H}$  into 3-order TT-cores and two matrices according to

$$\mathcal{H} \stackrel{\text{TTD}}{=} \mathbf{G}_1 \times_2^1 \mathbf{G}_2 \times_3^1 \mathbf{G}_3 \times_4^1 \mathbf{G}_4 \times_5^1 \mathbf{G}_5,$$

where  $\mathbf{G}_1 \in \mathbb{C}^{M_{T_x} \times K}$ ,  $\mathbf{G}_2 \in \mathbb{C}^{K \times M_{T_y} \times K}$ ,  $\mathbf{G}_3 \in \mathbb{C}^{K \times M_{R_x} \times K}$ ,  $\mathbf{G}_4 \in \mathbb{C}^{K \times M_{R_y} \times K}$ , and  $\mathbf{G}_5 \in \mathbb{C}^{K \times 4}$ . The product  $\times_p^q$  is defined as in [156]. As mentioned in the previous section,  $\mathcal{H}$  follows a 5-order CPD of canonical rank- $K$ . In this context, the TT-cores can be analytically relied to the desired factors of the CPD. This is the subject of the next section.

## 3.4.3 Joint Dimensionality Reduction and Factor Retrieval (JIRAFE)

### 3.4.3.1 JIRAFE: dimensionality reduction

In the following, we present two theorems on the structure of the TT-cores resulting from the TT-SVD [104] algorithm applied to eq. (3.37).

1. Theorem 11 is a generalization of the result given in [156], *i.e.*, when there exists few ( $K < 4$ ) dominant propagation paths. In this case, all the factors are full-column rank.
2. Theorem 12 modifies Theorem 11 for the more challenging scenario where there is four or more propagation paths. In this case, the last factor is full-row rank.

**Theorem 11.** When all the factors are full-column rank ( $K < 4$ ), the TT-cores are given by

$$\begin{aligned}\mathbf{G}_1 &= \mathbf{A}_{T_x}^* \mathbf{M}_1^{-1}, \quad \mathbf{G}_2 = \mathcal{I}_{3,K} \times_1 \mathbf{M}_1 \times_2 \mathbf{A}_{T_y}^* \times_3 \mathbf{M}_2^{-T}, \\ \mathbf{G}_3 &= \mathcal{I}_{3,K} \times_1 \mathbf{M}_2 \times_2 \mathbf{A}_{R_x} \times_3 \mathbf{M}_3^{-T}, \\ \mathbf{G}_4 &= \mathcal{I}_{3,K} \times_1 \mathbf{M}_3 \times_2 \mathbf{A}_{R_y} \times_3 \mathbf{M}_4^{-T}, \quad \text{and} \quad \mathbf{G}_5 = \mathbf{M}_4 \mathbf{B}^T\end{aligned}$$

where, for  $1 \leq k \leq 4$ ,  $\mathbf{M}_k \in \mathbb{C}^{K \times K}$  are nonsingular transformation matrices. This means that the TT-ranks are all equal to the canonical rank  $K$ , where  $K$  is the number of paths.

**Theorem 12.** If the last factor is full row rank, i.e.,  $K \geq 4$ , then the TT-cores  $\{\mathbf{G}_1, \mathbf{G}_2, \mathbf{G}_3\}$  verify the same factorizations as in Theorem 11 but the two last TT-cores are given by:

$$\mathbf{G}_4 = \mathcal{I}_{3,K} \times_1 \mathbf{M}_3 \times_2 \mathbf{A}_{R_y} \times_3 \mathbf{M}^{-T} \mathbf{B}, \quad \text{and} \quad \mathbf{G}_5 = \mathbf{M}$$

where  $\mathbf{M} \in \mathbb{C}^{4 \times 4}$  is a nonsingular transformation matrix. This means that the TT-ranks are equal to  $(K, K, K, 4)$ .

*Proof.* Both theorems rely on constructive proofs based on the algebraic structure of the TT-SVD algorithm applied to a 5-order CPD tensor. Depending on the rank of  $\mathbf{B}$ , the reasoning in both cases will be the same for all but the two last TT-cores. We recall that  $\mathbf{M}_k$  are *change-of-basis* matrices that appear due to the use of the SVD to extract dominant subspaces [156].  $\square$

**Remark.** The two above theorems show that the TT-core structure mixes physical quantities, i.e.,  $\{\mathbf{A}_{T_x}^*, \mathbf{A}_{T_y}^*, \mathbf{A}_{R_x}, \mathbf{A}_{R_y}, \mathbf{B}\}$  and latent matrices, i.e.,  $\{\mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_3, \mathbf{M}_4, \mathbf{M}\}$ , and that each TT-core is coupled with its two neighbor TT-cores via the latent matrices.

### 3.4.3.2 JIRAFE: CPD factors retrieval

When the TT-cores have been estimated thanks to the TT-SVD algorithm for instance, the aim of the second step of JIRAFE is to propose an estimation strategy exploiting the TT-core structures given in Theorems 11 and 12.

**3.4.3.2.1 Few dominant paths scenario ( $K < 4$ )** When  $K < 4$ , we have to minimize the following criterion with respect to the physical quantities  $\{\mathbf{A}_{T_x}^*, \mathbf{A}_{T_y}^*, \mathbf{A}_{R_x}, \mathbf{A}_{R_y}, \mathbf{B}\}$  and the latent quantities  $\{\mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_3, \mathbf{M}_4\}$ :

$$\begin{aligned}\mathcal{C}_1 &= \|\hat{\mathbf{G}}_1 - \mathbf{A}_{T_x}^* \mathbf{M}_1^{-1}\|_F^2 + \|\hat{\mathbf{G}}_5 - \mathbf{M}_4 \mathbf{B}^T\|_F^2 \\ &+ \|\hat{\mathbf{G}}_2 - \mathcal{I}_{3,K} \times_1 \mathbf{M}_1 \times_2 \mathbf{A}_{T_y}^* \times_3 \mathbf{M}_2^{-T}\|_F^2 \\ &+ \|\hat{\mathbf{G}}_3 - \mathcal{I}_{3,K} \times_1 \mathbf{M}_2 \times_2 \mathbf{A}_{R_x} \times_3 \mathbf{M}_3^{-T}\|_F^2 \\ &+ \|\hat{\mathbf{G}}_4 - \mathcal{I}_{3,K} \times_1 \mathbf{M}_3 \times_2 \mathbf{A}_{R_y} \times_3 \mathbf{M}_4^{-T}\|_F^2.\end{aligned}$$

This criterion is the sum of coupled LS criteria. At this point, we choose deliberately to promote a local/sequential (but fast) optimization method (see Algorithm 9) instead of a global/optimal optimization strategy based for instance on the Lagrangian minimization [47]. In Algorithm 9, we denote by Tri-ALS, the ALS algorithm applied to a 3-order tensor, the acronym TR<sub>1</sub>A stands Toeplitz Rank-1 Approximation and is dedicated to a Vandermonde rectification strategy presented in [15], while KRF denotes a non-iterative method called Khatri-Rao Factorization proposed in [74]. KRF algorithm recovers 3-order CPD factors assuming that one factor is known and full column rank. It computes  $K$  SVDs of rank-one matrices to

recover the remaining two other factors. Since the focus of this work is on the new TT representation for the channel tensor  $\mathcal{H}$ , we could adopt any state-of-art method for estimating the channel, such as least squares or matched filtering techniques.

---

**Algorithm 9** JIRAFE for few dominant propagation paths

---

**Input:** 5-order rank- $K$  tensor  $\mathcal{H}$ , TT-ranks:  $(K, K, K, K)$ .

**Output:** Estimated CPD factors:  $\hat{\mathbf{A}}_{T_x}^*, \hat{\mathbf{A}}_{T_y}^*, \hat{\mathbf{A}}_{R_x}, \hat{\mathbf{A}}_{R_y}, \hat{\mathbf{B}}$ .

- 1: Dimensionality reduction:  $[\hat{\mathbf{G}}_1, \hat{\mathcal{G}}_2, \hat{\mathcal{G}}_3, \hat{\mathcal{G}}_4, \hat{\mathbf{G}}_5] \leftarrow \text{TT-SVD}(\mathcal{H}, R)$ ,
  - 2: CPD factors retrieval:  $[\hat{\mathbf{M}}_1, \hat{\mathbf{A}}_{T_y}^*, \hat{\mathbf{M}}_2^{-T}] \leftarrow \text{Tri-ALS}(\hat{\mathcal{G}}_2, K)$ ,
  - 3:  $[\hat{\mathbf{A}}_{R_x}, \hat{\mathbf{M}}_3^{-T}] \leftarrow \text{KRF}(\hat{\mathcal{G}}_3, \hat{\mathbf{M}}_2, K)$ ,
  - 4:  $[\hat{\mathbf{A}}_{R_y}, \hat{\mathbf{M}}_4^{-T}] \leftarrow \text{KRF}(\hat{\mathcal{G}}_4, \hat{\mathbf{M}}_3, K)$ ,
  - 5:  $\hat{\mathbf{A}}_{T_x}^* = \hat{\mathbf{G}}_1 \hat{\mathbf{M}}_1$ , and  $\hat{\mathbf{B}} = \hat{\mathbf{G}}_5^T \hat{\mathbf{M}}_4^{-T}$ ;
  - 6: Rectification:  $[\hat{\mathbf{A}}_{T_x}^*, \hat{\mathbf{A}}_{T_y}^*, \hat{\mathbf{A}}_{R_x}, \hat{\mathbf{A}}_{R_y}] \leftarrow \text{TR}_1\text{A}(\hat{\mathbf{A}}_{T_x}^*, \hat{\mathbf{A}}_{T_y}^*, \hat{\mathbf{A}}_{R_x}, \hat{\mathbf{A}}_{R_y})$ .
- 

**Remark.** Note that the steering factors have a Vandermonde structure. It then makes sense to use a Tri-ALS algorithm that takes into account the structure of these factors. In the simulations, we will use a class of ALS-based methods that is called RecALS, for Rectified ALS [15]. Note that other methods for angle estimation could also be applied. The chosen RecALS method is a 3-order ALS that integrates a rectification strategy of the Vandermonde structure of the factor matrices. This strategy is also applied on the Vandermonde factors resulting from the KRF algorithm. In the simulations, and considering the use of  $\text{TR}_1\text{A}$  to rectify the Vandermonde structure, we will call the proposed method JIRAFE.

**3.4.3.2.2 More general multi-path scenario** ( $K \geq 4$ ) Based on Theorem 12, we propose a second algorithm which minimizes the following criterion over the physical quantities  $\{\mathbf{A}_{T_x}^*, \mathbf{A}_{T_y}^*, \mathbf{A}_{R_x}, \mathbf{A}_{R_y}, \mathbf{B}\}$  and over the latent quantities  $\{\mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_3, \mathbf{Q}\}$ :

$$\begin{aligned} C_2 = & \|\hat{\mathbf{G}}_1 - \mathbf{A}_{T_x}^* \mathbf{M}_1^{-1}\|_F^2 + \|\hat{\mathbf{G}}_5 - \mathbf{Q}^T \mathbf{B}^T\|_F^2 \\ & + \|\hat{\mathcal{G}}_2 - \mathcal{I}_{3,K} \times_1 \mathbf{M}_1 \times_2 \mathbf{A}_{T_y}^* \times_3 \mathbf{M}_2^{-T}\|_F^2 \\ & + \|\hat{\mathcal{G}}_3 - \mathcal{I}_{3,K} \times_1 \mathbf{M}_2 \times_2 \mathbf{A}_{R_x} \times_3 \mathbf{M}_3^{-T}\|_F^2 \\ & + \|\hat{\mathcal{G}}_4 - \mathcal{I}_{3,K} \times_1 \mathbf{M}_3 \times_2 \mathbf{A}_{R_y} \times_3 \mathbf{Q}\|_F^2 \end{aligned}$$

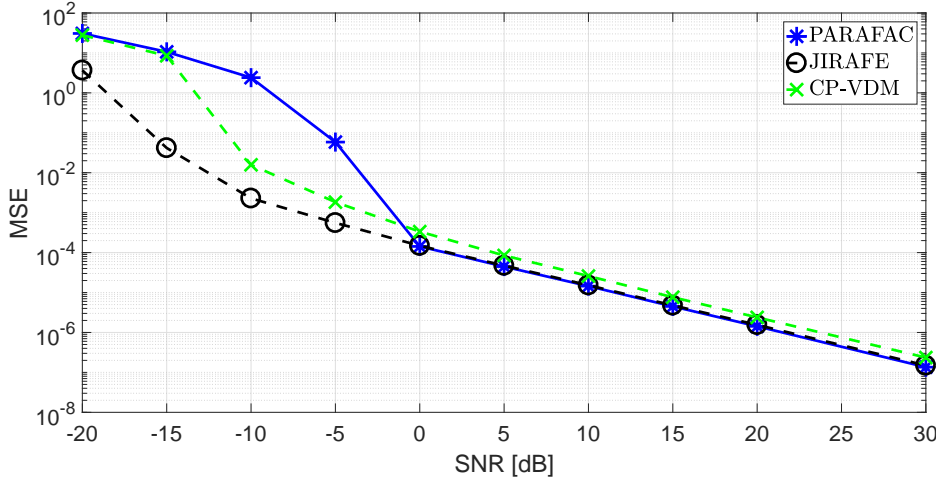
where  $\mathbf{Q} = \mathbf{M}^{-T} \mathbf{B}$  and  $\hat{\mathbf{G}}_5 = \mathbf{M}$ .

Note that the Algorithm 2 is deduced from Algorithm 11 by replacing the TT-ranks by  $(K, K, K, 4)$  in the dimensionality reduction. In addition, lines 4 and 5 become  $[\hat{\mathbf{A}}_{R_y}, \hat{\mathbf{Q}}] \leftarrow \text{KRF}(\hat{\mathcal{G}}_4, \hat{\mathbf{M}}_3, 4)$  and  $\hat{\mathbf{A}}_{T_x}^* = \hat{\mathbf{G}}_1 \hat{\mathbf{M}}_1$ , and  $\hat{\mathbf{B}} = \hat{\mathbf{G}}_5^T \hat{\mathbf{Q}}$ , respectively.

It is worth noting that it has been proven in [156] that the JIRAFE method estimates the CPD factors up to the same trivial ambiguities as for the ALS algorithm. In contrast to the scheme presented in [156], the proposed JIRAFE algorithm replaces the Bi-ALS algorithm by the non-iterative KRF estimator, which allows to mitigate potential ill-convergence problems.

### 3.4.4 Simulation Results

In this section, we show the interest of using the TTD of Section 3.4.2.2 over the CPD through the JIRAFE-based proposed algorithms. The Vandermonde factors

FIGURE 3.21: MSE vs SNR in dB with Alg. 9 for  $K = 3$ .

$\mathbf{A}_{T_x}, \mathbf{A}_{T_y}, \mathbf{A}_{R_x}$ , and  $\mathbf{A}_{R_y}$  are generated, respectively, based on single random realizations of the angular frequencies  $\omega_{T_x}(k), \omega_{T_y}(k), \omega_{R_x}(k)$  and  $\omega_{R_y}(k)$  following a uniform distribution in  $]0, \pi]$ . The factor  $\mathbf{B}$  is drawn from a complex Gaussian distribution with zero mean and unit variance. The scenario for the simulations consists of a receiver and a transmitter both with  $10 \times 8$  URAs, implying that  $\mathcal{H}$  satisfies a 5-order rank- $K$  CPD, of dimensions  $10 \times 8 \times 10 \times 8 \times 4$ . The channel tensor  $\mathcal{H}$  is assumed to be pre-estimated using a supervised approach with orthogonal pilots. The considered MSE concerns the estimation error over the angular frequencies, *i.e.*,

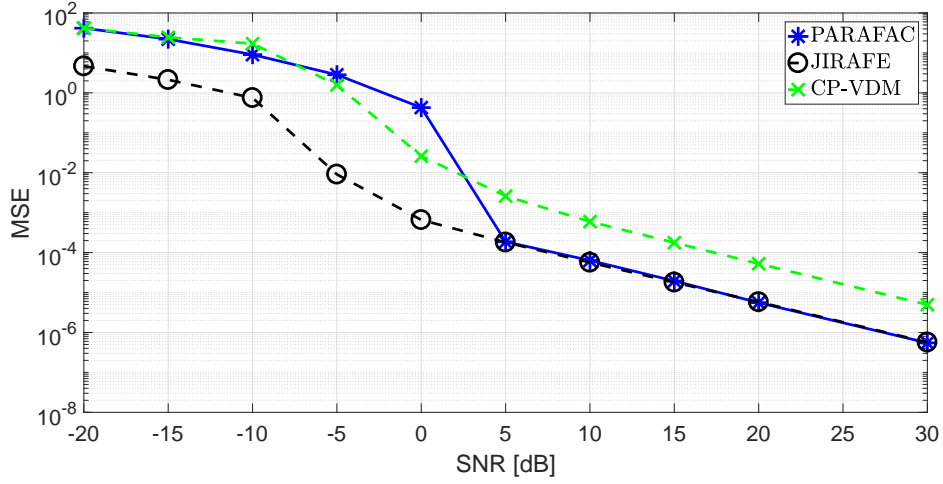
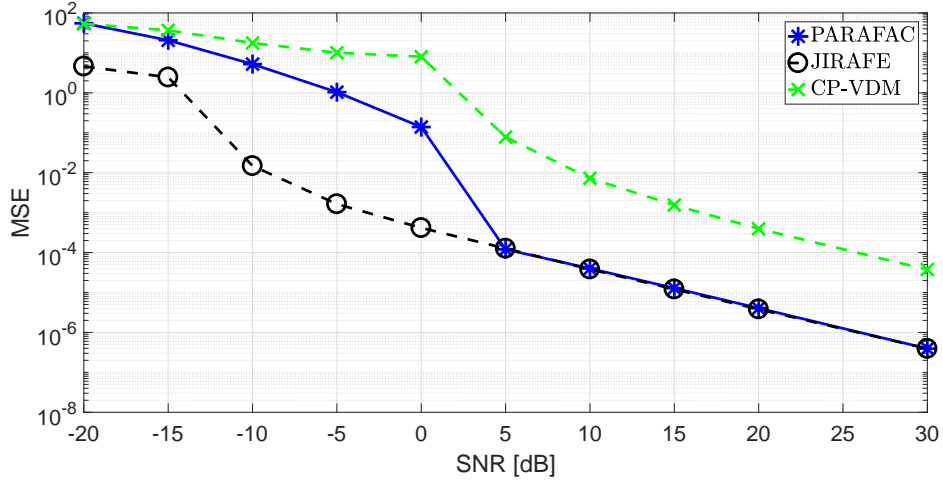
$$\begin{aligned} \text{MSE} = \sum_{k=1}^K & \left( (\omega_{T_x}(k) - \hat{\omega}_{T_x}(k))^2 + (\omega_{T_y}(k) - \hat{\omega}_{T_y}(k))^2 \right. \\ & \left. + (\omega_{R_x}(k) - \hat{\omega}_{R_x}(k))^2 + (\omega_{R_y}(k) - \hat{\omega}_{R_y}(k))^2 \right), \end{aligned}$$

the signal to noise ratio (SNR) is defined as

$$\text{SNR [dB]} = 10 \log \frac{\|\mathcal{H}\|_F^2}{\|\mathcal{N}\|_F^2}.$$

The depicted MSE is calculated by averaging the results over 1000 independent Monte Carlo runs, truncated from 5% worst and 5% best MSEs to eliminate the influence of ill-convergence experiments and outliers. In this work, we assume random initialization in the ALS for  $\mathcal{G}_2$ , eigendecomposition-based initialization can also be considered at the expense of an additional computational cost. The proposed method is compared to two state-of-art algorithms, the ALS-based solution proposed in [112], called PARAFAC, which uses an ALS algorithm followed by closed-form solutions to estimate the parameters from the factors, and the so-called CP-VDM, for CPD with Vandermonde factor matrix, proposed in [135]. All algorithms are applied to the more general model in eq. (3.37). In Fig. 3.21, we fix  $K = 3$ , *i.e.*, the last factor has a full column rank. One may remark that JIRAFE and PARAFAC have same robustness to noise for a wide range of positive SNR. For negative SNRs, JIRAFE is the most robust estimator. This can be justified by the noise reduction property of the truncated SVD when the TT-SVD is applied. The same remark can be made for Fig. 3.22 where  $K = 4$ . In this figure, we can see that both, JIRAFE and



FIGURE 3.22: MSE vs SNR in dB with Alg. 2 for  $K = 4$ .FIGURE 3.23: MSE vs SNR in dB with Alg. 2 for  $K = 5$ .

PARAFAC, are competing for high SNRs. Meanwhile, JIRAFE is more robust than the other estimators for low SNRs. In Fig. 3.23, the number of paths is fixed at  $K = 5$ , which means that the last factor is full row rank. We have a similar behavior for JIRAFE and PARAFAC as in the last experiment. On the other hand, CP-VDM has difficulties when the rank increases. It is worth noting that breaking the dimensionality of tensor  $\mathcal{H}$  helps to improve the convergence of the ALS algorithm, since with JIRAFE, the ALS is applied to 3-order tensors instead of the original fifth-order tensor. In Fig. 3.24, we plot the mean number of iterations for JIRAFE and PARAFAC. The bars represent the standard deviation for each SNR. We notice that 3-order ALS needs less iterations to converge, while keeping in mind that the computational cost of a 3-order ALS iteration, that needs  $O(3K^2M^2)$  flops, is very low compared to that of a 5-order ALS with  $O(5K^2M^4)$  flops, where  $M = \max(M_{T_x}, M_{T_y}, M_{R_x}, M_{R_y}) = 10$ . Tab. 3.6 gives the average computation time for each method. The proposed JIRAFE provides the best tradeoff between noise robustness and computational complexity.

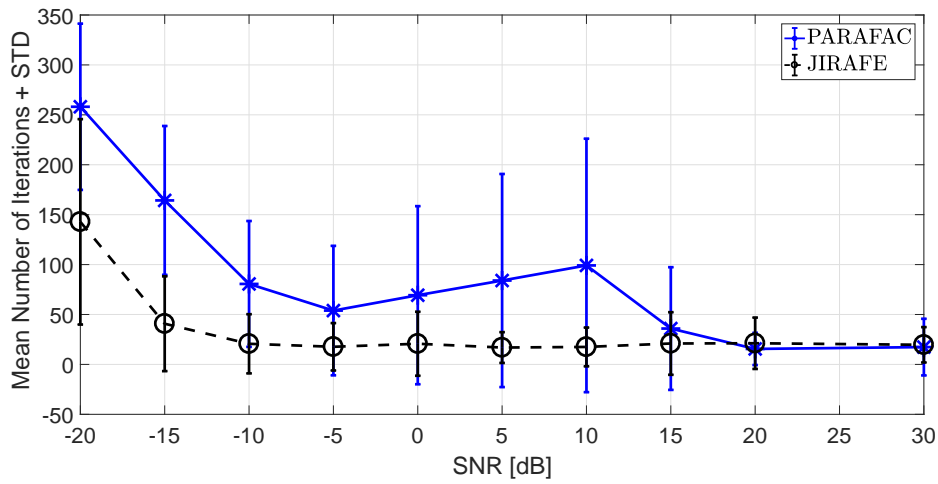
FIGURE 3.24: Mean number of iteration for  $K = 2$ .

TABLE 3.6: Comparison of the computation time with SNR= 15dB.

