Information theoretic aspects of tensor based multi-domain communication systems

Divyanshu Pandey

Department of Electrical & Computer Engineering
McGill University
Montreal, Canada

October 2022

A thesis submitted to McGill University in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Abstract

Most modern communication systems employ several domains for transmission and reception such as space, time, frequency, users, code sequences, and transmission media. Thus, the signals and systems involved in information transfer have an inherent multi-domain structure which can be well represented using tensors. A tensor is a multi-way array which can be seen as a higher order generalization of vectors or matrices. A unified mathematical framework capable of intuitively modelling multi-domain communication systems can be developed with the help of tensors. The use of tensors to characterize, analyze, and build multi-domain communication systems is proposed in this thesis. A generic system model is defined in this work for multi-domain communication systems with N input domains and M output domains. The multi-linear channel between such higher order input and output signals is defined as an order $M+N$ tensor, which couples the input and output through the Einstein product. The suggested framework is generic, where the physical interpretations of the domains can vary depending on the specific system being modelled.

An information theoretic analysis of multi-domain communication systems is considered by deriving the Shannon capacity and input power allocation for a fixed higher order tensor channel under a family of power constraints. Owing to the multi-domain nature of the input signals, the power constraints in multi-domain communication systems can span one or more domains. This thesis demonstrates the tensor framework's ability to mathematically represent a variety of such power constraints. Shannon capacity of tensor channels under such family of power constraints is derived. Water-filling is extended from a matrix setting to higher domains in such a tensor-based formulation, encapsulating the impact of various domains and allowing collaborative multi-domain precoding and power allocation. It is also shown that as the number of domains increases, the multiplexing gain for a tensor channel can increase exponentially, indicating the ability of the tensor-based communication systems to offer the enormous information transmission rates required for beyond 5G systems. In addition, this thesis illustrates how the tensor framework can be used to characterize the capacity and rate regions of multi-user MIMO channels. The tensor-based technique leads to a coordinated users transmission scheme. The tensor framework treats the multi-domain interference terms as information bearing entities, and thus ensures higher achievable sum rates as compared to the independent users transmissions.

Further, the Einstein Product of tensors is used to develop a framework for minimum mean square error (MMSE) estimation for multi-domain signals and data. Both proper and improper complex tensors are addressed by the framework. The traditional linear and widely linear MMSE estimators are extended to the tensor setting, resulting in multi-linear and widely multi-linear MMSE estimation. Further, a relation between the MMSE error covariance tensor and the gradient of the mutual information is extended from a vector setting to tensors, known as the tensor I-MMSE relation. Furthermore, the tensor I-MMSE relation is used to find the capacity of tensor channels when the input is drawn from arbitrary distributions. In the presence of circularly symmetric Gaussian noise and under no constraint on the input constellation, an input drawn from a circularly symmetric Gaussian distribution achieves the channel capacity. However, under practical scenarios, the input is often drawn from discrete signalling constellations which are far from Gaussian distributed. By making use of the tensor I-MMSE relation, an iterative precoder is developed in this thesis which achieves capacity of the tensor channels when the input is limited by the choice of signalling constellations.

Sommaire

La plupart des systèmes de communication modernes recourent à une variété de domaines pour la transmission et la réception d'information, tels l'espace, le temps, la fréquence, les utilisateurs, les séquences de codes et les supports de transmission. Par conséquent, les signaux et les systèmes impliqués dans le transfert d'information possèdent une structure à plusieurs domaines, qui peut être représentée par des tenseurs. Un tenseur est un tableau multidirectionnel qui généralise les vecteurs et les matrices à un nombre supérieur d'indices. Les tenseurs sont au cœur du développement d'un cadre mathématique unifié et intuitif pour la modélisation des systèmes de communications à plusieurs domaines. Le déploiement de tenseurs pour caractériser, analyser et concevoir ces systèmes est proposé dans cette thèse. Un modèle y est défini pour les systèmes de communications comportant N domaines d'entrée et M domaines de sortie. Le canal multilinéaire entre ces signaux d'entrée et de sorties est défini par un tenseur d'ordre $M + N$, reliant les entrées aux sorties par le truchement du produit d'Einstein. Le cadre exposé est général, les interprétations physiques des domaines pouvant varier selon le système spécifique modélisé.