Canonical rank $K$	2	3	4	5
PARAFAC	0,207 (s)	1,117 (s)	1,593 (s)	2,143 (s)
JIRAFE	0,026 (s)	0,063 (s)	0,088 (s)	0,215 (s)
CP-VDM	0,015 (s)	0,018 (s)	0,021 (s)	0,021 (s)

### 3.4.5 Discussion

In this application, an extension of a MIMO channel is considered using URAs both at the transmitter and the receiver, which leads to a fifth-order channel tensor. A TT-based representation has been derived for this tensor, highlighting the coupling between two adjacent core tensors via the latent matrices. For a multi-path scenario, a new JIRAFE-based method has been proposed for channel parameters estimation. This method allows to break the dimensionality of the original fifth-order CPD into a train of third-order tensors. Simulation results show the effectiveness of the proposed TT-based channel representation in terms of noise robustness and computation time for retrieving the physical channel parameters.



### 3.5 Conclusion

The proposed JIRAFE framework represents a suited solution for the curse of dimensionality. JIRAFE allows to break a high dimensional optimization problem into a set of coupled 3-dimensional optimization problems. We did show that JIRAFE has several advantages in terms of storage cost, computational complexity, convergence and estimation accuracy. It is a general framework where different solutions for the factor retrieval step can be considered depending on the structure of the factors. Moreover, some results on the uniqueness when the factors have linear dependencies have been details. We showed than more restrictive uniqueness conditions are the price to pay for having a numerically efficient JIRAFE-based algorithm, that guarantees recovery of the factor matrices. Two realistic applications have been considered in this work, first, we applied the JIRAFE framework to the Multidimensional Harmonic Retrieval problem, and then, for estimating the physical parameters regarding a MIMO channel tensor. In both applications, interesting simulation results were found for the JIRAFE framework.

## Chapter 4

# A TT-based hierarchical framework

In this chapter, we exploit the ideas developed for the hierarchical/tree Tucker decomposition in the context of the TT decomposition. Specifically, a new efficient TT decomposition scheme [155], called TT-HSVD for Tensor-Train Hierarchical SVD, is proposed as a solution to compute the TT decomposition of a high-order tensor. The new algorithm simultaneously delivers the TT-core tensors and their TT-ranks in a hierarchical way. It is a stable (*i.e.*, non-iterative) and much less complex than the TT-SVD, which is very important when dealing with large-scale data to break the curse of dimensionality. Then, we will also expose a communication-based application, namely MIMO-OFDM relaying systems [159], where the TT modeling can be a natural/direct model, and where a TTD based solution using the TT-HSVD is proposed allowing a joint channel and symbol estimation.

### 4.1 TT-HSVD algorithm

#### 4.1.1 Motivations for a new hierarchical framework

Massive and heterogeneous data processing and analysis have been clearly identified by the scientific community as key problems in several application areas [28, 31, 30]. It was popularized under the generic terms of “data science” or “big data”. Processing large volumes of data, extracting their hidden patterns, while performing prediction and inference tasks has become crucial in economy, industry and science. Modern sensing systems exploit simultaneously different physical technologies. Assume that  $D$  specific sensing devices are available, to measure  $D$  signals parameterized by different parameters. Treating independently each set of measured data is clearly a reductive approach. By doing that, “hidden relationships” or inter-correlations between the datasets may be totally missed. Tensor decompositions have received a particular attention recently due to their capability to handle a variety of mining tasks applied to massive datasets, being a pertinent framework taking into account the heterogeneity and multi-modality of the data. In this case, data can be arranged as a  $D$ -dimensional array, also referred to as a  $D$ -order tensor, and denoted by  $\mathcal{X}$ . In the context of big data processing and analysis, the following properties are desirable:

- a stable (*i.e.*, non-iterative) recovery algorithm
- a low storage cost (*i.e.*, the number of free parameters must be linear in  $D$ )
- an adequate graphical formalism allowing a simple but rigorous visualization of the decomposition of tensors with  $D > 3$

In this section, our interest is on the TT decomposition and the associated TT-SVD algorithm [105]. It is worth noting that TT-SVD is a sequential algorithm, *i.e.*, the

TT-cores are computed one after the other and not at the same time. Moreover, it is a very costly algorithm in terms of complexity, since it involves the application of SVD to matrices of very large sizes. To tackle the computational and storage complexity problem associated with the decomposition of large-scale tensors, a number of methods have been proposed [30], which either replace non-iterative methods by closed-form ones when the tensor is structured [5], or exploit sparsity of the data [80, 107] or, yet, reduce the size of the data using compression with parallel processing [98, 129]. As an alternative to the TT-SVD algorithm, we propose a new algorithm called TT-HSVD algorithm. This algorithm allows to simultaneously recover the TT-core tensors and their TT-ranks in a hierarchical way and is much less complex than TT-SVD. The proposed Tensor Train - Hierarchical SVD (TT-HSVD) algorithm adopts a new unfolding and reshaping strategy that, on one hand, enables to parallelize the decomposition across several processors and, on the other hand, results in a less expensive computational cost compared to competing solutions based on the TT-SVD. An algebraic analysis of the two algorithms is carried out, showing that TT-SVD and TT-HSVD compute the same TT-ranks and TT-core tensors up to specific bases. Simulation results for different tensor orders and dimensions corroborate the effectiveness of the proposed TT-HSVD algorithm.

## 4.1.2 Hierarchical methodology to compute the TT decomposition

### 4.1.2.1 Description of the TT-HSVD algorithm

In this section, the TT-HSVD algorithm is presented, with the aim to derive the TT-cores in a parallel hierarchical way. The main difference between the TT-SVD and TT-HSVD algorithms lies in the reshaping strategy, i.e. the way to reshape the SVD factors  $(\mathbf{U}_d, \mathbf{V}_d)$ , at each step. Fig. 4.1 illustrates the proposed strategy by means of the graph-based representation of the TT decomposition of a 4-order tensor. This figure is to be compared with Fig. 2.4. With the TT-HSVD algorithm, for an *a priori* chosen index  $\bar{D} \in \{1, \dots, D\}$ , the first matrix unfolding  $\mathbf{X}_{(\bar{D})}$  is of size  $(I_1 \cdots I_{\bar{D}}) \times (I_{\bar{D}+1} \cdots I_D)$  instead of  $I_1 \times (I_2 \cdots I_D)$  as for the TT-SVD algorithm, which leads to a more rectangular matrix. Its  $R_{\bar{D}}$ -truncated SVD provides two factors  $\mathbf{U}_{\bar{D}}$  and  $\mathbf{V}_{\bar{D}}$  of size  $(I_1 \cdots I_{\bar{D}}) \times R_{\bar{D}}$  and  $R_{\bar{D}} \times (I_{\bar{D}+1} \cdots I_D)$ , respectively. These two factors are now reshaped in parallel, which constitutes the main difference with the TT-SVD algorithm for which only a single reshaping operation is applied to  $\mathbf{V}_1$ . This processing is repeated after each SVD computation, as illustrated in Fig. 4.1 for a 4-order tensor.

Generally speaking, the choice of the best reshaping strategy, i.e., the choice of the index  $\bar{D}$ , is depending on an *a priori* physical knowledge related to each application [79, 18], as for instance, in bio-medical signal analysis, or in wireless communications [4, 152, 48]. In Section 4.1.5, this best choice is discussed in terms of algorithmic complexity. To illustrate this choice, two reshaping strategies are considered in Fig. 4.2 for computing the TT decomposition of a 8-order tensor with the TT-HSVD algorithm. More precisely, Fig. 4.2 (left) corresponds to a balanced unfolding with  $\bar{D} = 4$ , while Fig. 4.2 (right) corresponds to an unbalanced unfolding with  $\bar{D} = 3$ . From this simple example, one can conclude that this graph-based representation is not sufficiently compact in case of high order, which motivated the new graph-based representation using patterns, introduced in the next section.

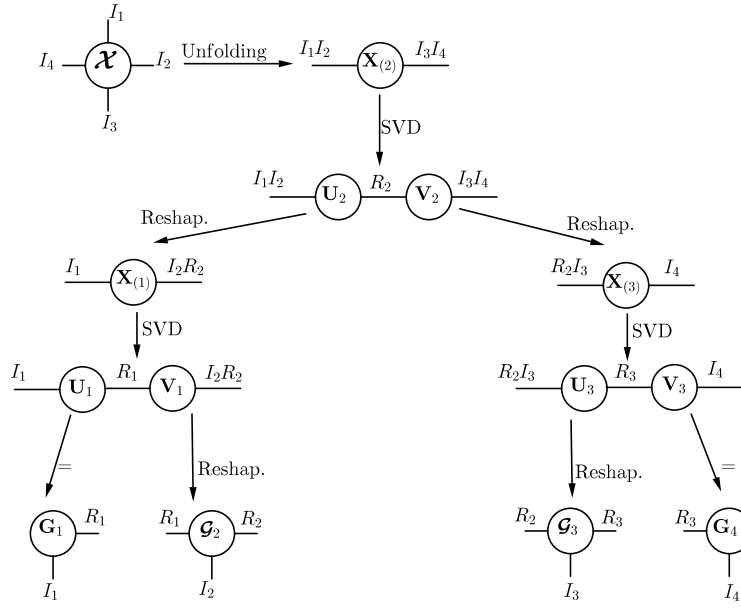


FIGURE 4.1: TT-HSVD applied to a 4-order tensor.

#### 4.1.2.2 Graph-based representation with patterns

A group of algorithmic instructions can be identified as recurrent in the TT-SVD and TT-HSVD algorithms. Such a group will be called a **pattern**. The concept of pattern is introduced in this work as a useful graphical tool to model the processing steps of TT decomposition algorithms in an illustrative way. A pattern can be described equivalently under a pseudo-code formalism. This equivalence allows to represent the TTD of high-order tensor in a more efficient way. It also facilitates understanding of the main steps of the TT-HSVD algorithm, although it can also be used to describe any algorithm. Three types of pattern are now described.

##### 4.1.2.2.1 Splitting/Splitting pattern

The Splitting/Splitting pattern takes as input any matrix unfolding  $\mathbf{X}_{(D)}$  of size  $R_{\bar{D}_f}(I_{\bar{D}_f+1}I_{\bar{D}_f+2} \cdots I_{\bar{D}}) \times (I_{\bar{D}+1}I_{\bar{D}+2} \cdots I_{\bar{D}_l})R_{\bar{D}_l}$ , where  $\bar{D}_f$  stands for the first index and  $\bar{D}_l$  for the last index. This pattern applies the SVD to the input matrix and generates a matrix  $\mathbf{U}_{\bar{D}}$ , of size  $R_{\bar{D}_f}(I_{\bar{D}_f+1}I_{\bar{D}_f+2} \cdots I_{\bar{D}}) \times R_{\bar{D}}$ , which contains the left singular vectors, and a matrix  $\mathbf{V}_{\bar{D}}$ , of size  $R_{\bar{D}} \times (I_{\bar{D}+1}I_{\bar{D}+2} \cdots I_{\bar{D}_l})R_{\bar{D}_l}$ , equal to the product of the diagonal singular values matrix with the matrix composed of the right singular vectors. These two factors ( $\mathbf{U}_{\bar{D}}, \mathbf{V}_{\bar{D}}$ ) are then reshaped using two new indices  $\bar{D}'$  and  $\bar{D}''$ . This pattern is represented in Fig. 4.3. The corresponding graph is characterized by one input  $\mathbf{X}_{(D)}$ , two indices  $\bar{D}'$  and  $\bar{D}''$ , and two outputs  $\mathbf{X}_{(\bar{D}_f+\bar{D}' )}$  and  $\mathbf{X}_{(\bar{D}+\bar{D}'' )}$ . This pattern plays the role of data processing unit, before generating the desired TT-cores using other patterns that will be called core generation patterns. One can notice that this pattern is always used at the top of the TT-HSVD algorithm, with  $\bar{D}_f = 0$ , and  $\bar{D}_l = D$ , the tensor order. (See Figs. 4.1, 4.2 (left) and 4.2 (right)).

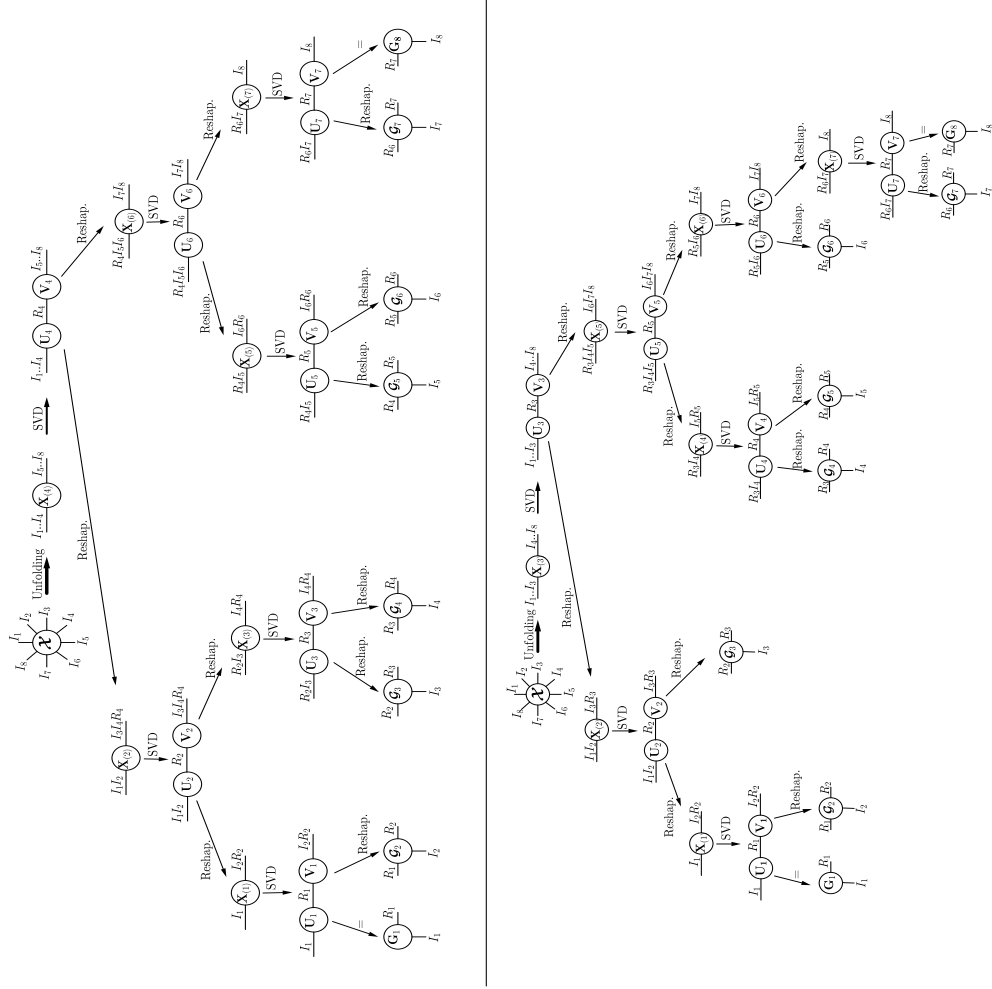


FIGURE 4.2: Balanced (left) and Unbalanced (right) TT-HSVD applied to a 8-order tensor.

**4.1.2.2.2 Mixed patterns: data processing and TT-core generation** The first mixed pattern, that will be called Splitting/Generation pattern, takes as input any matrix  $\mathbf{X}_{(\bar{D})}$  of size  $R_{\bar{D}_f}(I_{\bar{D}_f+1}I_{\bar{D}_f+2} \cdots I_{\bar{D}}) \times (I_{\bar{D}+1}R_{\bar{D}+1})$ , and returns a reshaped matrix and a tensor. It has the same structure as the splitting/splitting pattern, except that it generates a matrix and a tensor instead of two matrices. This pattern is represented in Fig. 4.4 (left). For example, this pattern can be seen in Fig. 4.2 (right) as the function that takes as input  $\mathbf{X}_{(2)}$  and generates  $\mathbf{X}_{(1)}$  and  $\mathcal{G}_3$ . The second mixed pattern, that will be called Generation/Splitting pattern, takes as input any matrix  $\mathbf{X}_{(\bar{D})}$  of size  $(R_{\bar{D}-1}I_{\bar{D}}) \times (I_{\bar{D}+1}I_{\bar{D}+2} \cdots I_{\bar{D}_l})R_{\bar{D}_l}$ , and returns a tensor and a reshaped matrix. This pattern is represented in Fig. 4.4 (right). For example, this pattern can be seen in Fig. 4.2 (right) as the function that takes as input  $\mathbf{X}_{(6)}$  and generates  $\mathcal{G}_6$  and  $\mathbf{X}_{(7)}$ .

**4.1.2.2.3 TT-core generation pattern** The TT-core generation pattern generates two core tensors in parallel. It will be called Generation/Generation pattern. It is represented in Fig. 4.5. It takes as input  $\mathbf{X}_{(\bar{D})}$  of size  $(R_{\bar{D}-1}I_{\bar{D}}) \times (I_{\bar{D}+1}R_{\bar{D}+1})$  and returns two core tensors as outputs. For example, this pattern can be recognized in Fig. 4.2 (right) as the function that takes as input  $\mathbf{X}_{(4)}$  and generates  $\mathcal{G}_4$  and  $\mathcal{G}_5$ .

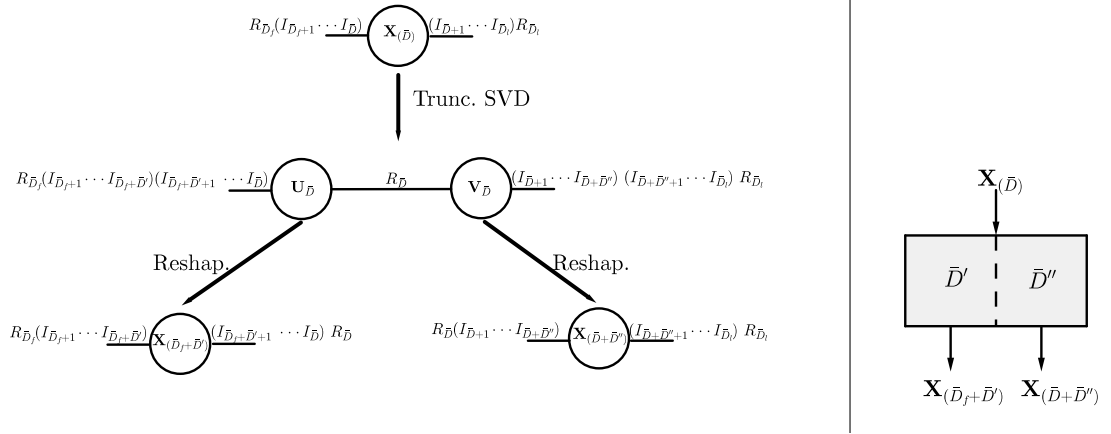


FIGURE 4.3: Splitting/Splitting pattern.

#### 4.1.2.3 Application of the pattern formalism

Note that the TT-SVD algorithm uses one Splitting/Splitting pattern at the beginning and then a sequence of Generation/Splitting patterns, while the three different patterns can be used for the TT-HSVD algorithm. Figs. 4.6 and 4.7 illustrate the pattern-based representation of the TT-HSVD algorithm applied to an 8-order tensor, in the balanced and unbalanced cases, respectively. These figures are to be compared with Figs. 4.2 (left) and 4.2 (right).

A pseudo-code of the TT-HSVD based on the patterns formalism is presented in Algorithm 10.

---

#### Algorithm 10 TT-HSVD algorithm

---

**Input:**  $D$ -order tensor  $\mathcal{X}$ , set of indices  $\bar{D}, \bar{D}'$  and  $\bar{D}''$ .

**Output:** TT-cores:  $\hat{\mathcal{G}}_1^{hrl}, \hat{\mathcal{G}}_2^{hrl}, \dots, \hat{\mathcal{G}}_{D-1}^{hrl}, \hat{\mathcal{G}}_D^{hrl}$ .

- 1: Apply a splitting/splitting pattern to the matrix  $\mathbf{X}_{(\bar{D})}$  of size  $I_1 I_2 \dots I_{\bar{D}} \times I_{\bar{D}+1} I_{\bar{D}+2} \dots I_{\bar{D}}$  with respect to indices  $\bar{D}'$  and  $\bar{D}''$ .  
**while** the outputs have 2 or more of the original tensor dimensions in their row or column dimensions
  - 2: Choose to apply either a splitting/splitting, a generation/splitting or a splitting/generation patterns on the matrices  $\mathbf{X}_{(\bar{D})}$  of size  $R_{\bar{D}_f}(I_{\bar{D}+1} I_{\bar{D}+2} \dots I_{\bar{D}}) \times (I_{\bar{D}+1} I_{\bar{D}+2} \dots I_{\bar{D}_l}) R_{\bar{D}_l}$  with respect to indices  $\bar{D}'$  and  $\bar{D}''$ .  
**end**
  - 3: Apply a generation/generation pattern to any output matrix  $\mathbf{X}_{(\bar{D})}$  of size  $(R_{\bar{D}-1} I_{\bar{D}}) \times (I_{\bar{D}+1} R_{\bar{D}+1})$ .
- 

#### 4.1.3 Algebraic analysis of the TT-SVD and TT-HSVD algorithms

From an algorithmic point of view, the TT-HSVD algorithm is based on a different reshaping-strategy compared to the TT-SVD algorithm leading to a more flexible way to compute the TT-cores. From the algebraic point of view, it is crucial to study the relationship between the estimated TT-cores and the true ones. It is important

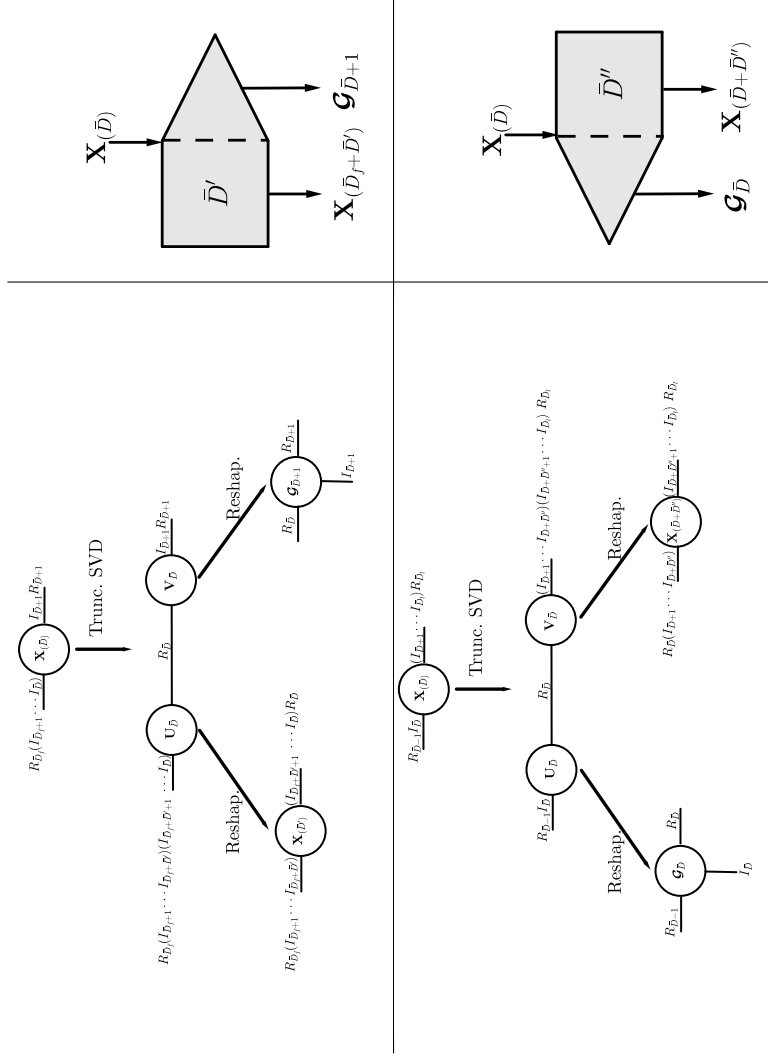


FIGURE 4.4: Splitting/Generation pattern (left), Generation/Splitting pattern (right).

to note that although the TT-SVD has been proposed in [105], its algebraic analysis is to the best of our knowledge original and is carried out for the first time in this work. In [144, 119], the non-uniqueness of the TT decomposition is discussed. Each TT-core can be post and pre-multiplied by any sequence of invertible matrices. In Lemma 4 and Lemma 5, we expose this property in the context of the TT-SVD and the TT-HSVD algorithms. Specifically, we show that these quantities play the role of change-of-basis matrices when a dominant subspace is extracted by the SVD. First, we recall that for a given rank-deficient matrix, the dominant left singular vectors of the SVD span the column space up to an invertible *change-of-basis* matrix. In the sequel, these matrices are denoted by  $\mathbf{P}$  or  $\mathbf{Q}$ .

#### 4.1.3.1 Structure of the estimated TT-cores for the TT-SVD algorithm

**Lemma 4.** Let  $\mathcal{G}_d^{\text{seq}}$  be the sequentially estimated TT-core using the  $d$ -th TT-SVD algorithm and define a set of change-of-basis matrices  $\{\mathbf{P}_1, \dots, \mathbf{P}_{D-1}\}$  with  $\mathbf{P}_d$  of dimensions  $R_d \times$

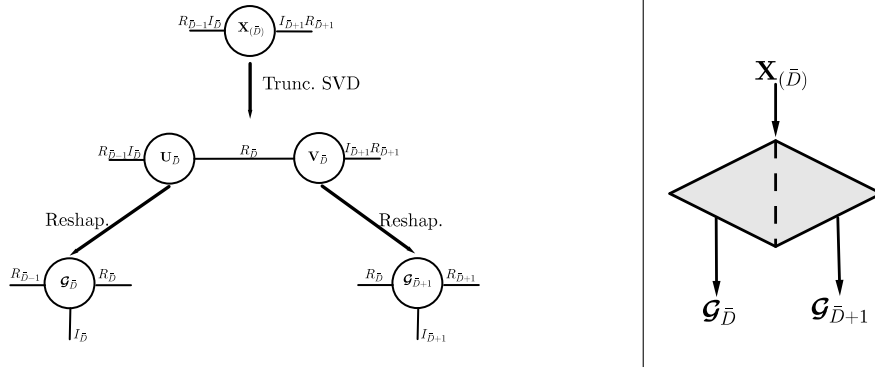


FIGURE 4.5: Generation/Generation pattern.

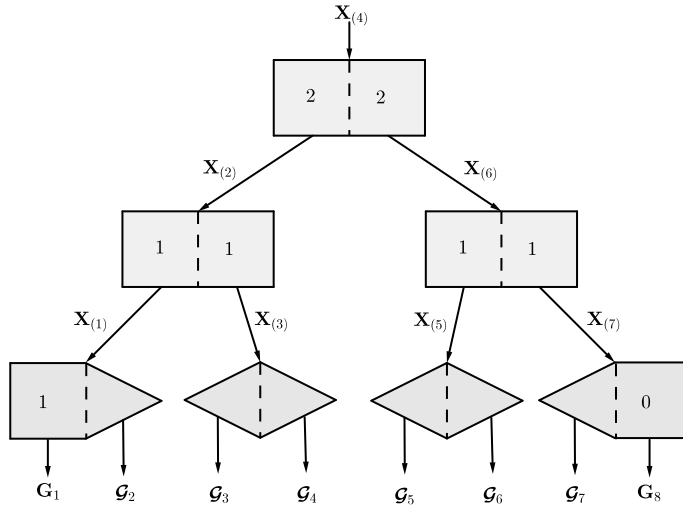


FIGURE 4.6: Graph-based representation using patterns for a balanced TT-HSVD applied to an 8-order tensor.

$R_d$ . The TT-cores associated with the TT-SVD algorithm verify the following relations:

$$\begin{aligned}\hat{\mathbf{G}}_1^{seq} &= \mathbf{G}_1 \mathbf{P}_1, \\ \hat{\mathbf{G}}_d^{seq} &= \mathbf{P}_{d-1}^{-1} \times_2^1 \mathbf{G}_d \times_3^1 \mathbf{P}_d, \quad \text{for } 2 \leq d \leq D-1, \\ \hat{\mathbf{G}}_D^{seq} &= \mathbf{P}_{D-1}^{-1} \mathbf{G}_D.\end{aligned}\tag{4.1}$$

*Proof.* Based on the algebraic formalism of the patterns given in Appendix D, and giving the facts that

- Splitting/Splitting patterns are always applied first, before applying any other type of patterns in the TT-SVD algorithm, *i.e.*, we always generate matrices  $\mathbf{G}_1$  and  $\mathbf{X}_{(2)}$  from matrix  $\mathbf{X}_{(1)}$  at the first step.
- Generation/Splitting pattern is always applied after Generation/Splitting and Splitting/Splitting patterns in the TT-SVD. A combination of this type is allowed, since the format of the outputs of these latters corresponds to the format of the input of the Generation/Splitting pattern. This means that the expressions of the outputs in (D.7) and (D.9) have the same structure of the input in (D.8).



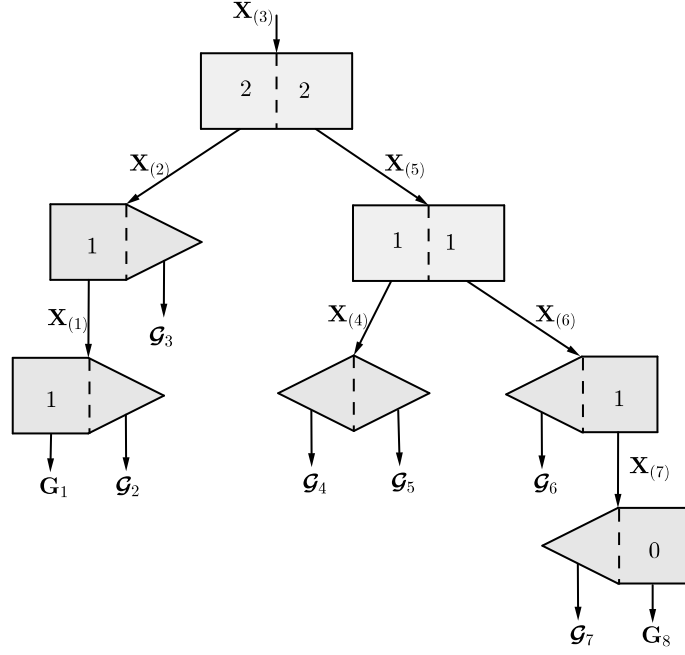


FIGURE 4.7: Graph-based representation using patterns for an unbalanced TT-HSVD applied to an 8-order tensor.

- The TT-SVD algorithm can be seen as a Splitting/Splitting pattern followed by a succession of Generation/Splitting patterns.

The expression of the Generation/Splitting pattern output, given in (D.10), shows that the TT-core associated to the TT-SVD admits a general and final formulation given by (4.1).  $\square$

#### 4.1.3.2 Presentation and analysis of the TT-HSVD algorithm

In the following, we present the TT-HSVD algorithm. We also establish the link between the TT-HSVD and TT-SVD algorithms, in terms of the TT-cores and TT-ranks.

**4.1.3.2.1 Structure of the estimated TT-cores** Hereafter, we formulate a similar result for the TT-HSVD.

**Lemma 5.** Let  $\hat{\mathcal{G}}_d^{hrl}$  be the hierarchically estimated  $d$ -th TT-core using the TT-HSVD algorithm and define a set of change-of-basis matrices  $\{\mathbf{Q}_1, \dots, \mathbf{Q}_{D-1}\}$  where  $\mathbf{Q}_d$  is a  $R_d \times R_d$ . The TT-cores associated with the TT-HSVD algorithm verify the following relations:

$$\begin{aligned} \hat{\mathbf{G}}_1^{hrl} &= \mathbf{G}_1 \mathbf{Q}_1 \\ \hat{\mathcal{G}}_d^{hrl} &= \mathbf{Q}_{d-1}^{-1} \times_2^1 \mathcal{G}_d \times_3^1 \mathbf{Q}_d, \quad \text{for } 2 \leq d \leq D-1 \\ \hat{\mathbf{G}}_D^{hrl} &= \mathbf{Q}_{D-1}^{-1} \mathbf{G}_D. \end{aligned} \tag{4.2}$$

*Proof.* The demonstration of this Lemma is based on the algebraic formalism of the patterns given in Appendix D. Note that in the TT-HSVD algorithm:

- The Splitting/Splitting patterns are always applied first, before applying any other type of patterns.

- The input and the outputs of the Splitting/Splitting pattern have the same format. Any combination of Splitting/Splitting patterns is possible and allowed (See (D.1), (D.6) and (D.7)).
- The output of the Splitting/Splitting pattern has the same format as the input of all other generation patterns. Any combination of Splitting/Splitting patterns with the other patterns is possible and allowed (See (D.6), (D.7), (D.8) and (D.11)).

□

Based on the expressions of the generation patterns outputs, given in (D.10), (D.12) and (D.13), it can be noticed that the TT-core associated to the TT-HSVD admits a general and final formulation given by (4.2).

#### 4.1.3.3 Comparison of the two schemes

In Fig. 4.8 and Fig. 4.9, we can see that the TT-SVD and the TT-HSVD in the algebraic perspective estimate the same TT-core up to different change-of-basis matrices.

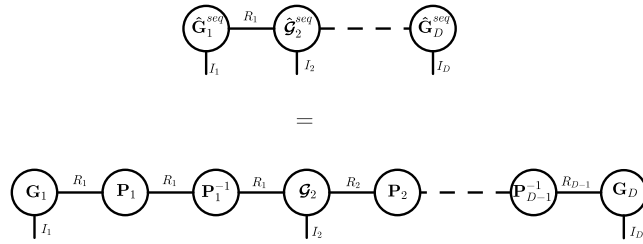


FIGURE 4.8: Graph-based illustration of the TT-SVD algorithm.

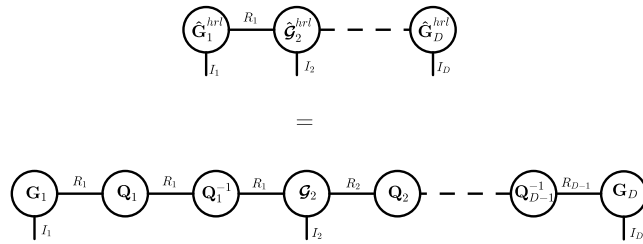


FIGURE 4.9: Graph-based illustration of the TT-HSVD algorithm.

Based on the previous relations on the structure of the TT-cores, the following result can be formulated.

**Lemma 6.** Define a set of matrices:  $\{\mathbf{H}_1, \dots, \mathbf{H}_{D-1}\}$ , with  $\mathbf{H}_d = \mathbf{P}_d^{-1} \mathbf{Q}_d$ . The TT-cores obtained by applying the TT-SVD and TT-HSVD algorithms satisfy the following relations:

$$\hat{\mathbf{G}}_1^{hrl} = \hat{\mathbf{G}}_1^{seq} \mathbf{H}_1, \quad (4.3)$$

$$\hat{\mathbf{G}}_d^{hrl} = \mathbf{H}_{d-1}^{-1} \times_2^1 \hat{\mathbf{G}}_d^{seq} \times_3^1 \mathbf{H}_d \quad \text{for } 2 \leq d \leq D-1, \quad (4.4)$$

$$\hat{\mathbf{G}}_D^{hrl} = \mathbf{H}_{D-1}^{-1} \hat{\mathbf{G}}_D^{seq}. \quad (4.5)$$

*Proof.* Note that to demonstrate Eq. (4.4), we use the identity  $\mathbf{A} \times_2^1 \mathbf{B} = \mathbf{AB}$ . From (4.1), we deduce:

$$\mathcal{G}_d = \mathbf{P}_{d-1} \times_2^1 \hat{\mathcal{G}}_d^{seq} \times_3^1 \mathbf{P}_d^{-1}.$$

Substituting this last equation into (4.2), we have:

$$\hat{\mathcal{G}}_d^{hrl} = (\mathbf{P}_{d-1}^{-1} \mathbf{Q}_{d-1})^{-1} \times_2^1 \hat{\mathcal{G}}_d^{seq} \times_3^1 (\mathbf{P}_d^{-1} \mathbf{Q}_d) \quad \text{for } 2 \leq d \leq D-1$$

□

The above Lemma allows us to formulate the following theorem.

**Theorem 13.** *The TT-ranks are equal for the TT-SVD and the TT-HSVD algorithms.*

*Proof.* From (4.3) and (4.5) in Lemma 6, the proof of the theorem is straightforward for  $d = 1$  and  $d = D$ . For  $2 \leq d \leq D-1$ , two alternative Tucker-based formulations of (4.4) are

$$\begin{aligned} \hat{\mathcal{G}}_d^{hrl} &= \hat{\mathcal{G}}_d^{seq} \times_1 \mathbf{H}_{d-1}^{-1} \times_2 \mathbf{I}_{I_d} \times_3 \mathbf{H}_d^T, \\ \hat{\mathcal{G}}_d^{seq} &= \hat{\mathcal{G}}_d^{hrl} \times_1 \mathbf{H}_{d-1} \times_2 \mathbf{I}_{I_d} \times_3 \mathbf{H}_d^{-T}. \end{aligned}$$

Based on the two above relations, tensors  $\hat{\mathcal{G}}_d^{hrl}$  and  $\hat{\mathcal{G}}_d^{seq}$  have a multilinear  $(R_{d-1}, I_d, R_d)$ -rank. Since the TT-ranks correspond to the dimensions of the first and third modes of  $\hat{\mathcal{G}}_d^{hrl}$  or  $\hat{\mathcal{G}}_d^{seq}$ , the proof is completed. □

Intuitively, the TT-ranks are essentially related to the TT model, and in particular to the matrices given by (2.6), and not to the choice of the algorithm.

One may note that in the case where the tensor reshapings are of low rank only approximately, Oseledets in [105] has established a bound on the error of reconstruction for the TT-SVD algorithm. Unfortunately, establishing a similar bound for the TT-HSVD is not straightforward since we reshape and use both the left and right parts of the SVD at each step, leading to the manipulation of some non-orthonormal factors causing this issue. In [138, 46], the authors have proposed a good strategy to deal with the non-orthonormal factors that can be applied in the context of the TT-HSVD algorithm. However, we will show in the next section that even if the tensor is affected by noise, both algorithms, namely the TT-SVD and the TT-HSVD, can still have the same robustness when the TT-ranks are either assumed to be known or when they are estimated in the algorithms.

## 4.1.4 Computational complexity and simulation results

### 4.1.4.1 Numerical computation of the SVD

In this section, we compare the computational complexity of both algorithms using the truncated SVD. Note that TT-HSVD and TT-SVD involve the same number of SVDs. However, the TT-HSVD has the advantage of applying SVDs to matrices of smaller dimensions compared to the TT-SVD algorithm, which results in a lower computational complexity. The numerical stability of the TT-SVD and the TT-HSVD is relied to the well-known numerical rank determination in the SVD. We have essentially two scenarios of interest.

1. The true rank is a priori known. This is often realistic due to *a priori* knowledge on the physical model as for instance in wireless communication applications where the rank can be the number of path/user, in array processing for sky imaging where the rank is the number of most brining stars (known due to decade and decade of sky observation), .... For these important scenarios, the TT-SVD and the TT-HSVD algorithms provide an exact factorization of a high-order tensor into a collection of low-order tensors. This is mainly the framework of our contribution. The computation of the SVD is done based on two orthogonal iteration (OI) algorithms [53] executed in parallel. We recall that computing the matrix  $\mathbf{U}$  of size  $m \times r$  consists in [53] recursively computing the QR factorization of the  $m \times r$  matrix  $(\mathbf{A}\mathbf{A}^T)\mathbf{U}_{i-1} = \mathbf{U}_i\mathbf{R}$  with dominant complexity [53]  $O(r^2m)$ . In the same way, we can calculate matrix  $\mathbf{V}$  of size  $n \times r$ , using the QR decomposition of  $(\mathbf{A}^T\mathbf{A})\mathbf{V}_{i-1} = \mathbf{V}_i\mathbf{R}$  with dominant complexity [53]  $O(r^2n)$ . The singular values are automatically obtained with the calculation of  $\mathbf{U}$  and  $\mathbf{V}$  from the matrix  $\mathbf{R}$ . Considering the matrix multiplication cost, the overall SVD cost is evaluated at  $O(r^2(m+n) + rmn)$ .
2. The true rank is unknown. In this case, the problem is to find a robust estimation of the true rank, called the numerical rank. Considering a numerical rank larger than the true one can be problematic since the last columns of the basis matrices are pondered by near-zero singular values. Typically, the true rank is numerically estimated by searching a numerical gap toward the dominant singular values and the others. Many dedicated methods exist to find this gap [113, 143, 26, 83, 24]. In our work, the numerical rank is computed with the native routine `rank.m` of MatLab as the number of singular values that are larger than a tolerance. The tolerance is a function of the spacing of floating point (`eps`), the size and the norm of the matrix as for instance  $\max\{m, n\} \cdot \text{eps}(\|\mathbf{A}\|)$  where  $\text{eps}(x)$  is the distance from  $|x|$  to the next larger in magnitude floating point number of the same precision as  $x$ . It makes sense to use the bi-iteration algorithm based on the sequential computation of two OI(s). Unlike the OI algorithm, the singular-values are an output of the bi-iteration algorithm [53, 139]. The complexity cost per iteration of the bi-iteration algorithm is in the same order as a single OI.

Finally, the two schemes, namely the popular TT-SVD and our proposition called the TT-HSVD, inherit from the numerical robustness of the SVD. But, we think that the optimization of this important operation (the numerical rank estimation) is out of scope of our work.

#### 4.1.4.2 Complexity analysis and simulations

Considering a 4-order hypercubic ( $I$ ) tensor of TT-ranks equal to  $R$ , the complexity of the TT-SVD algorithm is given by  $O(RI^4) + O(R^2I^3) + O(R^2I^2)$  where,

1. the complexity of the first  $R$ -truncated SVD is

$$O(R^2(I^3 + I) + RI^4) \approx O(RI^4)$$

for large  $I$ .

2. The complexity of the second  $R$ -truncated SVD applied on  $\mathbf{X}_{(2)}$  is  $O(R^2(RI + I^2 + R^2I^3) + RI^3) \approx O(R^2I^3)$  if  $I \gg R$ .

3. The complexity of the last  $R$ -truncated SVD applied on  $\mathbf{X}_{(3)}$  is  $O(R^2(RI + I) + R^2I^2) \approx O(R^2I^2)$ .

Following the same reasoning, the complexity of the TT-HSVD algorithm for a mono-core architecture is given by  $O(RI^4) + O(R^2I^2)$  where,

1. The complexity of the first  $R$ -truncated SVD of  $\mathbf{X}_{(2)}$  is  $O(R^2(I^2 + I^2) + RI^2) = O(RI^4)$ .
2. The complexity of the  $R$ -truncated SVD of  $\mathbf{X}_{(1)}$  related to the left part of the tree is  $O(R^2(IR + I) + R^2I^2) \approx O(R^2I^2)$ .
3. The complexity of the  $R$ -truncated SVD of  $\mathbf{X}_{(3)}$  related to the right part of the tree is  $O(R^2(IR + I) + R^2I^2) \approx O(R^2I^2)$ .

The dominant complexity of the TT-HSVD is much lower than that of the TT-SVD due to the large term  $O(R^2I^3)$ .

Note that, for both algorithms, the term dominating the complexity corresponds to the first SVD step, which is applied to the largest matrix at the beginning of the process. For a  $D$ -order tensor, and by fixing  $I_d = I$  for all  $d$ , the dominant complexities of TT-SVD and TT-HSVD (in the balanced case), are respectively given by

$$\kappa_{seq} = O(RI^D) + O(R^2I^{(D-1)}) + O(R^2I^{(D-2)}) + \dots + O(R^2I^2), \quad (4.6)$$

and

$$\kappa_{hrl} = O(RI^D) + O(R^2I^{\frac{D}{2}}) + O(R^2I^{\frac{D}{4}}) + \dots + O(R^2I^2). \quad (4.7)$$

This means that the complexity of TT-SVD grows faster than that of the TT-HSVD algorithm as a function of the tensor order. Thus, TT-HSVD offers a significant advantage over the TT-SVD algorithm especially for high-order data tensors. This gain of complexity for TT-HSVD is due to the low storage cost of the considered matrices  $\mathbf{X}_{(d)}$  in the TT-HSVD compared to those of the TT-SVD. One may note that the storage cost of these matrices in TT-SVD and TT-HSVD is proportional respectively to the complexities in (4.6) and (4.7) up to a multiplication by  $R$ , since the complexity of a truncated SVD of a matrix is evaluated to the number of its entries multiplied by the rank. This means that the TT-HSVD has advantages in both the algorithmic complexity and the intermediate storage cost before recovering the TT-cores.

In Table 4.1, we compare the computation time of both algorithms using the “Tic-Toc” functions of MATLAB to decompose a  $D$ -th order tensor  $\mathcal{X}$ , for  $9 \leq I \leq 13$ .

To this end, we generate the tensor  $\mathcal{X}$  with  $I = I_d$  and  $R = R_i$ , for  $1 \leq d \leq D$ ,  $1 \leq i \leq D - 1$ , where  $R_d$  are the associated TT-ranks. The execution time is expressed in seconds (s). The simulations were performed on a personal computer equipped with an Intel(R) CORE(TM) i7-3687U CPU @ 2.10GHz processor and 8Gb RAM.

**Remark.** Note that the execution time of TT-HSVD is that of a sequential processing, where all the SVD steps are run in batch. This execution time could be significantly reduced [65] if a parallel processing architecture is used. However, the complexity estimation given in (4.7) remains valid when using a multicore structure for the decomposition.

In Table 4.1, we consider  $(D, R) = (8, 3)$ . Note that the results shown in this table are in agreement with the analytical complexity analysis, confirming that a significant complexity gain is achieved by TT-HSVD in comparison with TT-SVD,

TABLE 4.1: Comparison of the computation time of both algorithms ( $D = 8, R = 3$ ).

Tensor dimension	$I = 9$	$I = 10$	$I = 11$	$I = 12$	$I = 13$
TT-SVD algorithm	5.9	13.3	29	88.1	—
TT-HSVD algorithm	1.9	4	8.3	17.3	43.9
Gain	3.1	3.3	3.5	5.1	$\infty$

especially for very high-order tensors. In particular, for  $I = 13$ , the TT-SVD algorithm can not be executed by the computer, which returned an out of memory error. Note that the first unfolding matrix  $\mathbf{X}_{(1)}$  of the TT-SVD algorithm is of size  $13 \times 13^7$ , which requires a right singular vectors matrix of size  $3 \times 13^7$  for the truncated SVD. The size of this matrix is actually beyond the allowed storage capacity. Notice that the TT-HSVD algorithm has also the advantage of having a storage cost lower than the TT-SVD algorithm.

In Table 4.2, we evaluate the impact of the choice of the unfolding used as the starting point of the TT-HSVD, in terms of computational complexity. To this end, we choose different values of  $I_i$ , with  $(D, R) = (8, 4)$ . The two configurations used in the results of Figs. 4.2 (left) and 4.2 (right) are considered here. The results shown in this table correspond to the best computation time of the TT-HSVD algorithm.