Une analyse fondée sur la théorie de l'information est conduite en établissant la capacité de Shannon et la répartition de la puissance d'entrée pour un canal tensoriel d'ordre fixe soumis à une famille de contraintes relatives à la puissance. Le caractère à plusieurs domaines des signaux d'entrées fait en sorte que ces contraintes s'appliquent à un ou plusieurs domaines. Cette thèse démontre que le cadre tensoriel peut représenter mathématiquement une variété de contraintes relatives à la puissance. L'algorithme du *water-filling* est généralisé à une formulation tensorielle, combinant l'incidence de chaque domaine, le précodage et la répartition de la puissance entre les domaines de manière collaborative. De plus, on montre que le gain de multiplexage pour un canal tensoriel croît exponentiellement à mesure que le nombre de domaines augmente, indiquant la capacité des systèmes de communications déployant les tenseurs à fournir les hauts débits d'information requis par les technologies au-delà du 5G. Cette thèse illustre également comment le cadre tensoriel permet de caractériser la capacité et les régions de capacité de systèmes MIMO à plusieurs utilisateurs. La technique s'appuyant sur les tenseurs mène à un schéma de transmission d'utilisateurs coordonnés. Le cadre tensoriel traite les termes d'interférence multi-domaines en tant qu'entités porteuses d'information, assurant ainsi des *sum rates* réalisables supérieurs comparativement aux transmissions d'utilisateurs indépendants.

En outre, le produit d'Einstein des tenseurs est employé dans le développement d'un cadre d'estimation de l'erreur quadratique moyenne minimale (MMSE) pour les signaux et les données multi-domaines. Les tenseurs complexes propres et impropres sont pris en compte dans ce cadre. Les estimateurs MMSE traditionnels linéaires et largement linéaires sont étendus au cadre tensoriel, donnant lieu à une estimation MMSE multi-linéaire et largement multi-linéaire. Une relation entre le tenseur de covariance de l'erreur MMSE et le gradient de l'information mutuelle est étendue d'un cadre vectoriel aux tenseurs, sous le nom de relation I-MMSE du tenseur. De même, la relation I-MMSE tensorielle est utilisée pour trouver la capacité des canaux tensoriels lorsque l'entrée est tirée de distributions arbitraires. En présence d'un bruit gaussien à symétrie circulaire et sans contrainte sur la constellation de signalisation d'entrée, une entrée tirée d'une distribution gaussienne à symétrie circulaire atteint la capacité du canal. Toutefois, l'entrée en pratique est souvent tirée de constellations discrètes qui sont loin d'être distribuées normalement. En mettant à profit la relation I-MMSE tensorielle, un précodeur itératif est développé dans cette thèse et atteint la capacité des canaux tensoriels lorsque l'entrée est limitée par le choix des constellations de signalisation.

Acknowledgments

I believe that no personal accomplishment is completely ‘individual’, and in that spirit would like to acknowledge the multitude of factors and people involved that have led me towards this personal and professional milestone.

First and foremost, my deepest gratitude goes to my supervisor Prof. Harry Leib for his guidance. His diligent approach towards research and technical insights are truly inspiring. This thesis wouldn’t exist without his remarkable patience of reviewing countless iterations of my technical reports and paper drafts. His continuous push towards quality research helped me enhance my abilities significantly. In particular, given the unusual and distressing circumstances that the COVID-19 pandemic created, it was his encouragement that allowed me to persist and keep moving forward while ensuring my physical and mental well-being.

I am also thankful for the financial assistance that Prof. Harry Leib provided through his Natural Sciences and Engineering Research Council of Canada (NSERC) research grant RGPIN-2016-03647 for the project entitled “Tensor modulation for space-time-frequency communication systems”. Further, I am also grateful to the Faculty of Engineering, McGill University, for their financial support in the form of McGill Engineering Doctoral Award (MEDA) for the first three years of my studies, and the Graduate Research Enhancement and Travel (GREAT) awards for conference expenses.