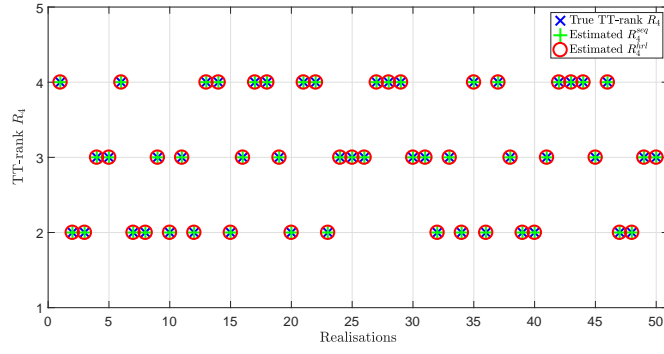
TABLE 4.2: Comparison of the computation time of TT-HSVD and TT-SVD.

Scenarios ( $I_1, \dots, I_8$ )	TT-HSVD (Fig. 4.2 (left))	TT-HSVD (Fig. 4.2 (right))	TT-SVD	Gain
(18, 36, 32, 16, 6, 6, 6, 6).	22.6	16.3	152.2	9.3
(22, 36, 32, 19, 6, 6, 6, 6).	45.1	23.7	993.7	42

It can be noted that choosing the most “square” unfolding matrix (i.e. the one with more balanced row and column dimensions) results in a lower overall computational complexity. The higher is the tensor order, more degrees of freedom are available for choosing the best unfolding to start the TT-HSVD algorithm.

In the following experiment, we generate an 8-order tensor  $\mathcal{X}$  of size  $I \times \dots \times I$  in the TT-format, with TT-ranks  $(R_1, \dots, R_7)$  chosen randomly between 2 and 4. This tensor is decomposed by means of the TT-SVD and TT-HSVD algorithms. We fix  $I = 4$  and observe the TT-ranks calculated by each algorithm. In Fig. 4.10, we plot the estimated ranks for both algorithms against the true TT-rank  $R_4$ . We can see that the TT-ranks calculated by both algorithms follow the true values. The same results are found for the other TT-ranks. We note here that for random realizations of the TT-ranks, we observe exactly the same ranks for both algorithms TT-SVD and TT-HSVD. This illustrates well the results on the equality between the TT-SVD ranks and the TT-HSVD ranks obtained previously.

In Table 4.3, we give the normalized reconstruction error of the estimated tensor for both the TT-SVD and the TT-HSVD, considering a tensor affected by an additive

FIGURE 4.10: The estimated ranks for the true TT-rank  $R_4$ .

Gaussian noise. The original tensor is an 8-order hypercubic tensor following a CPD of canonical rank  $R = 2$  with dimension  $I = 4$ . This is an interesting and realistic case regarding the Joint dDimensionality Reduction And Factor rEtrieval (JIRAFE) framework [158, 156] presented in Chapter 3, where the original tensor is a high-order CPD that is decomposed into a TT format to break its dimensionality before estimating its parameters. Two cases are considered in the following experiment, *i.e.*, when the original noisy tensor has a known TT-ranks equal to  $R$  and when the TT-ranks are unknown and are estimated at each step of the algorithm. The given errors are obtained by averaging the results over 1000 Monte-Carlo runs. One may note that for a wide range of SNR, either when the TT-ranks are estimated or are assumed to be known, both algorithms have the same robustness. This means that the TT-HSVD can be a better alternative for the TT-SVD algorithm in the JIRAFE framework for parameters estimation of high-order tensor.

TABLE 4.3: Comparison of the robustness of TT-HSVD and TT-SVD.

SNR (dB)	Unknown rank		Known rank	
	TT-HSVD	TT-SVD	TT-HSVD	TT-SVD
30	$1.28 \times 10^{-6}$	$1.27 \times 10^{-6}$	$1.28 \times 10^{-6}$	$1.27 \times 10^{-6}$
25	$4.02 \times 10^{-6}$	$4.01 \times 10^{-6}$	$4.06 \times 10^{-6}$	$4.05 \times 10^{-6}$
20	$1.3 \times 10^{-5}$	$1.3 \times 10^{-5}$	$1.29 \times 10^{-5}$	$1.28 \times 10^{-5}$
15	$4.06 \times 10^{-5}$	$4.05 \times 10^{-5}$	$4.06 \times 10^{-5}$	$4.06 \times 10^{-5}$
10	$1.3 \times 10^{-4}$	$1.3 \times 10^{-4}$	$1.3 \times 10^{-4}$	$1.3 \times 10^{-4}$
5	$4.07 \times 10^{-4}$	$4.1 \times 10^{-4}$	$4.1 \times 10^{-4}$	$4.2 \times 10^{-4}$
0	$1.3 \times 10^{-3}$	$1.3 \times 10^{-3}$	$1.3 \times 10^{-3}$	$1.4 \times 10^{-3}$
-5	$4 \times 10^{-3}$	$4.7 \times 10^{-3}$	$4.4 \times 10^{-3}$	$5.7 \times 10^{-3}$
-10	$1.36 \times 10^{-2}$	$2.11 \times 10^{-2}$	$1.71 \times 10^{-2}$	$2.95 \times 10^{-2}$

#### 4.1.5 Discussion

In this chapter, we have proposed the TT-HSVD algorithm, a hierarchical algorithm for Tensor Train decomposition. It is a new algorithm that allows to recover simultaneously/hierarchically the TT-core tensors and their TT-ranks. A new graph-based representation using patterns has also been proposed to simplify and make more illustrative the algorithmic representation of both TT-SVD and TT-HSVD algorithms.

Our analysis has shown that the TT-SVD and TT-HSVD algorithms estimate same TT-ranks and same TT-core tensors up to specific bases, with significant time savings for the TT-HSVD algorithm. Perspectives for future work include the study of the combination of the tensor modes in a random way, and the consequences of this modification on the estimation and the complexity of the algorithm, and evaluating the performance of the TT-HSVD algorithm for applications like tensor completion [81], blind source separation [13] and fast SVD of large scale matrices [91]. In the next section, TT-HSVD will be applied in the context of MIMO relay systems.



## 4.2 TT-based applications

### 4.2.1 Introduction

The use of relay stations in MIMO communication systems has shown to have a great potential to enhance coverage and increase system capacity [88, 96, 43, 100]. In the case of one-way two-hop MIMO relay systems, the communication can be divided into two phases. In the first one, the source node transmits the symbols to the relay. In the second one, the relay amplifies and forwards the signals to the destination node, meanwhile the source stays silent. To achieve the predictable gains of cooperative diversity, an accurate knowledge of channels associated with the multiple hops involved in the communication is required for the design of smart antenna schemes, such as beamforming or precoding [121].

In relay-assisted MIMO communication systems, the instantaneous channel for each hop is usually estimated by means of training sequences transmitted by the source and relay nodes during successive transmission phases. The works [94, 120] proposed training sequence based schemes to estimate the individual channel matrices for two-hop MIMO relaying systems. The first one [94] relies on an SVD-based solution, while the second [120] is based on a parallel factor (PARAFAC) modeling of the received signals at the destination. By aiming at a joint channel and symbol estimation in a semi-blind fashion, tensor-based receivers have been proposed in several works [150, 49] for two-hop MIMO relaying systems. Of particular interest to this work is the approach of [49], which is based on tensor coding-and-forwarding (TCF) scheme by means of a tensor space-time coding applied at both the source and relay nodes. More recently, generalizations to multihop systems [51] and to two-way systems [52] have also been proposed. All these works have assumed that the propagation channels are frequency-flat, which is the case of narrowband communication systems.

In this work, we assume a more general and practical scenario where the MIMO relaying system is operating in frequency-selective fading environment. As will be shown later, when generalizing the scheme of [49] to the wideband communication scenario, the tensor modeling of the received signals involves the estimation of larger quantities compared to the narrowband case. These quantities are represented by third-order channel tensors and one symbol matrix, the third dimension being associated to the frequency domain. By resorting to multicarrier modulation using orthogonal frequency division multiplexing (OFDM), we propose a new receiver design for a TCF MIMO-OFDM relaying system that is capable to solve the joint channel and symbol estimation problem. The proposed receiver fits the resulting 6-order received signal tensor to a tensor train decomposition (TTD), where the knowledge of the tensor coding structure is used to ensure identifiability of the channel tensors and the symbol matrix. It is important to note that our methodology is able to manage the case of any number of relays. To the best of our knowledge, this is the first work where the TTD approach is used to design a semi-blind receiver for a MIMO communication system.

### 4.2.2 System Model

In this work, we consider a MIMO-OFDM relaying system, where the communication is divided into two hops<sup>1</sup> as illustrated in Fig. 4.11. The 6 dimensions/diversities of the system are associated with time, source code, frequency (during the first hop),

<sup>1</sup>A generalization with more than two hops can be easily derived from the contribution of our work.

relay code, frequency (during the second hop), and space. At both source and relay nodes, we consider a tensor space-time coding (TSTC) scheme, following the idea of [49]. Note, however, that the system model considered in this work is a generalization of that of [49] to a MIMO-OFDM system. Due to the added frequency dimensions at both the source  $\rightarrow$  relay (SR), and relay  $\rightarrow$  destination (RD) channels, the SR and RD channels are modeled as third-order tensors and denoted as  $\mathcal{H}^{(SR)}$  and  $\mathcal{H}^{(RD)}$ . Let  $\mathcal{X}$  be the 6-order tensor, of dimensions  $M_D \times F_1 \times K \times F_2 \times P \times N$ ,

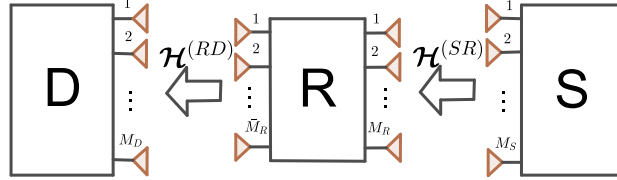


FIGURE 4.11: One-way two-hop MIMO-OFDM relay system illustration.

representing the received signals at the destination. In a free-noise scenario, this tensor  $\mathcal{X}$  can be expressed as follows:

$$[\mathcal{X}]_{m_D, f_1, k, f_2, p, n} = \sum_{\bar{m}_R=1}^{\bar{M}_R} \sum_{m_R=1}^{M_R} \sum_{m_S=1}^{M_S} \sum_{r=1}^R [\mathcal{H}^{(RD)}]_{m_D, f_1, \bar{m}_R} [\mathcal{C}^{(R)}]_{\bar{m}_R, k, m_R} [\mathcal{H}^{(SR)}]_{m_R, f_2, m_S} [\mathcal{C}^{(S)}]_{m_S, p, r} [\mathbf{S}]_{r, n}. \quad (4.8)$$

We provide in Table 4.4 the description and dimensions of tensors used in eq. (4.8).

TABLE 4.4: Description of tensors used in eq. (4.8).

Symbols	Description	Dimensions
$\mathcal{H}^{(RD)}$	RD channel tensor	$M_D \times F_1 \times \bar{M}_R$
$\mathcal{C}^{(R)}$	Coding tensor at the relay	$\bar{M}_R \times K \times M_R$
$\mathcal{H}^{(SR)}$	SR channel tensor	$M_R \times F_2 \times M_S$
$\mathcal{C}^{(S)}$	Coding tensor at the source	$M_S \times P \times R$
$\mathbf{S}$	Transmitted symbols matrix	$R \times N$

It is worth mentioning that the tensor  $\mathcal{X}$  results from the transmission of  $R$  data streams, each composed of  $N$  symbols, during  $N$  different time-blocks. During each block  $n$ , each antenna  $m_S$  transmits a combination of  $R$  information symbols  $[\mathbf{S}]_{r, n}$  to the relay after a space-time coding by means of the coding tensor  $\mathcal{C}^{(S)}$  and through the channel  $\mathcal{H}^{(SR)}$ . The signals received at the relay are encoded by means of the coding tensor  $\mathcal{C}^{(R)}$  and then transmitted to the destination through the channel  $\mathcal{H}^{(RD)}$ . The received signals satisfy eq. (4.8). Different assumptions are considered for the model of eq. (4.8):

- the coding tensors are constant during the whole transmission
- the channels are quasi-static, *i.e.* do not change, within a transmission cycle
- the coding tensors and the structural parameters (tensor dimensions) are known

Note that  $M_D, \bar{M}_R, M_R, M_S$ , and  $R$  refer to the number of antennas at the destination, transmitting antennas at the relay, receiving antennas at the relay, antennas at the source, and the data streams, respectively.

### 4.2.3 Tensor Train decomposition (TTD)

#### 4.2.3.1 Definition of the TTD

**Definition 15.** Let  $\{R_1, \dots, R_{Q-1}\}$  be the TT-ranks with bounding conditions  $R_0 = R_Q = 1$ . A  $Q$ -order tensor of size  $N_1 \times \dots \times N_Q$  admits a decomposition into a train of low-order tensors if

$$\mathcal{X} = \mathbf{G}_1 \times_2^1 \mathcal{G}_2 \times_3^1 \mathcal{G}_3 \times_4^1 \dots \times_{Q-1}^1 \mathcal{G}_{Q-1} \times_Q^1 \mathbf{G}_Q, \quad (4.9)$$

where the TT-cores  $\mathbf{G}_1, \mathcal{G}_q$ , and  $\mathbf{G}_Q$  are, respectively, of dimensions  $N_1 \times R_1, R_{q-1} \times N_q \times R_q$ , and  $R_{Q-1} \times N_Q$ , for  $2 \leq q \leq Q-1$ , with  $\text{rank}(\mathbf{G}_1) = R_1$ ,  $\text{rank}(\mathbf{G}_Q) = R_{Q-1}$ ,  $\text{rank}(\text{unfold}_1 \mathcal{G}_q) = R_{q-1}$ , and  $\text{rank}(\text{unfold}_3 \mathcal{G}_q) = R_q$ .

Using the contraction product, the model in eq. (4.8) can be written in a compact form such that:

$$\mathcal{X} = \mathbf{I}_{M_D} \times_2^1 \mathcal{H}^{(RD)} \times_3^1 \mathcal{C}^{(R)} \times_4^1 \mathcal{H}^{(SR)} \times_5^1 \mathcal{C}^{(S)} \times_6^1 \mathbf{S}, \quad (4.10)$$

to match the definition of the TTD given in eq. (4.9).

#### 4.2.3.2 The TTD multiplicative ambiguities

As shown in eq. (4.10), the considered MIMO system is modeled as a 6-order TTD with a priori known TT-ranks  $\{M_D, \bar{M}_R, M_R, M_S, R\}$ . The multiplicative ambiguities in the TTD correspond to post- and pre-multiplications by nonsingular matrices, *i.e.*, we can replace two successive TT-cores  $\mathcal{G}_q$  and  $\mathcal{G}_{q+1}$  in eq. (4.9), respectively, by  $\mathcal{G}'_q$  and  $\mathcal{G}'_{q+1}$  such that

$$\begin{aligned} \mathcal{G}'_q &= \mathcal{G}_q \times_3^1 \mathbf{U}_q^{-1}, \\ \mathcal{G}'_{q+1} &= \mathbf{U}_q \times_2^1 \mathcal{G}_{q+1}, \end{aligned}$$

to recover the same tensor  $\mathcal{X}$  of eq. (4.9), where  $\mathbf{U}_q$  is a nonsingular matrix of size  $R_q \times R_q$ .

Applying the TT-HSVD algorithm to tensor  $\mathcal{X}$  allows to recover the original TT-cores  $\mathcal{G}_q$  up to these nonsingular matrices, called  $\mathbf{U}_q$  in the sequel. The TT-HSVD algorithm is based on several truncated SVD(s). Knowing the TT-ranks and in the noise-free case, the TT-HSVD algorithm recovers exactly the TT-cores. In the context of the TT-HSVD algorithm these nonsingular matrices correspond to transformation (*change-of-basis*) matrices due to the extraction of the dominant singular subspaces using the SVD.

### 4.2.4 TT-based receiver

#### 4.2.4.1 TT-cores structure

The following theorem gives the structure of the TT-cores when the TT-HSVD algorithm is applied to tensor  $\mathcal{X}$  in eq. (4.10).

**Theorem 14.** Consider the 6-order tensor  $\mathcal{X}$  defined in eq. (4.10). In the noise-free scenario and knowing a priori the TT-ranks, the structure of the recovered TT-cores is given by

$$\begin{aligned}\mathbf{G}_1 &= \mathbf{U}_1^{-1}, \\ \mathcal{G}_2 &= \mathcal{H}^{(RD)} \times_1 \mathbf{U}_1 \times_3 \mathbf{U}_2^{-T}, \\ \mathcal{G}_3 &= \mathcal{C}^{(R)} \times_1 \mathbf{U}_2 \times_3 \mathbf{U}_3^{-T}, \\ \mathcal{G}_4 &= \mathcal{H}^{(SR)} \times_1 \mathbf{U}_3 \times_3 \mathbf{U}_4^{-T}, \\ \mathcal{G}_5 &= \mathcal{C}^{(S)} \times_1 \mathbf{U}_4 \times_3 \mathbf{U}_5^{-T}, \\ \mathbf{G}_6 &= \mathbf{U}_5 \mathbf{S},\end{aligned}$$

where  $\mathbf{U}_1, \dots, \mathbf{U}_5$  are square nonsingular transformation matrices of ranks  $M_D, \bar{M}_R, M_R, M_S$  and  $R$ , respectively, corresponding to the TT-ranks of the model.

It is worth noting that the above theorem is the result of the application of the multiplicative ambiguities of the TTD given in eq. (4.10). This means that the cores of eq. (4.10) are estimated up to post- and pre-multiplications by nonsingular matrices as shown in Section 4.2.3.2. In Theorem 14, the TT-cores follow a Tucker decomposition (TD) with the following equivalence:

$$\mathcal{T}' = \mathbf{A}_1 \times_2^1 \mathcal{T} \times_3^1 \mathbf{A}_2 = \mathcal{T} \times_1 \mathbf{A}_1 \times_3 \mathbf{A}_2^T.$$

The TD formalism helps us to introduce our estimation scheme in the next section.

#### 4.2.4.2 Estimation algorithm: TT-MRS

In this section, we propose an estimation scheme of tensor channels  $\mathcal{H}^{(RD)}, \mathcal{H}^{(SR)}$ , and of the transmitted symbols matrix  $\mathbf{S}$ , assuming the knowledge of the code tensors  $\mathcal{C}^{(R)}$  and  $\mathcal{C}^{(S)}$ . It is a TT-based semi-blind receiver for MIMO relay systems (TT-MRS). The idea of Algorithm 11 is to eliminate the latent ambiguity matrices  $\mathbf{U}_1, \dots, \mathbf{U}_5$  of Theorem 14 using the knowledge of tensors  $\mathcal{C}^{(R)}$  and  $\mathcal{C}^{(S)}$ . The proposed algorithm is a 3 steps scheme that is based on:

- the TT-HSVD algorithm to decompose  $\mathcal{X}$  into the TTD
- the estimation of the transformation matrices using  $\mathcal{C}^{(R)}$  and  $\mathcal{C}^{(S)}$
- channels and symbols estimation using the TT-cores of step 1 and the ambiguity matrices of step 2.

It is worth mentioning that the processing of step 2 (same for step 3) can be done in a parallel way. Note that tensors  $\mathcal{C}^{(R)}$  and  $\mathcal{C}^{(S)}$  represent the core tensors of the respective TDs of  $\mathcal{G}_3$  and  $\mathcal{G}_5$ , as shown in Theorem 14, and we have

$$\text{unfold}_1 \mathcal{G}_3 = \mathbf{U}_2 \cdot \text{unfold}_1 \mathcal{C}^{(R)} \cdot (\mathbf{U}_3^{-T} \otimes \mathbf{I}_K)^T, \quad (4.11)$$

$$\text{unfold}_2 \mathcal{G}_3 = \mathbf{I}_K \cdot \text{unfold}_2 \mathcal{C}^{(R)} \cdot (\mathbf{U}_3^{-T} \otimes \mathbf{U}_2)^T, \quad (4.12)$$

$$\text{unfold}_3 \mathcal{G}_3 = \mathbf{U}_3^{-T} \cdot \text{unfold}_3 \mathcal{C}^{(R)} \cdot (\mathbf{I}_K \otimes \mathbf{U}_2)^T. \quad (4.13)$$

Regarding eq. (4.11) and eq. (4.13), we can notice that recovering matrices  $\mathbf{U}_2$  and  $\mathbf{U}_3$  (the same reasoning is valid for  $\mathbf{U}_4$  and  $\mathbf{U}_5$ ), using  $\mathcal{C}^{(R)}$  and  $\mathcal{G}_3$ , can be done in a general case using an iterative Tucker-ALS algorithm [85].

---

**Algorithm 11** TT-MRS algorithm
 

---

**Input:** 6-order tensor  $\mathcal{X}$ ,  $\mathcal{C}^{(R)}$  and  $\mathcal{C}^{(S)}$  defined in eq. (4.10).

**Output:**  $\hat{\mathcal{H}}^{(RD)}$ ,  $\hat{\mathcal{H}}^{(SR)}$ , and  $\hat{\mathbf{S}}$ .

1: TTD: (using the TT-HSVD algorithm)

$$\mathcal{X} = \hat{\mathbf{G}}_1 \times_2^1 \hat{\mathcal{G}}_2 \times_3^1 \hat{\mathcal{G}}_3 \times_4^1 \hat{\mathcal{G}}_4 \times_5^1 \hat{\mathcal{G}}_5 \times_6^1 \hat{\mathbf{G}}_6.$$

2: Transformation matrices estimation:

$$\hat{\mathbf{U}}_1^{-1} = \hat{\mathbf{G}}_1.$$

$$[\hat{\mathbf{U}}_2, \hat{\mathbf{U}}_3^{-T}] = \text{Tucker-ALS}(\hat{\mathcal{G}}_3, \mathcal{C}^{(R)}).$$

$$[\hat{\mathbf{U}}_4, \hat{\mathbf{U}}_5^{-T}] = \text{Tucker-ALS}(\hat{\mathcal{G}}_5, \mathcal{C}^{(S)}).$$

3: Channels and symbols estimation:

$$\hat{\mathcal{H}}^{(RD)} = \hat{\mathcal{G}}_2 \times_1 \hat{\mathbf{U}}_1^{-1} \times_3 \hat{\mathbf{U}}_2^T$$

.

$$\hat{\mathcal{H}}^{(SR)} = \hat{\mathcal{G}}_4 \times_1 \hat{\mathbf{U}}_3^{-1} \times_3 \hat{\mathbf{U}}_4^T$$

.

$$\hat{\mathbf{S}} = \hat{\mathbf{U}}_5^{-1} \hat{\mathbf{G}}_6.$$


---

One may note that  $\mathbf{U}_2$  and  $\mathbf{U}_3$  involved in eq. (4.12) are estimated up to a scalar multiplication, since

$$\mathbf{A} \otimes \mathbf{B} = \alpha \cdot \mathbf{A} \otimes \left(\frac{1}{\alpha}\right) \cdot \mathbf{B}.$$

Applying this property on the proposed scheme, and taking into account that matrices  $\mathbf{U}_2, \dots, \mathbf{U}_5$  are estimated up to scalar multiplication, the outputs of Algorithm 11 are then expressed as

$$\hat{\mathcal{H}}^{(RD)} = \alpha_1 \cdot \mathcal{H}^{(RD)},$$

$$\hat{\mathcal{H}}^{(SR)} = \alpha_2 \cdot \mathcal{H}^{(SR)},$$

$$\hat{\mathbf{S}} = \left(\frac{1}{\alpha_1 \alpha_2}\right) \cdot \mathbf{S},$$

this means that the parameters of interest are estimated up to a scalar multiplication. To resolve this ambiguity, two assumptions are possible:

- the knowledge of the first-order statistic (the mean) of channels  $\mathcal{H}^{(RD)}$  and  $\mathcal{H}^{(SR)}$  can be assumed as in [149].

- assume the knowledge of one entry of tensors  $\mathcal{H}^{(RD)}$  and  $\mathcal{H}^{(SR)}$  as assumed in [49].

#### 4.2.5 Simulation Results

In this section, we evaluate the performance of the proposed receiver by means of numerical computer simulations for various system configurations. The aim of the following experiments is twofold. First, we want to validate the TT modeling for the proposed MIMO relay system, and show that the proposed method allows a joint channels and symbols estimation. Second, we want to evaluate the influence of the system configurations on the quality of estimation, in particular, the choice and interest of the new introduced parameter  $F$ . The 6-order MIMO relay system is generated with random channel and coding tensors whose elements are drawn from a Gaussian distribution with zero mean and unit variance. The transmitted symbols are uniform random variables from a 4-QAM constellation. The additive noise tensors at relay and destination, noted respectively  $\mathcal{N}^{(R)}$  and  $\mathcal{N}^{(D)}$ , are assumed to be composed of elements which are zero-mean Gaussian variables, with a unit variance. The final noise, noted  $\mathcal{N}^{(SRD)}$  corresponding to  $\mathcal{N}^{(D)}$  at destination, and  $\mathcal{N}^{(R)}$  at relay filtered by the relay coding tensor  $\mathcal{C}^{(R)}$  and the channel  $\mathcal{H}^{(RD)}$  is expressed as

$$\mathcal{N}^{(SRD)} = \mathcal{H}^{(RD)} \times_3^1 \mathcal{C}^{(R)} \times_4^1 \mathcal{N}^{(R)} + \mathcal{N}^{(D)},$$

where  $\mathcal{N}^{(R)}$  and  $\mathcal{N}^{(D)}$  are respectively of size  $M_R \times F_1 \times P \times N$  and  $M_D \times F_1 \times K \times F_2 \times P \times N$ . The depicted NMSE are obtained by averaging the NSE over  $10^4$  independent Monte Carlo runs, with

$$\text{NSE} = \frac{\|\hat{\mathcal{X}} - \mathcal{X}\|_F^2}{\|\mathcal{X}\|_F^2},$$

where  $\mathcal{X}$  and  $\hat{\mathcal{X}}$  denote, respectively, the received and reconstructed signals tensors.

Fig. 4.12 shows the NMSE of  $\mathcal{X}$  after the TTD using the TT-HSVD algorithm (1st step of Algo. 11), *i.e.*, the reconstruction error for the TTD of the received signals at the destination, when  $F_1 = F_2 = K = P = N = D_{TT}$  and  $M_D = \bar{M}_R = M_R = M_S = R = R_{TT}$ . This shows how the TT modeling fits successfully the considered MIMO relay system. Moreover, it can be concluded that the bigger is the dimension  $D_{TT}$  for a fixed TT-rank  $R_{TT}$ , the better is the estimation. In the opposite, when  $R_{TT}$  grows for a fixed  $D_{TT}$ , the estimation is degrading. Indeed, increasing  $D_{TT}$  implies higher diversity gains due to spreading across a higher number of subcarriers and time slots. On the other hand, increasing  $R_{TT}$  corresponds to a higher number of parameters (channel and symbols) to be estimated at the receiver. In Fig. 4.13, we plot the NMSE of the estimation of  $\mathcal{H}^{(RD)}$  by fixing  $F_2 = K = P = R = 4$ ,  $N = 10$  and  $M_D = \bar{M}_R = M_R = M_S = 2$ , and varying the parameter  $F_1$ . This result shows the correct estimation of the channel tensors using Algo. 11. Furthermore, it shows that the canal estimation becomes more difficult when several frequencies  $F_1$  are considered.

Finally, Fig. 4.14 shows the symbol error rate (SER) of the estimation of the symbols  $\mathbf{S}$  as a function of the SNR. The system configuration is as follows,  $F_1 = F_2 = K = P = 4$ ,  $N = 10$  and  $M_D = \bar{M}_R = M_R = M_S = 2$ . The conclusions of this

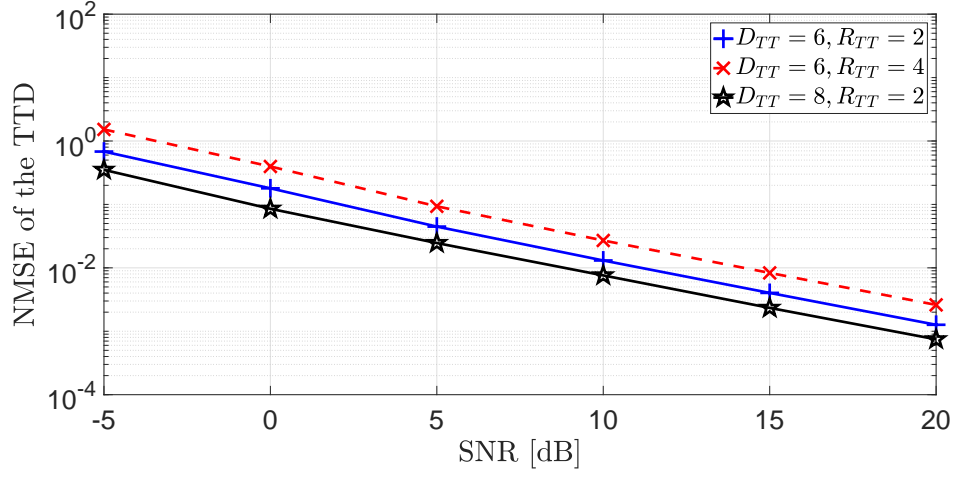
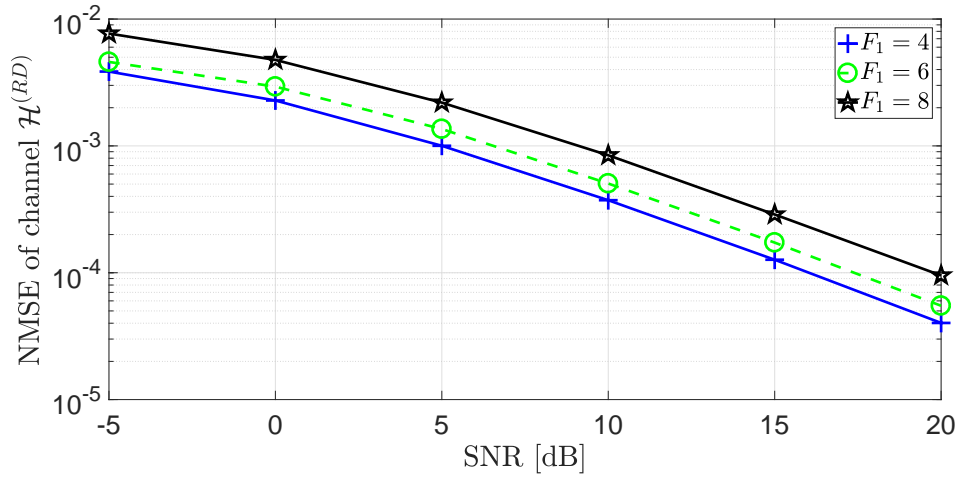


FIGURE 4.12: NMSE vs SNR in dB with the TTD.

FIGURE 4.13: NMSE of  $\mathcal{H}^{(RD)}$  for the proposed algorithm.

experiment join the previous one in the sense that the symbols matrix  $\mathbf{S}$  is correctly estimated using Algo. 11, which means that this latter allows a joint channels and symbols estimation. In addition, we can see the influence of the parameter  $R$ , one may note that the estimation becomes a difficult task when there is more symbols to transmit.

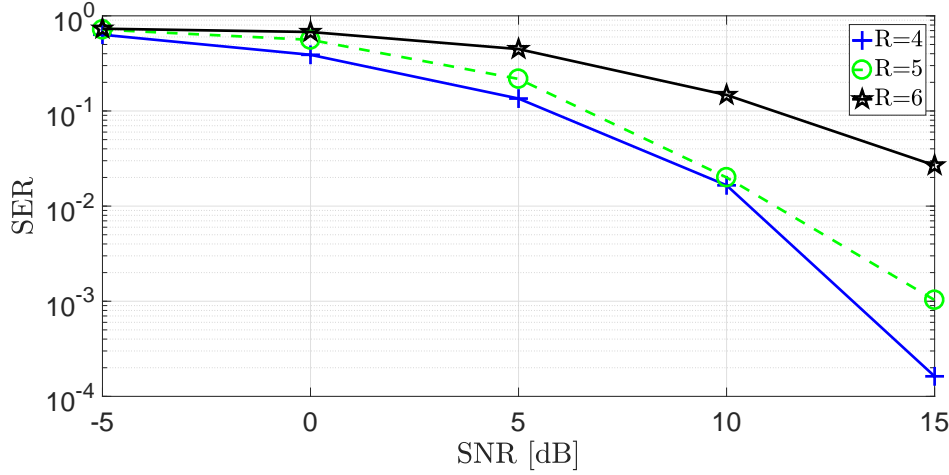


FIGURE 4.14: NMSE of symbols  $\mathbf{S}$  for the proposed algorithm.

#### 4.2.6 Discussion

A new TTD modeling approach has been proposed for MIMO-OFDM relay systems to jointly estimate the channels and the information symbols. Our approach generalizes the NTD based system of [49] by considering the case of an OFDM relay system. A new semi-blind receiver which uses a closed-form TTD algorithm and Tucker-ALS algorithms, and which relies on some TTD ambiguity results, had also been proposed. To the best of our knowledge, it does not exist competitive algorithm for solve this problem. The effectiveness of the proposed receiver is demonstrated by means of Monte Carlo simulations. Some extensions of this work include a generalization to TCF MIMO-OFDM with multiple relays.





## Chapter 5

# Conclusion and prospects

The work described in this PhD thesis deals with the curse of dimensionality and the challenge was to reformulate a high-order tensor as a set of graph connected low-order tensors. In particular, the goal was to have new tensor processing methods adapted to big data tensors, having the advantages of being robuste, scalable with a parallel computing. Among the all the existing tensor networks, the tensor train, due to its simplicity, has shown to be a good candidate to mitigate this issue. To address this problem, the tensor train decomposition is exploited in this work to derive new equivalence results between usual tensor decompositions and the TTD, with the aim to propose a new optimisation strategies. Indeed, a new framework called JIRAFE, for Joint dImensionality Reduction And Factor rEtrieval, is proposed. It is a two-step scheme that consists in, firstly, split the high-order tensor into a collection of graph-connected core tensors of lower order, and secondly, in the factor retrieval step thanks to an ad-hoc optimization strategy. JIRAFE allows to split the initial high-order optimization problem into a sum of low dimensionality optimization problems. This scheme shows its efficiency by allowing flexible parameters estimation strategies, stable algorithms (*i.e.*, insensitive to ill-converging problems), less computational and storage complexity and better robustness to noise compared to other state-of-art methods.

Moreover, by using the tensor networks graphical formalism, the tensor train allows then to visualize high-order tensors in a simple, rigorous and intuitive representation.

Also, to deal with the high computational complexity of the state-of-the-art TTD algorithm, namely the TT-SVD, we have proposed a new hierarchical framework, namely the TT-HSVD, which allows to simultaneously recover the TT-cores and their TT-ranks in a hierarchical way and which has a less complexity compared to the TT-SVD. This algorithm is well adapted to the big data tensors and is a good alternative to the TT-SVD algorithm to deal with the curse of dimensionality issue.

These two methodological parts have had applications in the context of multidimensional spectral analysis and relay telecommunication systems. The different simulation results showed the JIRAFE and TT-HSVD methods interests in the desired properties mentioned before.

Among the possible future prospects which naturally follow this work, we can mention the following points:

- Regarding the JIRAFE framework and from a fundamental perspective, the equivalence TT-CPD needs to be studied when the formulated full column rank assumption is violated, *i.e.*, when the canonical rank is greater than several factors mode dimensions.

- In the same sense, a more general case of factors with linear dependencies can be considered. The TT-cores structure, the TT-ranks and the uniqueness conditions in that case need to be studied.
- Applications of JIRAFE for the PARALIND can be considered in future works, as for example in the estimation of the probability density function of multidimensional flow cytometry data [20], where linear dependencies may exist.
- Regarding the TT-HSVD, a study of the combination of the tensor modes in a random way may be interesting, showing the consequences of this modification in the estimation and the complexity of the algorithm.
- Also, replacing the TT-SVD algorithm by the TT-HSVD one in the JIRAFE framework, and apply this methodology on estimation problems, especially and in particular for applications where only some factors are required as in clustering. In this case joining the hierarchical framework to the JIRAFE framework may be interesting.
- From an application perspective, an extension of the TT-based receiver for relay systems can include a generalization to MIMO-OFDM systems with multiple relays, where the TT-HSVD algorithm can be of a bigger interest.
- An application of the TTD in the context of coupled matrix tensor factorization [11] can also be investigated.

## Appendix A

# Proof of Theorem 2

In this constructive proof, the TT-SVD algorithm is applied “step by step” to a TD presented in (2.4), under the two assumptions that

- all the factors  $\mathbf{F}_q$  have full column rank, and
- $\mathbf{C}_{(q)} = \text{reshape}(\mathcal{C}; \prod_{i=1}^q T_i, \prod_{i=q+1}^Q T_i)$  has full rank.

The first unfolding  $\mathbf{X}_{(1)}$  of size  $N_1 \times (N_2 \cdots N_Q)$ , using Lemma 3, is given by:

$$\begin{aligned} \mathbf{X}_{(1)} &= \text{reshape}(\mathcal{X}; N_1, \prod_{q=2}^Q N_q) \\ &\stackrel{\Delta}{=} \mathbf{F}_1 \mathbf{C}_{(1)} (\mathbf{F}_Q \otimes \cdots \otimes \mathbf{F}_2)^T \\ &\stackrel{\text{SVD}}{=} \mathbf{U}^{(1)} \mathbf{V}^{(1)}. \end{aligned}$$

Note that  $\text{rank}(\mathbf{X}_{(1)}) = \text{rank}(\mathbf{F}_1) = R_1 = T_1$  ( $R_1$  is the first TT-rank), which means that  $\mathbf{U}^{(1)}$  is of size  $N_1 \times R_1$ , and  $\mathbf{V}^{(1)}$  is of size  $R_1 \times (N_2 \cdots N_Q)$ . It exists a  $R_1 \times R_1$  nonsingular *change-of-basis* matrix  $\mathbf{M}_1$  that verifies

$$\mathbf{U}^{(1)} = \mathbf{F}_1 \mathbf{M}_1^{-1} \quad (\text{A.1})$$

$$\mathbf{V}^{(1)} = \mathbf{M}_1 \mathbf{C}_{(1)} (\mathbf{F}_Q \otimes \cdots \otimes \mathbf{F}_2)^T \quad (\text{A.2})$$

Following the methodology of the TT-SVD algorithm, the first TT-core  $\mathbf{G}_1$  is given by eq. (A.1), i.e.,

$$\mathbf{G}_1 = \mathbf{U}^{(1)} = \mathbf{F}_1 \mathbf{M}_1^{-1}. \quad (\text{A.3})$$

Applying the same methodology to a reshaped version, denoted by  $\mathbf{V}_{(2)}^{(1)}$ , of  $\mathbf{V}^{(1)}$ , of size  $(R_1 N_2) \times (N_3 \cdots N_Q)$ , provides from eq. (A.2)

$$\begin{aligned} \mathbf{V}_{(2)}^{(1)} &= \text{reshape}(\mathbf{V}^{(1)}; R_1 N_2, \prod_{q=3}^Q N_q) \\ &\stackrel{\Delta}{=} (\mathbf{F}_2 \otimes \mathbf{M}_1) \mathbf{C}_{(2)} (\mathbf{F}_Q \otimes \cdots \otimes \mathbf{F}_3)^T \\ &\stackrel{\text{SVD}}{=} \mathbf{U}^{(2)} \mathbf{V}^{(2)} \end{aligned} \quad (\text{A.4})$$

Once again, we have  $\text{rank}(\mathbf{V}_{(2)}^{(1)}) = \text{rank}(\mathbf{F}_2 \otimes \mathbf{M}_1) = R_2 = T_1 T_2$ . In addition,  $\mathbf{U}^{(2)}$  of size  $(R_1 N_2) \times R_2$ , and  $\mathbf{V}^{(2)}$  of size  $R_2 \times (N_3 \cdots N_Q)$  are defined by

$$\begin{aligned}\mathbf{U}^{(2)} &= (\mathbf{F}_2 \otimes \mathbf{M}_1) \mathbf{M}_2^{-1} \\ \mathbf{V}^{(2)} &= \mathbf{M}_2 \mathcal{C}_{(2)} (\mathbf{F}_Q \otimes \cdots \otimes \mathbf{F}_3)^T\end{aligned}\tag{A.5}$$

where  $\mathbf{M}_2$  is a  $R_2 \times R_2$  nonsingular *change-of-basis* matrix. From eq.(A.5), the 2nd TT-core  $\mathcal{G}_2$  is expressed as

$$\begin{aligned}\mathcal{G}_2 &= \text{reshape}(\mathbf{U}^{(2)}; R_1, N_2, R_2) \\ &= \mathcal{T}_2 \times_1 \mathbf{M}_1 \times_2 \mathbf{F}_2 \times_3 \mathbf{M}_2^{-T},\end{aligned}\tag{A.6}$$

where  $\mathcal{T}_2 = \text{reshape}(\mathbf{I}_{R_2}; R_1, T_2, R_2)$ .

This strategy can be continued to obtain the  $q$ -th TT-cores for  $3 \leq q \leq \bar{q} - 1$  since  $\mathcal{C}_{(q)}$ , defined in Theorem 1, has always a full row rank and can always be absorbed in  $\mathbf{V}^{(q)}$ , when applying the SVD. Let us now see what happens when we express  $\mathbf{V}_{(2)}^{(\bar{q}-1)}$  of size  $(R_{\bar{q}-1}N_{\bar{q}}) \times (N_{\bar{q}+1} \cdots N_Q)$ . From (A.4), we obtain

$$\begin{aligned}\mathbf{V}_{(2)}^{(\bar{q}-1)} &= \text{reshape}(\mathbf{V}^{(\bar{q}-1)}; R_{\bar{q}-1}N_{\bar{q}}, \prod_{q=\bar{q}+1}^Q N_q) \\ &\triangleq (\mathbf{F}_{\bar{q}} \otimes \mathbf{M}_{\bar{q}-1}) \mathcal{C}_{(\bar{q})} (\mathbf{F}_Q \otimes \cdots \otimes \mathbf{F}_{\bar{q}+1})^T \\ &\stackrel{\text{SVD}}{=} \mathbf{U}^{(\bar{q})} \mathbf{V}^{(\bar{q})}\end{aligned}$$

Note that here we have  $\text{rank}(\mathbf{V}_{(2)}^{(\bar{q}-1)}) = \text{rank}((\mathbf{F}_{\bar{q}} \otimes \mathbf{M}_{\bar{q}-1}) \mathcal{C}_{(\bar{q})}) = R_{\bar{q}} = T_{\bar{q}+1} \cdots T_Q$ , since  $\mathcal{C}_{(\bar{q})}$  has a full column rank, and we can write

$$\mathbf{U}^{(\bar{q})} = (\mathbf{F}_{\bar{q}} \otimes \mathbf{M}_{\bar{q}-1}) \mathcal{C}_{(\bar{q})} \mathbf{M}_{\bar{q}}^{-1}\tag{A.7}$$

$$\mathbf{V}^{(\bar{q})} = \mathbf{M}_{\bar{q}} (\mathbf{F}_Q \otimes \cdots \otimes \mathbf{F}_{\bar{q}+1})^T\tag{A.8}$$

From eq.(A.7), the  $\bar{q}$ -th TT-core can be expressed as:

$$\begin{aligned}\mathcal{G}_{\bar{q}} &= \text{reshape}(\mathbf{U}^{(\bar{q})}; R_{\bar{q}-1}, N_{\bar{q}}, R_{\bar{q}}) \\ &= \mathcal{C}_{\bar{q}} \times_1 \mathbf{M}_{\bar{q}-1} \times_2 \mathbf{F}_{\bar{q}} \times_3 \mathbf{M}_{\bar{q}}^{-T},\end{aligned}\tag{A.9}$$

where  $\mathcal{C}_{\bar{q}} = \text{reshape}(\mathcal{C}; R_{\bar{q}-1}, T_{\bar{q}}, R_{\bar{q}})$  is absorbed in  $\mathcal{G}_{\bar{q}}$ .

Applying the same computation to the reshaping of  $\mathbf{V}^{(\bar{q})}$ , denoted by  $\mathbf{V}_{(2)}^{(\bar{q})}$  of size  $(R_{\bar{q}}N_{\bar{q}+1}) \times (N_{\bar{q}+2} \cdots N_Q)$ , and considering an identity matrix  $\mathbf{I}_{R_{\bar{q}} \times R_{\bar{q}}}$  in (A.8) such as  $\mathbf{V}^{(\bar{q})} = \mathbf{M}_{\bar{q}} \mathbf{I}_{R_{\bar{q}} \times R_{\bar{q}}} (\mathbf{F}_Q \otimes \cdots \otimes \mathbf{F}_{\bar{q}+1})^T$ , we obtain

$$\begin{aligned}\mathbf{V}_{(2)}^{(\bar{q})} &= \text{reshape}(\mathbf{V}^{(\bar{q})}; R_{\bar{q}}N_{\bar{q}+1}, \prod_{k=\bar{q}+2}^Q N_k) \\ &\triangleq (\mathbf{F}_{\bar{q}+1} \otimes \mathbf{M}_{\bar{q}}) \mathbf{I}_{(\bar{q}+1)} (\mathbf{F}_Q \otimes \cdots \otimes \mathbf{F}_{\bar{q}+2})^T \\ &\stackrel{\text{SVD}}{=} \mathbf{U}^{(\bar{q}+1)} \mathbf{V}^{(\bar{q}+1)}\end{aligned}$$

where  $\mathbf{I}_{(\bar{q}+1)} = \text{reshape}(\mathbf{I}_{R_{\bar{q}} \times R_{\bar{q}}}; R_{\bar{q}} \cdot T_{\bar{q}+1}, \frac{R_{\bar{q}}}{T_{\bar{q}+1}})$ . Matrices  $\mathbf{U}^{(\bar{q}+1)}$  and  $\mathbf{V}^{(\bar{q}+1)}$  are then expressed as

$$\begin{aligned}\mathbf{U}^{(\bar{q}+1)} &= (\mathbf{F}_{\bar{q}+1} \otimes \mathbf{M}_{\bar{q}}) \mathbf{I}_{(\bar{q}+1)} \mathbf{M}_{\bar{q}+1}^{-1}, \\ \mathbf{V}^{(\bar{q}+1)} &= \mathbf{M}_{\bar{q}+1} (\mathbf{F}_Q \otimes \cdots \otimes \mathbf{F}_{\bar{q}+2})^T.\end{aligned}\tag{A.10}$$

From eq.(A.10),  $\mathcal{G}_{\bar{q}+1}$  is expressed as:

$$\begin{aligned}\mathcal{G}_{\bar{q}+1} &= \text{reshape}(\mathbf{U}^{(\bar{q}+1)}; R_{\bar{q}}, N_{\bar{q}+1}, R_{\bar{q}+1}) \\ &= \tilde{\mathcal{T}}_{\bar{q}+1} \times_1 \mathbf{M}_{\bar{q}} \times_2 \mathbf{F}_{\bar{q}+1} \times_3 \mathbf{M}_{\bar{q}+1}^{-T},\end{aligned}\tag{A.11}$$

where  $\tilde{\mathcal{T}}_{\bar{q}+1} = \text{reshape}(\mathbf{I}_{R_{\bar{q}} \times R_{\bar{q}}}; R_{\bar{q}}, T_{\bar{q}+1}, R_{\bar{q}+1})$ . From (A.11), the result can be generalized to the  $Q - \bar{q}$  last TT-cores giving

$$\mathcal{G}_q = \tilde{\mathcal{T}}_q \times_1 \mathbf{M}_{q-1} \times_2 \mathbf{F}_q \times_3 \mathbf{M}_q^{-T}, (\bar{q} < q < Q).\tag{A.12}$$

Given the expressions of the TT-cores in (A.3), (A.6), (A.9) and (A.12), we can therefore deduce the result of Theorem 2.