I would also like to thank my Ph.D. committee members Prof. Ioannis Psaromiligkos and Prof. Tim Hoheisel for their constructive feedbacks and evaluation during the committee meetings. In addition, I would like to sincerely thank the internal and external examiners of this thesis, Prof. Benoit Champagne and Prof. Xiaodai Dong (University of Victoria) respectively, for their constructive comments and feedback which helped improve the thesis. I am also grateful to all my colleagues in the Telecommunications and Signal Processing (TSP) lab throughout my studies for providing a rich learning environment. In particular, a special mention goes to Adithya Venugopal whose help was instrumental in providing me an expedited start in the field of Tensor Algebra. My research benefited a lot from several technical discussions with him. Also, thanks to Kenz Zaghib for helping with the French translation of the Abstract.

Given that a significant time of my Ph.D. was spent working from home due to the pandemic, I must thank my room-mates Shantanil and Shubhankar for providing the best company that I could hope for in those times of despair. I am glad to have made friends with them and also Ameya, Nikhil, Shikha, Aditya, and Asil who are all like extended family to me in Montreal. Some friends from a far also played an important role in keeping me motivated and were there to lend me an ear whenever needed. For that, I am always grateful to Niranjan, Arvind, Krishna, Ayush, and Sidhant.

Most importantly, the role that my entire family has played in this endeavour must be acknowledged. The unconditional care, encouragement and prayers of my parents, Braj Ballabh Pandey and Pushpa Pandey, have always been my driving force. My wife Prachi, with her unwavering trust in my abilities and aspirations, has always boosted my confidence. The love and wishes of my other family members, Priyanka, Shishir, Priyanshu, Jyoti and Pratyush always kept me joyful and hopeful. My brother Priyanshu in particular deserves a special mention since he has been my pillar of support in every possible manner. I have been able to take decisions regarding my life and career without any hesitations only knowing that he has got my back.

Lastly, I would like to thank Prof. Vishwanath Sinha, former director of the LNMIIT-Jaipur (my undergraduate alma mater), for being a role model and inspiring me to pursue a Ph.D. in the first place.

Contents

List of Figures

List of Tables

List of Acronyms

Acronym	Full form	First used on Page.
3GPP	Third Generation Partnership Project	3
APSK	Amplitude Phase Shift Keying	185
BER	Bit Error Rate	159
BPSK	Binary Phase Shift Keying	159
BS	Base Station	46
CDIR	Channel Distribution Information at the Receiver	11
CDIT	Channel Distribution Information at the Transmitter	11
CDMA	Code Division Multiple Access	8
CP	Canonical Polyadic	6
CSI	Channel State Information	11
CSIR	Channel State Information at the Receiver	11
CSIT	Channel State Information at the Transmitter	11
DFT	Discrete Fourier Transform	3
DS-CDMA	Direct Sequence- Code Division Multiple Access	8
DSL	Digital Subscriber Line	52
EVD	Eigen Value Decomposition	7
FBMC	Filter Bank Multi-Carrier	4

Acronym	Full form	First used on Page.
FFT	Fast Fourier Transform	3
GFDM	Generalized Frequency Division Multiplexing	4
GPU	Graphics Processing Unit	39
HOBG	Higher Order Bi-conjugate Gradient	36
HOSVD	Higher order Singular Value Decomposition	7
IBC	Interfering Broadcast Channel	46
IC	Interference Channels	124
ICI	Inter-Carrier Interference	45
I-MMSE	Mutual Information - Minimum Mean Squared Error	13
IoT	Internet of Things	2
IRS	Intelligent Reflecting Surfaces	9
KKT	Karush-Kuhn-Tucker	13
LMMSE	Linear Minimum Mean Squared Error	144
M2M	Machine-to-Machine	2
MA	Multiple Access	5
MAC	Multiple Access Channels	114
MIMO	Multiple-Input Multiple-Output	2
MMSE	Minimum Mean Squared Error	15
MSE	Mean Squared Error	137
MU	Multi-User	45
NM	Newton's Method	36
OFDM	Orthogonal Frequency Division Multiplexing	3
PAPR	Peak to Average Power Ratio	4