## Appendix B

# Proof of Theorem 4

To prove Theorem 4, a constructive proof based on the TT-SVD algorithm is given for (2.3). We assume that the factors have full column rank. For that aim, we will follow step by step the methodology of the TT-SVD algorithm.

- The first unfolding  $\mathbf{X}_{(1)}$  of size  $N_1 \times (N_2 \cdots N_Q)$ , using Lemma 2, is given by:

$$\begin{aligned}\mathbf{X}_{(1)} &= \text{reshape}(\mathcal{X}; N_1, N_2 \cdots N_Q) \\ &= \mathbf{P}_1(\mathbf{P}_Q \odot \mathbf{P}_{Q-1} \odot \cdots \odot \mathbf{P}_2)^T.\end{aligned}$$

Note that, according to Lemma 1,  $(\mathbf{P}_Q \odot \mathbf{P}_{Q-1} \odot \cdots \odot \mathbf{P}_2)^T$  is a full row rank matrix of size  $R \times (N_2 \cdots N_Q)$ , since the factors  $\mathbf{P}_q$  are assumed to be full column rank.

- Applying the SVD to  $\mathbf{X}_{(1)}$ , the following expression holds:

$$\mathbf{X}_{(1)} = \mathbf{U}^{(1)}\mathbf{V}^{(1)} = \mathbf{G}_1\mathbf{V}^{(1)}$$

where  $\mathbf{G}_1 = \mathbf{U}^{(1)}$  contains the left singular vectors, and is the first TT-core of size  $N_1 \times R_1$ .  $\mathbf{V}^{(1)}$  contains the singular values and the the right singular vectors. It is of size  $R_1 \times (N_2 \cdots N_Q)$ . From the above two relations, we can conclude that

$$\text{rank}(\mathbf{X}_{(1)}) = \text{rank}(\mathbf{P}_1) = \text{rank}(\mathbf{G}_1) = R = R_1.$$

and we can express  $\mathbf{G}_1$  and  $\mathbf{V}^{(1)}$  as:

$$\begin{aligned}\mathbf{G}_1 &= \mathbf{P}_1\mathbf{M}_1^{-1}, \\ \mathbf{V}^{(1)} &= \mathbf{M}_1(\mathbf{P}_Q \odot \mathbf{P}_{Q-1} \odot \cdots \odot \mathbf{P}_2)^T\end{aligned}\tag{B.1}$$

where  $\mathbf{M}_1$  is a  $R \times R$  matrix that follows definition 8.

- Applying the TT-SVD algorithm, we have to reshape the matrix  $\mathbf{V}^{(1)}$  as a matrix of size  $(RN_2) \times (N_3 \cdots N_Q)$ , according to Lemma 2, which gives:

$$\begin{aligned}\mathbf{V}_{(2)}^{(1)} &= \text{reshape}(\mathbf{V}^{(1)}; RN_2, N_3 \cdots N_Q) \\ &= (\mathbf{P}_2 \odot \mathbf{M}_1)(\mathbf{P}_Q \odot \mathbf{P}_{Q-1} \odot \cdots \odot \mathbf{P}_3)^T.\end{aligned}\tag{B.2}$$

Note that, according to Lemma 1, the matrix  $\mathbf{P}_2 \odot \mathbf{M}_1$  of size  $(RN_2) \times R$  is full column rank, and  $(\mathbf{P}_Q \odot \mathbf{P}_{Q-1} \odot \cdots \odot \mathbf{P}_3)^T$  of size  $R \times (N_3 \cdots N_Q)$  is full row rank.



- The SVD of  $\mathbf{V}_{(2)}^{(1)}$  gives:

$$\mathbf{V}_{(2)}^{(1)} = \mathbf{U}^{(2)} \mathbf{V}^{(2)} \quad (\text{B.3})$$

where  $\mathbf{U}^{(2)}$  of size  $(RN_2) \times R_2$  is the reshaping of the 2nd TT-core  $\mathcal{G}_2$ . From (B.2) and (B.3), we can conclude that:

$$\text{rank} \mathbf{V}_{(2)}^{(1)} = \text{rank}(\mathbf{P}_2 \odot \mathbf{M}_1) = \text{rank} \mathbf{U}_2 \stackrel{\text{Lemma 1}}{=} R \stackrel{\text{TT}}{=} R_2$$

and we can write  $\mathbf{U}^{(2)}$  and  $\mathbf{V}^{(2)}$  as:

$$\begin{aligned} \mathbf{U}^{(2)} &= \mathbf{P}_2 \odot \mathbf{M}_1 \mathbf{M}_2^{-1}, \\ \mathbf{V}^{(2)} &= \mathbf{M}_2 (\mathbf{P}_Q \odot \mathbf{P}_{Q-1} \odot \cdots \odot \mathbf{P}_3)^T \end{aligned} \quad (\text{B.4})$$

where  $\mathbf{M}_2$  is a  $R \times R$  matrix that follows definition 8. Reshaping  $\mathbf{U}^{(2)}$ , and from (B.4), we obtain the expression of  $\mathcal{G}_2$  according to

$$\begin{aligned} \mathcal{G}_2 &= \text{reshape}(\mathbf{U}^{(2)}; R, N_2, R) \\ &= \mathcal{I}_{3,R} \times_1 \mathbf{M}_1 \times_2 \mathbf{P}_2 \times_3 \mathbf{M}_2^{-T} \end{aligned}$$

- Based on the same methodology, we find at the  $q$ -th step:

$$\mathbf{V}_{(2)}^{(q-1)} \stackrel{\text{Reshaping}}{=} (\mathbf{P}_q \odot \mathbf{M}_{q-1}) (\mathbf{P}_Q \odot \mathbf{P}_{Q-1} \odot \cdots \odot \mathbf{P}_{q+1})^T \stackrel{\text{SVD}}{=} \mathbf{U}^{(q)} \mathbf{V}^{(q)} \quad (\text{B.5})$$

where  $\mathbf{V}_{(2)}^{(q-1)}$  is of size  $(RN_q) \times (N_{q+1} \cdots N_Q)$ , and according to Lemma 1,  $(\mathbf{P}_q \odot \mathbf{M}_{q-1})$  is a full column rank matrix of size  $RN_q \times R$ ,  $(\mathbf{P}_Q \odot \mathbf{P}_{Q-1} \odot \cdots \odot \mathbf{P}_{q+1})^T$  is a full row rank matrix of size  $R \times (N_{q+1} \cdots N_Q)$ , and  $\mathbf{M}_{q-1}$  is a  $R \times R$  matrix that follows definition 8.

From (B.5), we can conclude that the relation between the factor matrices  $\mathbf{P}_q$  and the TT-cores  $\mathcal{G}_q$  is given by:

$$\mathbf{U}^{(q)} \mathbf{M}_q = \mathbf{P}_q \odot \mathbf{M}_{q-1}$$

or equivalently

$$\begin{aligned} \mathbf{U}^{(q)} &= (\mathbf{P}_q \odot \mathbf{M}_{q-1}) \mathbf{M}_q^{-1} \quad \text{for } 2 \leq q \leq Q-1 \\ &= (\mathbf{P}_q \odot \mathbf{M}_{q-1}) (\mathbf{M}_q^{-T})^T \end{aligned} \quad (\text{B.6})$$

where  $\mathbf{M}_q$  is a  $R \times R$  matrix that follows definition 8, and  $\mathbf{U}^{(q)} = \text{reshape}(\mathcal{G}_q; RN, R_q)$ . We can write:

$$\text{rank} \mathbf{V}_{(2)}^{(q-1)} = \text{rank}(\mathbf{P}_q \odot \mathbf{M}_{q-1}) = \text{rank} \mathbf{U}^{(q)} \stackrel{\text{Lemma 1}}{=} R \stackrel{\text{TT}}{=} R_q,$$

for  $2 \leq q \leq Q-1$ .

From (B.6) and considering  $\mathbf{U}^{(q)} = \text{reshape}(\mathcal{G}_q; RN_q, R_q)$ , we can see that  $\mathcal{G}_q$  follows a 3-order CPD according to:

$$\mathcal{G}_q = \mathcal{I}_{3,R} \times_1 \mathbf{M}_{q-1} \times_2 \mathbf{P}_q \times_3 \mathbf{M}_q^{-T} \quad (\text{B.7})$$

- At the last step, we have:

$$\mathbf{V}_{(2)}^{(Q-2)} \stackrel{\text{Reshaping}}{=} (\mathbf{P}_{Q-1} \odot \mathbf{M}_{Q-2}) \mathbf{P}_Q^T \stackrel{\text{SVD}}{=} \mathbf{U}^{(Q-1)} \mathbf{V}^{(Q-1)} = \mathbf{U}^{(Q-1)} \mathbf{G}_Q.$$

We then have for the last core:

$$\mathbf{G}_Q = \mathbf{M}_{Q-1} \mathbf{P}_Q^T. \tag{B.8}$$

Given (B.1), (B.7) and (B.8), we then have a proof for Theorem 4.



## Appendix C

# Proof of Theorem 5

The first part of the proof of Theorem 5 is obtained by the use of Theorem 4. In other words, assuming the factors in (3.4) have full column rank, expressions (3.5), (3.6), and (3.7) are obtained in a straightforward way thanks to Theorem 4.

To prove that two matrices span the same range space, it is equivalent to show the equality of their orthogonal projectors. This is the methodology used to prove (3.8) and (3.9). More precisely, assuming a full column rank for factor  $\mathbf{P}$  the following equalities hold:

$$\begin{aligned}\mathbf{G}_1 \mathbf{G}_1^\dagger &= (\mathbf{P} \mathbf{M}_1^{-1}) (\mathbf{P} \mathbf{M}_1^{-1})^\dagger \\ &= \mathbf{P} \mathbf{P}^\dagger\end{aligned}$$

$$\mathcal{G}_q(:, :, i)^\dagger \mathcal{G}_q(:, :, i) = \left( (\mathbf{M}_q^{-T}(i, :) \odot \mathbf{M}_{q-1}) \mathbf{P}^T \right)^\dagger \left( (\mathbf{M}_q^{-T}(i, :) \odot \mathbf{M}_{q-1}) \mathbf{P}^T \right) \quad (\text{C.1})$$

$$\begin{aligned}&= \mathbf{P}^{T\dagger} \left( (\mathbf{M}_q^{-T}(i, :) \odot \mathbf{M}_{q-1})^\dagger (\mathbf{M}_q^{-T}(i, :) \odot \mathbf{M}_{q-1}) \right) \mathbf{P}^T \quad (\text{C.2}) \\ &= (\mathbf{P} \mathbf{P}^\dagger)^T \\ &= \mathbf{P} \mathbf{P}^\dagger\end{aligned}$$

We recall that the  $i$ -th frontal slice of the  $q$ -th TT-core can be expressed as  $\mathcal{G}_q(:, :, i) = (\mathbf{M}_q^{-T}(i, :) \odot \mathbf{M}_{q-1}) \mathbf{P}^T$ . Note that, using Lemma 1, we can see that the product  $(\mathbf{M}_q^{-T}(i, :) \odot \mathbf{M}_{q-1})$  of size  $R \times R$ , is of rank  $R$ , since all  $\mathbf{M}_q$  are of rank  $R$ . This justifies the passage from (C.1) to (C.2). The same argument is used for the following equalities. Considering the  $j$ -th horizontal slice of the  $q'$ -th TT-core  $\mathcal{G}_{q'}(j, :, :) = \mathbf{P} (\mathbf{M}_{q'}^{-T} \odot \mathbf{M}_{q'-1}(j, :))^T$ , we have

$$\begin{aligned}\mathcal{G}_{q'}(j, :, :) \mathcal{G}_{q'}(j, :, :)^\dagger &= \left( \mathbf{P} (\mathbf{M}_{q'}^{-T} \odot \mathbf{M}_{q'-1}(j, :))^T \right) \left( \mathbf{P} (\mathbf{M}_{q'}^{-T} \odot \mathbf{M}_{q'-1}(j, :))^T \right)^\dagger \\ &= \mathbf{P} \left( (\mathbf{M}_{q'}^{-T} \odot \mathbf{M}_{q'-1}(j, :))^T (\mathbf{M}_{q'}^{-T} \odot \mathbf{M}_{q'-1}(j, :))^T \right) \mathbf{P}^\dagger \\ &= \mathbf{P} \mathbf{P}^\dagger \\ \mathbf{G}_Q^T \mathbf{G}_Q^{T\dagger} &= (\mathbf{P} \mathbf{M}_{Q-1}) (\mathbf{P} \mathbf{M}_{Q-1})^\dagger \\ &= \mathbf{P} \mathbf{P}^\dagger\end{aligned}$$

which justifies the equalities

$$\langle \mathbf{P} \rangle = \langle \mathbf{G}_1 \rangle = \langle \mathcal{G}_q(:, :, i) \rangle = \langle \mathcal{G}_{q'}(j, :, :) \rangle = \langle \mathbf{G}_Q^T \rangle,$$

and

$$\text{rank} \mathbf{P} = \text{rank} \mathbf{G}_1 = \text{rank} \mathcal{G}_q(:, :, i) = \text{rank} \mathcal{G}_{q'}(j, :, :) = \text{rank} \mathbf{G}_Q^T,$$

for  $2 \leq q \leq Q - 1, 2 \leq q' \leq Q - 1, 1 \leq i \leq R, 1 \leq j \leq R$ .

## Appendix D

# Algebraic analysis of the patterns

In this section, we give an algebraic analysis of the patterns to prove Lemmas 4 and 5. In the sequel only the analysis of the Splitting/Splitting pattern is given in detail. The analysis of the Generation/Generation, Splitting/Generation and Generation/Splitting patterns is very similar and is omitted here to the lack of space, only the expressions of inputs and outputs of these patterns are given. Note that in an informatic point of view, all the patterns can be implemented using the same function, that we call “Pattern”, since they can be considered similar up to index grouping. Before giving the algebraic analysis of the patterns, we present in Algorithm 12 a pseudocode of the function that can be used in the TT-HSVD algorithm. In this example, we suppose that the rank is estimated in the function “Pattern” using the `rank.m` of MatLab. Outputs  $\mathcal{S}_1$ ,  $\mathcal{S}_2$  can be either tensors or matrices depending on their respective dimensions `dim1` and `dim2`.

---

### Algorithm 12 Pattern

---

**Input:**  $\mathbf{M}$ , `dim1`, `dim2`

**Output:**  $\mathcal{S}_1$ ,  $\mathcal{S}_2$ ,  $R$ .

- 1:  $R = \text{rank}(\mathbf{M})$
  - 2:  $[\mathbf{U}, \mathbf{D}, \mathbf{V}] = \text{svd}(\mathbf{M}, R)$
  - 3:  $\mathbf{V} = \mathbf{D}\mathbf{V}^T$
  - 4:  $\mathcal{S}_1 = \text{reshape}(\mathbf{U}, \text{dim1})$
  - 5:  $\mathcal{S}_2 = \text{reshape}(\mathbf{V}, \text{dim2})$
- 

### Splitting/Splitting Pattern

This first pattern (Fig. 4.3) takes as input a matrix and returns 2 matrices. It applies the SVD to the input matrix and reshape the 2 matrices generated according to the choice we want. The graphical representation of this pattern is given in Fig. 4.3.

The input matrix  $\mathbf{X}_{(\bar{D})}$  can be expressed as:

$$\mathbf{X}_{(\bar{D})} = \sum_{r_{\bar{D}_f}, \dots, r_{\bar{D}_l}=1}^{R_{\bar{D}_f}, \dots, R_{\bar{D}_l}} \left( \mathbf{g}_{\bar{D}}(r_{\bar{D}-1}, r_{\bar{D}}) \otimes \dots \otimes \mathbf{g}_{\bar{D}_f+1}(r_{\bar{D}_f}, r_{\bar{D}_f+1}) \otimes \mathbf{Q}_{\bar{D}_f}^{-1}(:, r_{\bar{D}_f}) \right) \cdot \left( \mathbf{Q}_{\bar{D}_l}(r_{\bar{D}_l}, :)^T \otimes \mathbf{g}_{\bar{D}_l}(r_{\bar{D}_l-1}, r_{\bar{D}_l}) \otimes \dots \otimes \mathbf{g}_{\bar{D}+1}(r_{\bar{D}}, r_{\bar{D}+1}) \right)^T \quad (\text{D.1})$$

$$\begin{aligned} &= \sum_{r_{\bar{D}}=1}^{R_{\bar{D}}} \left( \sum_{r_{\bar{D}_f}, \dots, r_{\bar{D}-1}=1}^{R_{\bar{D}_f}, \dots, R_{\bar{D}-1}} \mathbf{g}_{\bar{D}}(r_{\bar{D}-1}, r_{\bar{D}}) \otimes \dots \otimes \mathbf{g}_{\bar{D}_f+1}(r_{\bar{D}_f}, r_{\bar{D}_f+1}) \otimes \mathbf{Q}_{\bar{D}_f}^{-1}(:, r_{\bar{D}_f}) \right) \\ &\quad \cdot \left( \sum_{r_{\bar{D}+1}, \dots, r_{\bar{D}_l}=1}^{R_{\bar{D}+1}, \dots, R_{\bar{D}_l}} \mathbf{Q}_{\bar{D}_l}(r_{\bar{D}_l}, :)^T \otimes \mathbf{g}_{\bar{D}_l}(r_{\bar{D}_l-1}, r_{\bar{D}_l}) \otimes \dots \otimes \mathbf{g}_{\bar{D}+1}(r_{\bar{D}}, r_{\bar{D}+1}) \right)^T \\ &= \sum_{r_{\bar{D}}=1}^{R_{\bar{D}}} \mathbf{S}(:, r_{\bar{D}}) \mathbf{T}(r_{\bar{D}}, :) = \mathbf{S} \mathbf{T}. \end{aligned} \quad (\text{D.2})$$

where  $\mathbf{S}$  and  $\mathbf{T}$  are respectively of size  $R_{\bar{D}_f}(I_{\bar{D}_f+1} \dots I_{\bar{D}}) \times R_{\bar{D}}$  and  $R_{\bar{D}} \times (I_{\bar{D}+1} \dots I_{\bar{D}_l})R_{\bar{D}_l}$ , and we have  $\text{rank}(\mathbf{X}_{(\bar{D})}) = R_{\bar{D}}$ .

Note that the expression (D.1) corresponds to the definition given in (2.7), where  $R_{\bar{D}_f} = R_0 = 1$ ,  $R_{\bar{D}_l} = R_D = 1$ , and  $\mathbf{Q}_0 = \mathbf{Q}_D = 1$ .

Applying the SVD on  $\mathbf{X}_{(\bar{D})}$  gives:

$$\mathbf{X}_{(\bar{D})} = \mathbf{U}_{\bar{D}} \mathbf{V}_{\bar{D}}. \quad (\text{D.3})$$

From (D.2) and (D.3), we can conclude that:

$$\begin{aligned} \mathbf{U}_{\bar{D}} &= \mathbf{S} \mathbf{Q}_{\bar{D}} \\ &= \sum_{r_{\bar{D}}=1}^{R_{\bar{D}}} \left( \sum_{r_{\bar{D}_f}, \dots, r_{\bar{D}-1}=1}^{R_{\bar{D}_f}, \dots, R_{\bar{D}-1}} \mathbf{g}_{\bar{D}}(r_{\bar{D}-1}, r_{\bar{D}}) \otimes \dots \otimes \mathbf{g}_{\bar{D}_f+1}(r_{\bar{D}_f}, r_{\bar{D}_f+1}) \otimes \mathbf{Q}_{\bar{D}_f}^{-1}(:, r_{\bar{D}_f}) \right) \\ &\quad \cdot \mathbf{Q}(r_{\bar{D}}, :) \\ &= \sum_{r_{\bar{D}_f}, \dots, r_{\bar{D}}=1}^{R_{\bar{D}_f}, \dots, R_{\bar{D}}} \left( \mathbf{g}_{\bar{D}}(r_{\bar{D}-1}, r_{\bar{D}}) \otimes \dots \otimes \mathbf{g}_{\bar{D}_f+1}(r_{\bar{D}_f}, r_{\bar{D}_f+1}) \otimes \mathbf{Q}_{\bar{D}_f}^{-1}(:, r_{\bar{D}_f}) \right) \mathbf{Q}(r_{\bar{D}}, :) \end{aligned} \quad (\text{D.4})$$

and

$$\begin{aligned} \mathbf{V}_{\bar{D}} &= \mathbf{Q}_{\bar{D}}^{-1} \mathbf{T} \\ &= \sum_{r_{\bar{D}}=1}^{R_{\bar{D}}} \mathbf{Q}_{\bar{D}}^{-1}(:, r_{\bar{D}}) \left( \sum_{r_{\bar{D}+1}, \dots, r_{\bar{D}_l}=1}^{R_{\bar{D}+1}, \dots, R_{\bar{D}_l}} \mathbf{Q}_{\bar{D}_l}(r_{\bar{D}_l}, :)^T \otimes \mathbf{g}_{\bar{D}_l}(r_{\bar{D}_l-1}, r_{\bar{D}_l}) \otimes \dots \otimes \mathbf{g}_{\bar{D}+1}(r_{\bar{D}}, r_{\bar{D}+1}) \right)^T \\ &= \sum_{r_{\bar{D}}, \dots, r_{\bar{D}_l}=1}^{R_{\bar{D}}, \dots, R_{\bar{D}_l}} \mathbf{Q}_{\bar{D}}^{-1}(:, r_{\bar{D}}) \left( \mathbf{Q}_{\bar{D}_l}(r_{\bar{D}_l}, :)^T \otimes \mathbf{g}_{\bar{D}_l}(r_{\bar{D}_l-1}, r_{\bar{D}_l}) \otimes \dots \otimes \mathbf{g}_{\bar{D}+1}(r_{\bar{D}}, r_{\bar{D}+1}) \right)^T \end{aligned} \quad (\text{D.5})$$

where  $\mathbf{Q}_{\bar{D}}$  is a  $R_{\bar{D}} \times R_{\bar{D}}$  change-of-basis matrix.