Acronym	**Full form**	**First used on Page.**
PARAFAC	Parallel Factorization	6
PCA	Principal Component Analysis	6
PD-NOMA	Power Domain Non-Orthogonal Multiple Access	5
PSK	Phase Shift Keying	185
QAM	Quadrature Amplitude Modulation	160
QPSK	Quadrature Phase Shift Keying	185
RC	Raised Cosine	110
RRC	Root Raised Cosine	111
SCMA	Sparse Coded Multiple Access	5
SISO	Single Input Single Output	11
SNR	Signal to Noise Ratio	14
SVD	Singular Value Decomposition	7
TL	Tensor multi-linear	148
TN	Tensor Network	30
TSTF	Tensor Space-Time-Frequency	9
TT	Tensor Train	6
TWL	Tensor widely multi-linear	148
UFMC	Universal Filtered Multi-Carrier	4
VLEO	Very-low Earth Orbit	2
WLMMSE	Widely Linear Minimum Mean Squared Error	138

Notations

$(.)^{-1}$	Inverse of the argument
$(.)^{*}$	Complex Conjugate of the argument
$(.)^{H}$	Conjugate transpose of the argument
$(.)^{T}$	Transpose of the argument
$\|\|\cdot\|\|_F$	Frobenius norm of the argument
$\mathbb{C}$	set of complex numbers
$\mathbb{R}$	set of real numbers
$\underline{a} \in \mathbb{C}^{N\times 1}$	Deterministic column vector $\underline{\mathrm{a}}$ of size N
$\mathrm{A} \in \mathbb{C}^{I\times J}$	Deterministic Matrix A of size $I \times J$
$\mathcal{A} \in \mathbb{C}^{I_1\times\ldots\times I_N}$	Deterministic Tensor $\mathcal{A}$ of size $I_1 \times \ldots \times I_N$
$\underline{\mathbf{a}} \in \mathbb{C}^{N\times 1}$	Random column vector $\underline{a}$ of size N
$\mathbf{A} \in \mathbb{C}^{I\times J}$	Random Matrix A of size $I \times J$
$\boldsymbol{\mathcal{A}} \in \mathbb{C}^{I_1\times\ldots\times I_N}$	Random Tensor $\mathcal{A}$ of size $I_1 \times \ldots \times I_N$
$\mathcal{A}_{i_1,\ldots,i_N}$	Individual elements of $\mathcal{A}$ denoted by indices in subscript
$\mathcal{A}_{:,i_2,\ldots,i_N}$	All elements of first mode corresponding to fixed other modes
$\ddot{\underline{x}}, \ddot{\mathrm{X}}, \ddot{\mathcal{X}}$	Augmented vector, matrix and tensor respectively
$\mathcal{A}^{(n)}$	The nth tensor in a sequence of tensors
$\underline{i}$	sequence of indices $(i_1, \ldots, i_N)$
$\mathcal{A} *_N \mathcal{B}$	Einstein Product over N common modes of $\mathcal{A}$ and $\mathcal{B}$
$\mathcal{A} \circ \mathcal{B}$	Outer product of $\mathcal{A}$ and $\mathcal{B}$
$\langle \mathcal{A}, \mathcal{B} \rangle$	Inner product of $\mathcal{A}$ and $\mathcal{B}$
$\mathrm{A} \otimes \mathrm{B}$	Kronecker product of A and B
$\mathcal{A} \times_n \mathrm{B}$	mode-n product between tensor $\mathcal{A}$ and matrix B
$\mathcal{I}_N$	Identity tensor of order $2N$

$\mathbb{E}[\cdot]$	Expected value
$\mathrm{cov}(\cdot)$	Covariance
$p_{\boldsymbol{\mathfrak{X}}}(\cdot)$	probability distribution of $\boldsymbol{\mathfrak{X}}$
$p_{\boldsymbol{\mathfrak{X}}\vert\boldsymbol{\mathfrak{Y}}}(\cdot)$	probability distribution of $\boldsymbol{\mathfrak{X}}$ given $\boldsymbol{\mathfrak{Y}}$
$\mathcal{H}(\cdot)$	Entropy
$\mathcal{I}(\cdot,\cdot)$	Mutual Information
$0_{\mathfrak{I}}$	All zero tensor
$\det(\cdot)$	Determinant
$\mathrm{tr}(\cdot)$	Trace
$\mathrm{numel}(\mathfrak{X})$	Number of elements in $\mathfrak{X}$
$\mathcal{A} \succeq 0$	Positive semi-definite tensor
$\mathcal{A} \succ 0$	Positive definite tensor
$\nabla_{\mathfrak{X}} f$	Gradient of function f with respect to tensor $\mathfrak{X}$
$\Re\{\cdot\}$	Real part of argument
$\Im\{\cdot\}$	Imaginary part of argument.

Chapter 1

Introduction

Most modern communication systems are inherently multi domain in nature where the transmission and reception spans across various domains such as space, time and frequency. Designing modulation schemes and signal processing techniques which can simultaneously benefit from the distinct nature of all the domains is a challenging task in emerging communication systems. To this end, a mathematical framework is much required which facilitates integrating all the available domains of transmission and reception into the system model in an intuitive, interpretable and structured manner. Modelling such multiple domains in a unified mathematical framework can help exploit the full potential of all the available resources while accounting for their mutual effects. Such a system representation can be suitably developed using tensors and tools from multi-linear algebra can be employed to devise suitable signal processing techniques.

Tensors are multi-way arrays which have found widespread applications in various science and engineering disciplines. This thesis focusses on the modelling of multi-domain communication systems using a tensor framework, the introduction of the tensor channel and the evaluation of its Shannon capacity under different constraints. In order to establish the motivation behind employing tensors for representation of input, output, and channel in multi-domain communication systems, the following section is dedicated to a detailed literature review on multi-domain communication systems, and tensor applications in related fields. Further, a brief summary of known results for the capacity of Multiple-Input Multiple-Output (MIMO) matrix channels is also included.

1.1 Literature Review

1.1.1 Multi-Domain Communication Systems

As Internet of Things (IoT) and Machine-to-Machine (M2M) communications gain prominence, we can expect a massive increase in the number of devices requiring wireless connectivity along with extremely high data rates. Ensuring seamless connectivity of such a large number of devices presents some significant challenges for modern communication systems and networks such as 5G and beyond. To meet such challenges, various technologies are being proposed including large MIMO [1], millimeter-wave [2], non-orthogonal multiple-access schemes [3, 4], and network densification [5]. These trends create the need for communication systems with transceivers that incorporate a diverse combination of domains. The word 'domain' refers to any available resource at the transmitter or receiver, and can incorporate a variety of parameters such as antennas, time slots, frequency bins, code sequences, users, to name a few. Henceforth, we refer to a communication system as a multi-domain communication system if the signal transmission or reception spans across more than one domain.

With 5G systems already being implemented in parts of the world, the current research is driving the narrative towards beyond 5G and 6G systems [6, 7]. The 6G systems aim to achieve more than three times the spectral efficiency, and more than ten times the energy efficiency as compared to 5G [8]. Also, one of the primary requirements of 6G is to deliver a data rate of more than 1 Tbps, for which there are suggestions to move from wireless radio communications to optical free space communications for indoor environments [9, 10]. In addition, the vision for 6G also includes an altogether new communications infrastructure bringing access points and cloud functionalities on drones and Very-low Earth Orbit (VLEO) satellites. Such systems aim to incorporate ubiquitous connectivity with very high data rates making use of the sub-THz spectrum and visible light spectrum as well [11]. This clearly paves the path for communication systems where the nature of transmission media is going to be much more evolved than what we have today. The physical layer for a specific system will no more be just restricted to only wired, wireless or optical mode, but a combination of any possible transmission media depending on the use-case. Addressing physical layer issues, such as modulation, coding, waveform selection, equalization etc., while keeping in mind the gamut of transmission domains available, will be a major challenge. Accounting for the distinct features of such domains and their mutual effects in the design process of future communication systems will be crucial.