Let  $\mathbf{X}_{(\bar{D}_f+\bar{D}')} and  $\mathbf{X}_{(\bar{D}+\bar{D}'')}$  be the reshaping of the matrices  $\mathbf{U}_{\bar{D}}$  and  $\mathbf{V}_{\bar{D}}$  according to$

the chosen  $\bar{D}'$  and  $\bar{D}''$ . From (D.4) and (D.5), we have:

$$\begin{aligned} \mathbf{X}_{(\bar{D}_f + \bar{D}')} &= \text{reshape}(\mathbf{U}_{\bar{D}}, R_{\bar{D}_f} I_{\bar{D}_f+1} \cdots I_{\bar{D}_f + \bar{D}'}, I_{\bar{D}_f + \bar{D}'+1} \cdots I_{\bar{D}} R_{\bar{D}}) \\ &= \sum_{r_{\bar{D}_f}, \dots, r_{\bar{D}}=1}^{R_{\bar{D}_f}, \dots, R_{\bar{D}}} \left( \mathbf{g}_{\bar{D}_f + \bar{D}'}(r_{\bar{D}_f + \bar{D}'-1}, r_{\bar{D}_f + \bar{D}'}) \otimes \cdots \otimes \mathbf{g}_{\bar{D}_f+1}(r_{\bar{D}_f}, r_{\bar{D}_f+1}) \otimes \mathbf{Q}_{\bar{D}_f}^{-1}(:, r_{\bar{D}_f}) \right) \\ &\quad \cdot \left( \mathbf{Q}_{\bar{D}}(r_{\bar{D}}, :)^T \otimes \mathbf{g}_{\bar{D}}(r_{\bar{D}-1}, r_{\bar{D}}) \otimes \cdots \otimes \mathbf{g}_{\bar{D}'+1}(r_{\bar{D}'}, r_{\bar{D}'+1}) \right)^T \end{aligned} \quad (\text{D.6})$$

and

$$\begin{aligned} \mathbf{X}_{(\bar{D} + \bar{D}'')} &= \text{reshape}(\mathbf{V}_{\bar{D}}, R_{\bar{D}} I_{\bar{D}+1} \cdots I_{\bar{D} + \bar{D}''}, I_{\bar{D} + \bar{D}''+1} \cdots I_{\bar{D}_l} R_{\bar{D}_l}) \\ &= \sum_{r_{\bar{D}}, \dots, r_{\bar{D}_l}=1}^{R_{\bar{D}}, \dots, R_{\bar{D}_l}} \left( \mathbf{g}_{\bar{D} + \bar{D}''}(r_{\bar{D} + \bar{D}''-1}, r_{\bar{D} + \bar{D}''}) \otimes \cdots \otimes \mathbf{g}_{\bar{D}+1}(r_{\bar{D}}, r_{\bar{D}+1}) \otimes \mathbf{Q}_{\bar{D}}^{-1}(:, r_{\bar{D}}) \right) \\ &\quad \cdot \left( \mathbf{Q}_{\bar{D}_l}(r_{\bar{D}_l}, :)^T \otimes \mathbf{g}_{\bar{D}_l}(r_{\bar{D}_l-1}, r_{\bar{D}_l}) \otimes \cdots \otimes \mathbf{g}_{\bar{D} + \bar{D}''+1}(r_{\bar{D} + \bar{D}''}, r_{\bar{D} + \bar{D}''+1}) \right)^T \end{aligned} \quad (\text{D.7})$$

### Splitting/Generation Pattern:

This pattern (Fig. 4.4 (left)) also takes as input a matrix and gives as outputs a tensor and a reshaped matrix according to our choice. According to (D.6) and (D.7), the matrix  $\mathbf{X}_{(\bar{D})}$  will be expressed as:

$$\begin{aligned} \mathbf{X}_{(\bar{D})} &= \sum_{r_{\bar{D}_f}, \dots, r_{\bar{D}+1}=1}^{R_{\bar{D}_f}, \dots, R_{\bar{D}+1}} \left( \mathbf{g}_{\bar{D}}(r_{\bar{D}-1}, r_{\bar{D}}) \otimes \cdots \otimes \mathbf{g}_{\bar{D}_f+1}(r_{\bar{D}_f}, r_{\bar{D}_f+1}) \otimes \mathbf{Q}_{\bar{D}_f}^{-1}(:, r_{\bar{D}_f}) \right) \\ &\quad \cdot \left( \mathbf{Q}_{\bar{D}+1}(r_{\bar{D}+1}, :)^T \otimes \mathbf{g}_{\bar{D}+1}(r_{\bar{D}}, r_{\bar{D}+1}) \right)^T \end{aligned} \quad (\text{D.8})$$

The outputs  $\mathbf{X}_{(\bar{D}')}$  and  $\hat{\mathcal{G}}_{\bar{D}+1}$  of this pattern will then be expressed as follows:

$$\begin{aligned} \mathbf{X}_{(\bar{D}_f + \bar{D}')} &= \text{reshape}(\mathbf{U}_{\bar{D}}, R_{\bar{D}_f} I_{\bar{D}_f+1} \cdots I_{\bar{D}_f + \bar{D}'}, I_{\bar{D}_f + \bar{D}'+1} \cdots I_{\bar{D}} R_{\bar{D}}) \\ &= \sum_{r_{\bar{D}_f}, \dots, r_{\bar{D}}=1}^{R_{\bar{D}_f}, \dots, R_{\bar{D}}} \left( \mathbf{g}_{\bar{D}_f + \bar{D}'}(r_{\bar{D}_f + \bar{D}'-1}, r_{\bar{D}_f + \bar{D}'}) \otimes \cdots \otimes \mathbf{g}_{\bar{D}_f+1}(r_{\bar{D}_f}, r_{\bar{D}_f+1}) \otimes \mathbf{Q}_{\bar{D}_f}^{-1}(:, r_{\bar{D}_f}) \right) \\ &\quad \cdot \left( \mathbf{Q}_{\bar{D}}(r_{\bar{D}}, :)^T \otimes \mathbf{g}_{\bar{D}}(r_{\bar{D}-1}, r_{\bar{D}}) \otimes \cdots \otimes \mathbf{g}_{\bar{D}'+1}(r_{\bar{D}'}, r_{\bar{D}'+1}) \right)^T \end{aligned} \quad (\text{D.9})$$

and

$$\begin{aligned} \hat{\mathcal{G}}_{\bar{D}+1} &= \text{reshape}(\mathbf{V}_{\bar{D}}, R_{\bar{D}}, I_{\bar{D}+1}, R_{\bar{D}+1}) \\ &= \mathbf{Q}_{\bar{D}}^{-1} \times_2^1 \mathcal{G}_{\bar{D}+1} \times_3^1 \mathbf{Q}_{\bar{D}+1} \end{aligned} \quad (\text{D.10})$$

The Generation/Splitting pattern has same expressions as the Splitting/Generation pattern if the outputs are reversed.



### Generation/Generation Pattern:

It is the third and last type of core generation patterns (Fig. 4.5). This pattern has as input matrices of dimension  $(R_{\bar{D}-1}I_{\bar{D}}) \times (I_{\bar{D}+1}R_{\bar{D}+1})$ . It outputs 2 tensors at time. From what we have seen before, the matrix  $\mathbf{X}_{(\bar{D})}$  will be expressed as:

$$\begin{aligned} \mathbf{X}_{(\bar{D})} = & \sum_{r_{\bar{D}-1}, r_{\bar{D}}, r_{\bar{D}+1}=1}^{R_{\bar{D}-1}, R_{\bar{D}}, R_{\bar{D}+1}} \left( \mathbf{g}_{\bar{D}}(r_{\bar{D}-1}, r_{\bar{D}}) \otimes \mathbf{Q}_{\bar{D}-1}^{-1}(:, r_{\bar{D}-1}) \right) \\ & \cdot \left( \mathbf{Q}_{\bar{D}+1}(r_{\bar{D}+1}, :)^T \otimes \mathbf{g}_{\bar{D}+1}(r_{\bar{D}}, r_{\bar{D}+1}) \right)^T \end{aligned} \quad (\text{D.11})$$

The outputs  $\hat{\mathcal{G}}_{\bar{D}}$  and  $\hat{\mathcal{G}}_{\bar{D}+1}$  of this pattern will then be expressed as follows:

$$\begin{aligned} \hat{\mathcal{G}}_{\bar{D}} &= \text{reshape}(\mathbf{U}_{\bar{D}}, R_{\bar{D}-1}, I_{\bar{D}}, R_{\bar{D}}) \\ &= \mathbf{Q}_{\bar{D}-1}^{-1} \times_2^1 \mathcal{G}_{\bar{D}} \times_3^1 \mathbf{Q}_{\bar{D}} \end{aligned} \quad (\text{D.12})$$

and

$$\begin{aligned} \hat{\mathcal{G}}_{\bar{D}+1} &= \text{reshape}(\mathbf{V}_{\bar{D}}, R_{\bar{D}}, I_{\bar{D}+1}, R_{\bar{D}+1}) \\ &= \mathbf{Q}_{\bar{D}}^{-1} \times_2^1 \mathcal{G}_{\bar{D}+1} \times_3^1 \mathbf{Q}_{\bar{D}+1} \end{aligned} \quad (\text{D.13})$$

# Bibliography

- [1] E. Acar, R. Bro, and A. Smilde. "Data fusion in metabolomics using coupled matrix and tensor factorizations". In: *Proceedings of the IEEE* 103 (2015), pp. 1602–1620.
- [2] E. Acar, T. G. Kolda, and D. M. Dunlavy. "All-at-once Optimization for Coupled Matrix and Tensor Factorizations". In: *arXiv:1105.3422v1* (2011).
- [3] A. L. F. de Almeida, G. Favier, and J. C. Mota. "PARAFAC-based unified tensor modeling for wireless communication systems with application to blind multiuser equalization". In: *Signal Processing* 87.2 (2007), pp. 337–351.
- [4] A. L. F. de Almeida, G. Favier, and J. Mota. "A constrained factor decomposition with application to MIMO antenna systems". In: *IEEE Transactions on Signal Processing* 56 (2008), pp. 2429–2442.
- [5] R. Badeau and R. Boyer. "Fast multilinear singular value decomposition for structured tensors". In: *SIAM Journal on Matrix Analysis and Applications* 30 (2008), pp. 1008–1021.
- [6] R. Badeau, B. David, and G. Richard. "A new perturbation analysis for signal enumeration in rotational invariance techniques". In: *IEEE Transactions on Signal Processing* 54 (2006), pp. 450–458.
- [7] J. Ballani, L. Grasedyck, and M. Kluge. "Black box approximation of tensors in hierarchical Tucker format". In: *Linear algebra and its Applications* 438.2 (2013), pp. 639–657.
- [8] J. M. F. ten Berge and N. D. Sidiropoulos. "On uniqueness in CANDECOMP/PARAFAC". In: *Psychometrika* 67.3 (Sept. 2002), pp. 399–409.
- [9] J. T. Berge. "The k-rank of a Khatri-Rao product". Unpublished Note, Heijmans Institute of Psychological Research, University of Groningen, The Netherlands. 2000.
- [10] G. Bergqvist and E. G Larsson. "The higher-order singular value decomposition: Theory and an application [lecture notes]". In: *IEEE Signal Processing Magazine* 27.3 (2010), pp. 151–154.
- [11] A. Boudehane, Y. Zniyed, A. Tenenhaus, L. L. Brusquet, and R. Boyer. "Breaking the curse of dimensionality for coupled tensor-matrix factorization". In: *submitted to CAMSAP*. 2019.
- [12] M. Boussé, O. Debals, and L. D. Lathauwer. "A Tensor-Based Method for Large-Scale Blind Source Separation Using Segmentation". In: *IEEE Transactions on Signal Processing* 65 (2016), pp. 346–358.
- [13] M. Boussé, O. Debals, and L. D. Lathauwer. "A Tensor-Based Method for Large-Scale Blind Source Separation Using Segmentation". In: *IEEE Transactions on Signal Processing* 65 (2016), pp. 346–358.
- [14] R. Boyer. "Deterministic asymptotic Cramér-Rao bound for the multidimensional harmonic model". In: *Elsevier Signal Processing* 88 (2008), pp. 2869–2877.

- [15] R. Boyer and P. Comon. "Rectified ALS Algorithm for Multidimensional Harmonic Retrieval". In: *Sensor Array and Multichannel Signal Processing Workshop (SAM)*. 2016.
- [16] R. Boyer and M. Haardt. "Noisy Compressive Sampling Based on Block-Sparse Tensors: Performance Limits and Beamforming Techniques". In: *IEEE Transactions on Signal Processing* 64 (2016), pp. 6075–6088.
- [17] R. Boyer and R. Badeau. "Adaptive Multilinear SVD for Structured Tensors". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2006.
- [18] R. Boyer, R. Badeau, and G. Favier. "Fast Orthogonal Decomposition of Volterra Cubic Kernels using Oblique Unfolding". In: *36th IEEE International Conference on Acoustics, Speech and Signal Processing*. 2011.
- [19] J. Brachat, P. Comon, B. Mourrain, and E. Tsigaridas. "Symmetric tensor decomposition". In: *Linear Algebra and its Applications* 433.11 (2010), pp. 1851 – 1872.
- [20] D. Brie, R. Klotz, S. Miron, S. Moussaoui, C. Mustin, P. Bécuwe, and S. Grandemange. "Joint analysis of flow cytometry data and fluorescence spectra as a non-negative array factorization problem". In: *Chemometrics and Intelligent Laboratory Systems* 137 (2014), pp. 21–32.
- [21] R. Bro. "PARAFAC. Tutorial and applications". In: *Chemometrics and Intelligent Laboratory Systems* 38.2 (1997), pp. 149–171.
- [22] R. Bro, N. D. Sidiropoulos, and G. B. Giannakis. "A Fast Least Squares Algorithm for Separating Trilinear Mixtures". In: *Int. Workshop on Independent Component Analysis and Blind Separation* (1999).
- [23] R. Bro, R. A. Harshman, N. D. Sidiropoulos, and M. E. Lundy. "Modeling multi-way data with linearly dependent loadings". In: *Journal of Chemometrics: A Journal of the Chemometrics Society* 23.7-8 (2009), pp. 324–340.
- [24] G. Camba-Mendez and G. Kapetanios. "Statistical tests and estimators of the rank of a matrix and their applications in econometric modelling". In: *Econometric Reviews* 28 (2009), pp. 581–611.
- [25] J. Carroll and J.-J. Chang. "Analysis of individual differences in multidimensional scaling via an  $n$ -way generalization of 'Eckart-Young' decomposition". In: *Psychometrika* 35.3 (1970), pp. 283–319.
- [26] T. F. Chan. "Rank revealing QR factorizations". In: *Linear algebra and its applications* 88 (1987), pp. 67–82.
- [27] E. Chaumette, J. Galy, A. Quinlan, and P. Larzabal. "A New Barankin Bound Approximation for the Prediction of the Threshold Region Performance of Maximum-Likelihood Estimators". In: *IEEE Transactions on Signal Processing* 56 (2008), pp. 5319–5333.
- [28] A. Cichocki. "Era of Big Data Processing: A New Approach via Tensor Networks and Tensor Decompositions". In: *CoRR* (2014).
- [29] A. Cichocki. "Era of Big Data Processing: A New Approach via Tensor Networks and Tensor Decompositions". In: *CoRR* <http://arxiv.org/pdf/1403.2048.pdf> (2014).

- [30] A. Cichocki, N. Lee, I. Oseledets, A.-H. Phan, Q. Zhao, and D. Mandic. "Low-Rank Tensor Networks for Dimensionality Reduction and Large-Scale Optimization Problems: Perspectives and Challenges PART 1". In: *CoRR, arXiv:1609.00893* (2016).
- [31] A. Cichocki. "Tensor Networks for Big Data Analytics and Large-Scale Optimization Problems". In: *arXiv preprint arXiv:1407.3124* (2014).
- [32] A. Cichocki, D. Mandic, L. De Lathauwer, G. Zhou, Q. Zhao, C. Caiafa, and H. A. Phan. "Tensor Decompositions for Signal Processing Applications". In: *IEEE Signal Processing Magazine* 32 (2015), pp. 145–163.
- [33] A. Cichocki, N. Lee, I. Oseledets, A.-H. Phan, Q. Zhao, and D. P. Mandic. "Tensor Networks for Dimensionality Reduction and Large-Scale Optimization Part 1 Low-Rank Tensor Decompositions". In: *Foundations and Trends in Machine Learning* 9.4-5 (2016), pp. 249–429.
- [34] M. Clark and L. Scharf. "Two-dimensional modal analysis based on maximum likelihood". In: *IEEE Transactions on Signal Processing* 42 (1994), pp. 1443–1452.
- [35] J. E. Cohen. "Environmental Multiway Data Mining". PhD thesis. Université Grenoble Alpes, 2016.
- [36] P. Comon, X. Luciani, and A. L. de Almeida. "Tensor Decompositions, Alternating Least Squares and other Tales". In: *Journal of Chemometrics* 23 (2009), pp. 393–405.
- [37] P. Comon, G. Golub, L.-H. Lim, and B. Mourrain. "Symmetric Tensors and Symmetric Tensor Rank". In: *SIAM Journal on Matrix Analysis and Applications* 30 (2008), pp. 1254–1279.
- [38] L. De Lathauwer, B. De Moor, and J. Vandewalle. "A multilinear singular value decomposition". In: *SIAM Journal on Matrix Analysis and Applications* 21 (2000), pp. 1253–1278.
- [39] L. De Lathauwer, B. De Moor, and J. Vandewalle. "On the Best Rank-1 and Rank- $(R_1, R_2, \dots, R_N)$  Approximation of Higher-Order Tensors". In: *SIAM Journal on Matrix Analysis and Applications* 21.4 (2006), pp. 1324–1342.
- [40] J. H. De Morais Goulart, M. B. R. Boyer, G. Favier, and P. Comon. "Tensor CP Decomposition with structured factor matrices: Algorithms and Performance". In: *IEEE Journal of Selected Topics in Signal Processing* 10.4 (2016), pp. 757–769.
- [41] V. De Silva and L.-H. Lim. "Tensor rank and the ill-posedness of the best low-rank approximation problem". In: *SIAM Journal on Matrix Analysis and Applications* 30.3 (2008), pp. 1084–1127.
- [42] V. De Silva and L.-H. Lim. "Tensor rank and the ill-posedness of the best low-rank approximation problem". In: *SIAM J. Matrix Anal. Appl.* 30.3 (2008), pp. 1084–1127.
- [43] M. Dohler and Y. Li. *Cooperative Communications: Hardware, Channel and PHY*. John Wiley & Sons, 2010.
- [44] I. Domanov and L. De Lathauwer. "Canonical polyadic decomposition of third-order tensors: Relaxed uniqueness conditions and algebraic algorithm". In: *Linear Algebra and its Applications* 513.Supplement C (2017), pp. 342–375.
- [45] C. Eckart and G. Young. "THE APPROXIMATION OF ONE MATRIX BY ANOTHER OF LOWER RANK". In: *Psychometrika* (1936), pp. 211–218.

- [46] S. Etter. "Parallel ALS Algorithm for Solving Linear Systems in the Hierarchical Tucker Representation". In: *SIAM J. Scientific Computing* 38 (2016), pp. 2585–2609.
- [47] R. C. Farias, J. Cohen, and P. Comon. "Exploring multimodal data fusion through joint decompositions with flexible couplings". In: *IEEE Trans. SP* 64 (2016), pp. 4830–4844.
- [48] G. Favier and A. L. F. de Almeida. "Tensor space-time-frequency coding with semi-blind receivers for MIMO wireless communication systems". In: *IEEE Transactions on Signal Processing* 62 (2014), pp. 5987–6002.
- [49] G. Favier, C. A. R. Fernandes, and A. L. F. de Almeida. "Nested Tucker Tensor Decomposition with Application to MIMO Relay Systems Using Tensor Space-Time Coding (TSTC)". In: *Signal Processing* 128 (2016), pp. 318–331.
- [50] G. Favier and A. L. F. de Almeida. "Tensor Space-Time-Frequency Coding With Semi-Blind Receivers for MIMO Wireless Communication Systems". In: *IEEE Transactions on Signal Processing* 62 (2014), pp. 5987–6002.
- [51] W. C. Freitas, G. Favier, and A. L. F. de Almeida. "Sequential Closed-Form Semiblind Receiver for Space-Time Coded Multihop Relaying Systems". In: *IEEE Signal Processing Letters* 24.12 (2017), pp. 1773–1777. ISSN: 1070-9908. DOI: [10.1109/LSP.2017.2756846](https://doi.org/10.1109/LSP.2017.2756846).
- [52] W. C. Freitas, G. Favier, and A. L. F. de Almeida. "Tensor-Based Joint Channel and Symbol Estimation for Two-Way MIMO Relaying Systems". In: *IEEE Signal Processing Letters* 26.2 (2019), pp. 227–231. ISSN: 1070-9908. DOI: [10.1109/LSP.2018.2885227](https://doi.org/10.1109/LSP.2018.2885227).
- [53] G. H. Golub and C. F. Van Loan. *Matrix Computations*. 4th. Baltimore: The Johns Hopkins University Press, 2013.
- [54] J. H. Goulart, M. Boizard, R. Boyer, G. Favier, and P. Comon. "Tensor CP Decomposition with structured factor matrices: Algorithms and Performance". In: *IEEE Journal of Selected Topics in Signal Processing* 10 (2016), pp. 757–769.
- [55] L. Grasedyck. "Hierarchical singular value decomposition of tensors". In: *SIAM J. Matrix Anal. Appl.* 31 (2010), pp. 2029–2054.
- [56] L. Grasedyck and W. Hackbusch. "An introduction to hierarchical (h-) rank and TT-rank of tensors with examples". In: *Comput. Meth. in Appl. Math.* 11.3 (2011), pp. 291–304.
- [57] L. Grasedyck, D. Kressner, and C. Tobler. "A literature survey of low-rank tensor approximation techniques". In: *CGAMM-Mitteilungen* 36.3 (2013), pp. 53–78.
- [58] X. Guo, S. Miron, D. Brie, and A. Stegeman. "Uni-mode and partial uniqueness conditions for CANDECOMP/PARAFAC of three-way arrays with linearly dependent loadings". In: *SIAM Journal on Matrix Analysis and Applications* 33.1 (2012), pp. 111–129.
- [59] W. Hackbusch and S. Kühn. "A new scheme for the tensor representation". In: *Journal of Fourier Analysis and Applications* 15 (Oct. 2009), pp. 706–722.
- [60] R. A. Harshman. "Foundations of the PARAFAC procedure: Models and conditions for an 'explanatory' multi-modal factor analysis". In: *UCLA Working Papers in Phonetics* 16 (1970), pp. 1–84.

- [61] R. A. Harshman. "Foundations of the PARAFAC procedure: Models and conditions for an explanatory multimodal factor analysis". In: *UCLA Working Papers in Phonetics* 16 (1970), pp. 1–84.
- [62] C. J. Hillar and L.-H. Lim. "Most Tensor Problems Are NP-Hard". In: *J. ACM* 60.6 (Nov. 2013), 45:1–45:39.
- [63] F. L. Hitchcock. "Multiple invariants and generalized rank of a p-way matrix or tensor". In: *Journal of Mathematics and Physics* 7 (1927), pp. 39–79.
- [64] Y. Hua and T. K. Sarkar. "Matrix pencil method for estimating parameters of exponentially damped/undamped sinusoids in noise". In: *IEEE Trans. Acoust., Speech, Signal Processing* 38 (1990), pp. 814–824.
- [65] M. Jacquelin, L. Marchal, and Y. Robert. "Complexity Analysis and Performance Evaluation of Matrix Product on Multicore Architectures". In: *International Conference on Parallel Processing, Vienna, Austria*. 2009.
- [66] T. Jiang, N. D. Sidiropoulos, and J. M.F. T. Berge. "Almost-sure identifiability of multidimensional harmonic retrieval". In: *IEEE Transactions on Signal Processing* 49.9 (2001), pp. 1849–1859.
- [67] T. Jiang, N. Sidiropoulos, and J. ten Berge. "Almost sure identifiability of multidimensional harmonic retrieval". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2001).
- [68] M. Jo, D. Araujo, T. Maksymyuk, A. L. F. de Almeida, T. F. Maciel, and J. C. M. Mota. "Massive MIMO: Survey and future research topics". In: *IET Communications* (2016).
- [69] A. Kammoun, H. Khanfir, Z. Altman, M. Debbah, and M. Kamoun. "Preliminary Results on 3D Channel Modeling: From Theory to Standardization". In: *IEEE Journal on Selected Areas in Communications* 32 (2014), pp. 1219–1229.
- [70] N. Kargas and N. Sidiropoulos. "Completing a joint PMF from projections: a low-rank coupled tensor factorization approach". In: *Proc. IEEE ITA, San Diego, CA* (2017).
- [71] V. Kazeev, O. Reichmann, and C. Schwab. "Low-rank tensor structure of linear diffusion operators in the TT and QTT formats". In: *Linear Algebra and its Applications* 438.11 (2013), pp. 4204–4221.
- [72] B. Khoromskij. " $O(d \log N)$ -Quantics Approximation of N-d Tensors in High-Dimensional Numerical Modeling". In: *Constructive Approximation* 34.2 (2011), pp. 257–280.
- [73] B. Khoromskij. "Tensors-structured numerical methods in scientific computing: Survey on recent advances". In: *Chemometrics and Intelligent Laboratory Systems* 110.1 (2011), pp. 1–19.
- [74] A. Y. Kibangou and G. Favier. "Non-iterative solution for PARAFAC with a Toeplitz matrix factor". In: *17th European Signal Processing Conference*. 2009.
- [75] A. Y. Kibangou and G. Favier. "Toeplitz-Vandermonde Matrix Factorization With Application to Parameter Estimation of Wiener-Hammerstein Systems". In: *IEEE Signal Processing Letters* 14 (2007), pp. 141–144.
- [76] L. Knockaert. "The Barankin bound and threshold behavior in frequency estimation". In: *IEEE Transactions on Signal Processing* 45 (1997), pp. 2398–2401.

- [77] T. G. Kolda. "A Counterexample to the Possibility of an Extension of the Eckart–Young Low-Rank Approximation Theorem for the Orthogonal Rank Tensor Decomposition". In: *SIAM Journal on Matrix Analysis and Applications* 24.3 (2003), pp. 762–767.
- [78] T. G. Kolda. "A Counterexample to the Possibility of an Extension of the Eckart–Young Low-Rank Approximation Theorem for the Orthogonal Rank Tensor Decomposition". In: *SIAM J. Matrix Anal. Appl.* 24.3 (2003), pp. 762–767.
- [79] T. G. Kolda and B. W. Bader. "Tensor decompositions and applications". In: *SIAM Rev.* 51 (2009), pp. 455–500.
- [80] T. G. Kolda and J. Sun. "Scalable Tensor Decompositions for Multi-aspect Data Mining". In: *Eighth IEEE International Conference on Data Mining*. 2008.
- [81] D. Kressner, M. Steinlechner, and B. Vandereycken. "Low-rank tensor completion by Riemannian optimization". In: *BIT Numerical Mathematics* 54 (2014), pp. 447–468.
- [82] D. Kressner and C. Tobler. "Algorithm 941: h-Tucker - A Matlab Toolbox for Tensors in Hierarchical Tucker Format". In: *Math. Softw.* 40.3 (2014), p. 22.
- [83] S. Kritchman and B. Nadler. "Non-parametric detection of the number of signals: Hypothesis testing and random matrix theory". In: *IEEE Transactions on Signal Processing* 57 (2009), pp. 3930–3941.
- [84] P. M. Kroonenberg and J. de Leeuw. "Principal component analysis of three-mode data by means of alternating least squares algorithms". In: *Psychometrika* 45 (1980), pp. 69–97.
- [85] P. M. Kroonenberg and J. de Leeuw. "Principal component analysis of three-mode data by means of alternating least squares algorithms". In: *Psychometrika* 45 (1980), pp. 69–97.
- [86] J. B. Kruskal. "Three-way arrays: Rank and uniqueness of trilinear decompositions with application to arithmetic complexity and statistics". In: *Linear Algebra Appl.* 18 (1977), pp. 95–138.
- [87] D. Lahat, T. Adali, and C. Jutten. "Multimodal Data Fusion: An Overview of Methods, Challenges and Prospects". In: *Proceedings of the IEEE* 103 (2015), pp. 1449–1477.
- [88] J. N. Laneman, D. Tse, and G. W. Wornell. "Cooperative Diversity in Wireless Networks: Efficient Protocols and Outage Behavior". In: *IEEE Transactions on Information theory* 50.12 (2004), pp. 3062–3080.
- [89] L. D. Lathauwer, B. D. Moor, and J. Vandewalle. "A multilinear singular value decomposition". In: *SIAM J. Matrix Anal. Appl.* 21 (2000), pp. 1253–1278.
- [90] S. L. Lauritzen. *Graphical models*. Vol. 17. Clarendon Press, 1996.
- [91] N. Lee and A. Cichocki. "Very Large-Scale Singular Value Decomposition Using Tensor Train Networks". In: *arXiv:1410.6895v2* (2014).
- [92] N. Li, S. Kindermann, and C. Navasca. "Some convergence results on the Regularized Alternating Least-Squares method for tensor decomposition". In: *Linear Algebra and its Applications* 438.2 (2013), pp. 796–812.
- [93] Y. Li, J. Razavilar, and K. Liu. "A high-resolution technique for multidimensional NMR spectroscopy". In: *IEEE Transactions on Biomedical Engineering* 45 (1998), pp. 78–86.

- [94] P. Lioliou, M. Viberg, and M. Coldrey. "Efficient Channel Estimation Techniques for Amplify and Forward Relaying Systems". In: *IEEE Transactions on Communications* 60.11 (2012), pp. 3150–3155.
- [95] J. Liu and X. Liu. "An eigenvector-based approach for multidimensional frequency estimation with improved identifiability". In: *IEEE Transactions on Signal Processing* 54 (2006), pp. 4543–4556.
- [96] K. J. R. Liu, A. K. Sadek, W. Su, and A. Kwasinski. *Cooperative Communications and Networking*. New York, NY, USA: Cambridge University Press, 2009.
- [97] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos. "A survey of multilinear subspace learning for tensor data". In: *Pattern Recognition* 44 (2011), pp. 1540–1551.
- [98] V.-D. Nguyen, K. Abed-Meraim, and N. Linh-Trung. "Fast Tensor Decompositions for Big Data Processing". In: *International Conference on Advanced Technologies for Communications (ATC), Hanoi, Vietnam*. 2016.
- [99] D. Nion and N. D. Sidiropoulos. "Tensor algebra and multidimensional harmonic retrieval in signal processing for mimo radar". In: *IEEE Transactions on Signal Processing* 58 (2010), pp. 5693–5705.
- [100] P. D. Oliveira, C. Fernandes, G. Favier, and R. Boyer. "PARATUCK semi-blind receivers for relaying multi-hop MIMO systems". In: *Digital Signal Processing* 92 (2019), pp. 127–138.
- [101] V. Ollier, M. E. Korso, R. Boyer, and P. Larzabal. "French SKA White Book- The French community towards the Square Kilometer Array". In: *SKA-France, Chapter 4* (2017).
- [102] R. Orus. "A practical introduction to tensor networks: Matrix product states and projected entangled pair states". In: *Annals of Physics* 349 (2014), pp. 117–158.
- [103] I. Oseledets and E. Tyrtyshnikov. "TT-cross approximation for multidimensional arrays". In: *Linear Algebra and its Applications* 432.1 (2010), pp. 70–88.
- [104] I. V. Oseledets. "Tensor-Train decomposition". In: *SIAM J. Scientific Computing* 33.5 (2011), pp. 2295–2317.
- [105] I. V. Oseledets. "Tensor-Train decomposition". In: *SIAM J. Scientific Computing* 33.5 (2011), pp. 2295–2317.
- [106] I. V. Oseledets and E. E. Tyrtyshnikov. "Breaking the curse of dimensionality, or how to use SVD in many dimensions". In: *SIAM J. Scientific Computing* 31 (2009), pp. 3744–3759.
- [107] E. E. Papalexakis, C. Faloutsos, and N. Sidiropoulos. "ParCube: Sparse Parallelizable Tensor Decompositions". In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 10 (2015).
- [108] J. M. Papy, L. D. Lathauwer, and S. V. Huffel. "A Shift Invariance-Based Order-Selection Technique for Exponential Data Modelling". In: *IEEE Signal Processing Letters* 14 (2007), pp. 473–476.
- [109] J. M. Papy, L. D. Lathauwer, and S. V. Huffel. "Exponential data fitting using multilinear algebra: the single-channel and multi-channel case". In: *Wiley Online Library* 12 (2005), pp. 809–826.



- [110] N. Petrochilos and P. Comon. "Link Between the Joint Diagonalisation of Symmetrical Cubes and PARAFAC: An Application to Secondary Surveillance Radar". In: *Fourth IEEE Workshop on Sensor Array and Multichannel Processing*. 2006.
- [111] L. Qi, Q. Wang, and Y. Chen. "Three dimensional strongly symmetric circulant tensors". In: *Linear Algebra and its Applications* 482.Supplement C (2015), pp. 207–220.
- [112] C. Qian, X. Fu, N. D. Sidiropoulos, and Y. Yang. "Tensor-Based Channel Estimation for Dual-Polarized Massive MIMO Systems". In: *arXiv:1805.02223v2* (2018).
- [113] G. Quintana-Orti and E. S. Quintana-Orti. "Parallel codes for computing the numerical rank". In: *Linear algebra and its applications* 275 (1998), pp. 451–470.
- [114] S. Ragnarsson and C. F. V. Loan. "Block tensors and symmetric embeddings". In: *Linear Algebra and its Applications* 438.2 (2013), pp. 853–874.
- [115] M. Rajih, P. Comon, and R. A. Harshman. "Enhanced Line Search: A novel method to accelerate Parafac". In: *SIAM Journal on Matrix Analysis and Applications* 30 (2008), pp. 1128–1147.
- [116] A. Renaux, L. Najjar-Atallah, P. Larzabal, and P. Forster. "A useful form of the Abel bound and its application to estimator threshold prediction". In: *IEEE Transactions on Signal Processing* 55 (2007), pp. 2365–2369.
- [117] C. D. Richmond. "Mean-squared error and threshold SNR prediction of maximum-likelihood signal parameter estimation with estimated colored noise covariances". In: *IEEE Transactions on Information Theory* 52.5 (2006), pp. 2146–2164.
- [118] F. Roemer and M. Haardt. "A closed-form solution for Parallel Factor (PARAFAC) Analysis". In: *In IEEE International Conference on Acoustics, Speech and Signal Processing* (2008).
- [119] T. Rohwedder and A. Uschmajew. "On Local Convergence of Alternating Schemes for Optimization of Convex Problems in the Tensor Train Format". In: *SIAM Journal on Numerical Analysis* 51.2 (2013), pp. 1134–1162.
- [120] Y. Rong, M. Khandaker, and Y. Xiang. "Channel Estimation of Dual-Hop MIMO Relay System via Parallel Factor Analysis". In: *IEEE Transactions on Wireless Communications* 11.6 (2012), pp. 2224–2233.
- [121] Y. Rong, X. Tang, and Y. Hua. "A Unified Framework for Optimizing Linear Nonregenerative Multicarrier MIMO Relay Communication Systems". In: *IEEE Trans. on Signal Process.* 57 (2009), pp. 4837–4851.
- [122] R. Roy and T. Kailath. "ESPRIT-Estimation of signal parameters via rotational invariance techniques". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 37 (1989), pp. 984–995.
- [123] S. Sahnoun, K. Usevich, and P. Comon. "Multidimensional ESPRIT for Damped and Undamped Signals: Algorithm, Computations, and Perturbation Analysis". In: *IEEE Transactions on Signal Processing* 65 (2017), pp. 5897–5910.
- [124] S. Sahnoun and P. Comon. "Joint Source Estimation and Localization". In: *IEEE Transactions on Signal Processing* 63.10 (2015), pp. 2485–2495.
- [125] A. Sandryhaila and J. Moura. "Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure". In: *IEEE Signal Processing Magazine* 31.5 (2014), pp. 80–90.

- [126] N. D. Sidiropoulos. "Generalizing Carathéodory's uniqueness of harmonic parameterization to N dimensions". In: *IEEE Transactions on Information Theory* 47.4 (2001), pp. 1687–1690.
- [127] N. D. Sidiropoulos, R. Bro, and G. B. Giannakis. "Parallel Factor Analysis in Sensor Array Processing". In: *IEEE Transactions on Signal Processing* 48 (2000), pp. 2377–2388.
- [128] N. Sidiropoulos, G. Giannakis, and R. Bro. "Blind PARAFAC receivers for DS-CDMA systems". In: *IEEE Transactions on Signal Processing* 48.3 (2000), pp. 810–823.
- [129] N. Sidiropoulos, E. Papalexakis, and C. Faloutsos. "A parallel algorithm for big tensor decomposition using randomly compressed cubes (PARACOMP)". In: *IEEE International Conference on Acoustics, Speech and Signal Processing, Florence, Italy*. 2014.
- [130] N. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. Papalexakis, and C. Faloutsos. "Tensor Decomposition for Signal Processing and Machine Learning". In: *IEEE Transactions on Signal Processing* 65 (2017), pp. 3551–3582.
- [131] N. D. Sidiropoulos and R. Bro. "On the uniqueness of multilinear decomposition of N-way arrays". In: *Journal of Chemometrics* 14 (2000), pp. 229–239.
- [132] C. D. Silva and F. J. Herrmann. "Optimization on the Hierarchical Tucker manifold – Applications to tensor completion". In: *Linear Algebra and its Applications* 481. Supplement C (2015), pp. 131–173.
- [133] M. Sørensen and L. De Lathauwer. "Multidimensional Harmonic Retrieval via Coupled Canonical Polyadic Decomposition - Part I: Model and Identifiability". In: *IEEE Transactions on Signal Processing* 65 (2016), pp. 517–527.
- [134] M. Sørensen and L. De Lathauwer. "Tensor decompositions with block-Toeplitz structure and applications in signal processing". In: *ASILOMAR 2011* (2011).
- [135] M. Sørensen and L. D. Lathauwer. "Blind Signal Separation via Tensor Decomposition With Vandermonde Factor: Canonical Polyadic Decomposition". In: *IEEE Transactions on Signal Processing* 61 (2013), pp. 5507–5519.
- [136] M. Sørensen and L. D. Lathauwer. "New Uniqueness Conditions for the Canonical Polyadic Decomposition of Third-Order Tensors". In: *SIAM Journal on Matrix Analysis and Applications* 36 (2015), pp. 1381–1403.
- [137] A. Stegeman and N. D. Sidiropoulos. "On Kruskal's uniqueness condition for the Candecomp/Parafac decomposition". In: *Linear Algebra and its Applications* 420.2 (2007), pp. 540–552. ISSN: 0024-3795.
- [138] E. M. Stoudenmire and S. R. White. "Real-space parallel density matrix renormalization group". In: *Physical review B* 87 (2013).
- [139] P. Strobach. "Bi-iteration svd subspace tracking algorithms". In: *IEEE Transactions on signal processing* 45 (1997), pp. 1222–1240.
- [140] G. Tomasi and R. Bro. "A comparison of algorithms for fitting the PARAFAC model". In: *Computational statistics & Data Analysis* 50 (2006), pp. 1700–1734.
- [141] N. D. Tran, A. Renaux, R. Boyer, S. Marcos, and P. Larzabal. "Weiss-Weinstein bound for MIMO radar with colocated linear arrays for SNR threshold prediction". In: *Elsevier Signal Processing* 92 (2012), pp. 1353–1358.
- [142] L. R. Tucker. "Some mathematical notes on three-mode factor analysis". In: *Psychometrika* 31.3 (1966), pp. 279–311.

- [143] S. Ubaru and Y. Saad. "Fast methods for estimating the numerical rank of large matrices". In: *International Conference on Machine Learning*. 2016, pp. 468–477.
- [144] A. Uschmajew and B. Vandereycken. "The geometry of algorithms using hierarchical tensors". In: *Linear Algebra and its Applications* 439.1 (2013), pp. 133–166.
- [145] N. Vannieuwenhoven, R. Vandebril, and K. Meerbergen. "A New Truncation Strategy for the Higher-Order Singular Value Decomposition". In: *SIAM Journal on Scientific Computing* 34 (2012), pp. 1027–1052.
- [146] M. Vasilescu and D. Terzopoulos. "Multilinear subspace analysis of image ensembles". In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Madison, WI, USA*. 2003.
- [147] N. Vervliet, O. Debals, L. Sorber, and L. D. Lathauwer. "Breaking the curse of dimensionality using decompositions of incomplete tensors: Tensor based scientific computing in big data analysis". In: *IEEE Signal Processing Magazine* 31 (2014), pp. 71–79.
- [148] H. Wang and N. Ahuja. "Compact representation of multidimensional data using tensor rank-one decomposition". In: *17th International Conference on Pattern Recognition*. 2004, pp. 44–47.
- [149] C.-K. Wen, J.-C. Chen, and P. Ting. "Robust Transmitter Design for Amplify-and-Forward MIMO Relay Systems Exploiting Only Channel Statistics". In: *IEEE Transactions on Wireless Communications* 11 (2012), pp. 668–682.
- [150] L. R. Ximenes, G. Favier, and A. L. F. de Almeida. "Semi-Blind Receivers for Non-Regenerative Cooperative MIMO Communications Based on Nested PARAFAC Modeling". In: *IEEE Transactions on Signal Processing* 63.18 (2015), pp. 4985–4998.
- [151] L. R. Ximenes, G. Favier, and A. L. F. de Almeida. "Closed-Form Semi-Blind Receiver For MIMO Relay Systems Using Double Khatri-Rao Space-Time Coding". In: *IEEE Signal Processing Letters* 23 (2016), pp. 316–320.
- [152] L. Ximenes, G. Favier, A. L. F. de Almeida, and Y. Silva. "PARAFAC-PARATUCK semi-blind receivers for two-hop cooperative MIMO relay systems". In: *IEEE Transactions on Signal Processing* 62 (2014), pp. 3604–3615.
- [153] C. Xu. "Hankel tensors, Vandermonde tensors and their positivities". In: *Linear Algebra and its Applications* 491. Supplement C (2016), pp. 56–72.
- [154] L. Xu, T. Jing, Y. Longxiang, and Z. Hongbo. "PARALIND-based identifiability results for parameter estimation via uniform linear array". In: *EURASIP Journal on Advances in Signal Processing* (2012).
- [155] Y. Zniyed, R. Boyer, A. de Almeida, and G. Favier. "A TT-based hierarchical framework for decomposing high-order tensors". In: *SIAM Journal on Scientific Computing (SISC)*, submitted (2018).
- [156] Y. Zniyed, R. Boyer, A. L. D. Almeida, and G. Favier. "High-order CPD estimator with dimensionality reduction using a tensor train model". In: *EU-SIPCO, Sep 2018, Rome, Italy*.
- [157] Y. Zniyed, R. Boyer, A. de Almeida, and G. Favier. "High-order tensor factorization via trains of coupled third-order CP and Tucker decompositions". In: *to appear in Linear Algebra and its Applications (LAA)* (2018).

- [158] Y. Zniyed, R. Boyer, A. L. de Almeida, and G. Favier. "Multidimensional Harmonic Retrieval Based on Vandermonde Tensor Train". In: *Elsevier Signal Processing* 163 (2019), pp. 75–86.
- [159] Y. Zniyed, R. Boyer, A. de Almedia, and G. Favier. "Tensor-Train Modeling for MIMO-OFDM Tensor Coding-And-Forwarding Relay Systems". In: *EU-SIPCO conference, accepted in Advances on tensor and multi-dimensional data representation session*. 2019.
- [160] Y. Zniyed, R. Boyer, A. de Almedia, and G. Favier. "Tensor Train Representation of Massive MIMO Channels using the Joint Dimensionality Reduction and Factor Retrieval (JIRAFE) Method". In: *Elsevier, Signal Processing (SP)*, submitted (2019).
- [161] Y. Zniyed, S. Miron, R. Boyer, and D. Brie. "Uniqueness of Tensor Train Decomposition with Linear Dependencies". In: *Gretsi, Lille*. 2019.
- [162] M. D. Zoltowski, M. Haardt, and C. P. Mathews. "Closed-form 2-D angle estimation with rectangular arrays in element space or beamspace via unitary ESPRIT". In: *IEEE Transactions on Signal Processing* 44 (1996), pp. 316–328.



**Titre :** Gérer le fléau de la dimension à l'aide des trains de tenseurs : modèles et algorithmes

**Mots clés :** trains de tenseurs, fléau de dimension, décompositions tensorielles

**Résumé :** Le traitement des données massives, communément connu sous l'appellation "Big Data", constitue l'un des principaux défis scientifiques de la communauté STIC.

Plusieurs domaines, à savoir économique, industriel ou scientifique, produisent des données hétérogènes acquises selon des protocoles technologiques multi-modales. Traiter indépendamment chaque ensemble de données mesurées est clairement une approche réductrice et insatisfaisante. En faisant cela, des "relations cachées" ou des inter-corrélations entre les données peuvent être totalement ignorées. Les représentations tensorielles ont reçu une attention particulière dans ce sens en raison de leur capacité à extraire de données hétérogènes et volumineuses une information physiquement interprétable confinée à un sous-espace de dimension réduite. Dans ce cas, les données peuvent être organisées selon un tableau à  $D$  dimensions, aussi appelé tenseur d'ordre  $D$ .

Dans ce contexte, le but de ce travail est que certaines propriétés soient présentes : (i) avoir des algorithmes de factorisation stables (ne souffrant pas de problème de convergence), (ii) avoir un faible coût de stockage (c'est-à-dire que le nombre de paramètres libres doit être linéaire en  $D$ ), et (iii) avoir un formalisme sous forme de graphe permettant une visualisation mentale simple mais rigoureuse des décompositions tensorielles de tenseurs d'ordre élevé, soit pour  $D > 3$ .

Par conséquent, nous nous appuyons sur la décomposition en train de tenseurs (TT) pour élaborer de nouveaux algorithmes de factorisation TT, et des nouvelles équivalences en termes de modélisation tensorielle, permettant une nouvelle stratégie de réduction de dimensionnalité et d'optimisation de critère des moindres carrés couplés pour l'estimation des paramètres d'intérêts nommé JIRAFE.

Ces travaux d'ordre méthodologique ont eu des applications dans le contexte de l'analyse spectrale multidimensionnelle et des systèmes de télécommunications à relais.

**Title :** Breaking the curse of dimensionality based on tensor train : models and algorithms

**Keywords :** tensor train, the curse of dimensionality, tensor decompositions

**Abstract :** Massive and heterogeneous data processing and analysis have been clearly identified by the scientific community as key problems in several application areas. It was popularized under the generic terms of "data science" or "big data". Processing large volumes of data, extracting their hidden patterns, while performing prediction and inference tasks has become crucial in economy, industry and science.

Treating independently each set of measured data is clearly a reductive approach. By doing that, "hidden relationships" or inter-correlations between the datasets may be totally missed. Tensor decompositions have received a particular attention recently due to their capability to handle a variety of mining tasks applied to massive datasets, being a pertinent framework taking into account the heterogeneity and multi-modality of the data. In this case, data can be arranged as a  $D$ -dimensional array, also referred to as a  $D$ -order tensor.

In this context, the purpose of this work is that the following properties are present: (i) having a stable factorization algorithms (not suffering from convergence problems), (ii) having a low storage cost (i.e., the number of free parameters must be linear in  $D$ ), and (iii) having a formalism in the form of a graph allowing a simple but rigorous mental visualization of tensor decompositions of tensors of high order, i.e., for  $D > 3$ .

Therefore, we rely on the tensor train decomposition (TT) to develop new TT factorization algorithms, and new equivalences in terms of tensor modeling, allowing a new strategy of dimensionality reduction and criterion optimization of coupled least squares for the estimation of parameters named JIRAFE.

This methodological work has had applications in the context of multidimensional spectral analysis and relay telecommunications systems.