When it comes to the physical layer design of a multi-domain communication system, one of the primary concerns is the selection of an appropriate waveform. Since most practical systems are constrained in both bandwidth and power, for any wireless communication system low transmit power with efficient use of spectrum is a major requirement which has to be considered while designing a waveform. A detailed survey of various multi-carrier candidate waveforms considered for 5G and beyond systems can be found in [12]. Among all the waveforms, Orthogonal Frequency Division Multiplexing (OFDM) has been one of the most popular multi-carrier schemes so far and has been extensively used with MIMO in 4G standards and Wi-Fi [13], and now also for 5G [14]. In OFDM, the data stream is divided and sent through multiple data sub-carriers or frequency sub-bands within allocated bandwidth, all orthogonal to each other to avoid interference, allowing for a simplified receiver structure over frequency selective channels. With a few modifications like scalable sub-carrier spacing, OFDM with cyclic prefix has recently been approved for 5G NR air interface as well in Third Generation Partnership Project (3GPP) Release 15 [14]. Power efficiency of OFDM can be improved by preprocessing the input symbols through a Discrete Fourier Transform (DFT) block before sub-carrier mapping at the transmitter, thereby making the resulting waveform behave like a single carrier. Such a waveform is called DFT-spread-OFDM [15] and it is supported in 5G for enhanced mobile broadband.

Despite its benefits such as robustness against multipath channels, easy implementation of Fast Fourier Transform (FFT) algorithms and convenient integration with MIMO, OFDM has its shortcomings as it suffers from low spectral efficiency due to cyclic prefix, sensitivity to carrier frequency offset, high Peak to Average Power Ratio (PAPR) and high out of band emissions [16] which warrants a necessity to explore other alternative waveforms which can provide better spectral efficiency. One such promising multi-carrier modulation scheme alternative is Filter Bank Multi-Carrier (FBMC) [17] where each sub-carrier is shaped by a prototype filter to suppress the side lobes of the signals. Another alternative of OFDM, initially proposed for 5G, is Generalized Frequency Division Multiplexing (GFDM) [18] which is a non-orthogonal multi-carrier scheme. In GFDM, the data is divided into two-dimensional time frequency grid introducing flexible pulse shaping for individual sub-carriers and reducing the amount of cyclic prefix. The sub-carrier filtering reduces the out-of-band leakage and makes it suitable for fragmented spectrum applications or cognitive radio scenarios. Another important candidate scheme contesting for beyond 5G wireless communication systems is the Universal Filtered Multi-Carrier (UFMC) [19],[20]. A detailed comparative analysis of of cyclic prefix -OFDM, FBMC, GFDM, UFMC and other multi-carrier schemes along with their pros and cons can be found in [21],[22].

While most of the proposed schemes were initially designed using time and frequency domains only, their integration with MIMO has to be an essential feature for their acceptance and implementation in modern wireless communication systems. The non-orthogonality of such schemes introduces an intrinsic interference and thus does not give a straight forward integration with the MIMO systems unlike in the case of MIMO-OFDM [23, 24]. That is why MIMO in conjunction with multi-carrier techniques such as MIMO-GFDM [25, 26], MIMO-FBMC [27], MIMO-UFMC [28], etc., has been extensively researched over past few years. Techniques have been developed to improve link reliability through space-time, space-frequency, and space-time-frequency coding methods [29] that exploit diversity in all the spatial, temporal and frequency domains. Thus it is evident that along side time and frequency, space as a domain has to be incorporated in the system model for efficiently extracting the benefit of all the available resources in the design process of new modulation schemes. With any of the multi-carrier schemes combined with MIMO, the received and transmitted signals have an inherent multi-domain structure which can be mathematically represented using multi-way arrays, more commonly known as tensors. The domains in a communication system need not be restricted to space, time, and frequency but can include other parameters such as users, propagation delay, spreading sequence, etc. depending on the specific system. For instance, enhancing Multiple Access (MA) schemes will be an important feature for beyond 5G systems [30]. Spectral efficiency can be increased with schemes like Power Domain Non-Orthogonal Multiple Access (PD-NOMA) or Sparse Coded Multiple Access (SCMA) [31]. Thus, it could be useful to incorporate users or code sequences as additional domains in a communication system while designing the transceiver schemes. Therefore, no matter which of these numerous waveform schemes being discussed for future communication systems triumphs and takes the mantle for meeting the high requirements of beyond 5G systems, it is certain that we need a system model which allows an integration over multiple domains of transmission and reception. The associated signal processing and coding involved at the transceivers to use such modulation schemes is invariably going to span more than one domain of communications. This necessitates a generic unified mathematical framework with which we can model any multi-domain communication system, and hence tensors naturally come into play. Having such a framework would not only provide a mathematical set up for the existing schemes, but would also act as a stepping stone for developing new and improved schemes spanning multiple domains. Since tensors provide a backbone for such a multi-domain system representation, next we present a brief review of tensors and their applications across various fields primarily in communications and signal processing.

1.1.2 Tensors and their Applications

A tensor is a multi-way array that can be seen as an Nth order generalization of a vector or a matrix, where a vector is a tensor of order one and a matrix is a tensor of order two [32]. Tensors were introduced in the early nineteenth century with applications in Physics [33]. Later, tensors found applications in Psychometrics in the sixties with the work of Tucker [34] as an extension of two-way data analysis to higher-order datasets, and in Chemometrics in the eighties [35, 36]. In the last few decades, tensors as an extension of matrices have found extensive applications in various engineering disciplines including computer vision

[37, 38], data mining [39, 40], machine learning [41], neuroscience [42], signal processing [43, 44] and multi-linear system theory [45, 46]. Tensors provide a unified and intuitive framework to represent processes with dependencies on more than two variables. Through a tensor based approach, we can develop models which capture interactions between various parameters enhancing the understanding of their mutual effects. A detailed summary of tensor algebra results and their applications can be found in many recent publications such as [32, 40, 47, 41, 48, 49, 50].

Tensor decompositions have been an area of extensive study because of its varied applications. One of the most commonly used factorization was developed independently under two names, in the form of Canonical Decomposition by Carroll and Chang [51] and Parallel Factorization (PARAFAC) by Harshman [52]. The factorization is now popularly referred as Canonical Polyadic (CP) decomposition [32] where a tensor is decomposed as a linear combination of rank one tensors. Uniqueness of CP decomposition for higher order tensors under certain conditions [53] has led to many applications particularly in signal processing for Blind Source Separation [54],[55]. Another widely used decomposition is the Tucker decomposition first introduced by Ledyard Tucker in [56] for the purpose of higher-order Principal Component Analysis (PCA). It decomposes a tensor into a core tensor transformed by a matrix along each mode known as the factor matrices. It has applications in finding low rank structures, classification, and feature extraction in high dimension data [57, 32]. A combination of the aspects of the CP and Tucker decomposition is known as PARALIND decomposition, which was proposed in [58], and was used for cases having linear dependencies in one or more modes of the tensor. Another decomposition combining the CP and Tucker decomposition was suggested in [59] known as PARATUCK2 for third order tensors with applications in Psychometrics. Further, the generalized PARATUCK2 decomposition with application in blind receiver design in space-time-frequency MIMO communication systems was proposed in [60]. Later on, a more general model for constrained PARAFAC decomposition modelling PARATUCK and PARALIND in a unified framework with applications in signal processing was developed in [61]. Furthermore, Tensor Train (TT) decomposition of tensors has been extensively used in Big Data applications for reducing storage complexity. The TT decomposition breaks a higher order tensor into a set of sparsely connected lower order tensors, known as the core tensors. The low rank structure of the core tensors is exploited to reduce storage complexity [62].

The most popular and widely used decompositions in case of matrices are the Singular Value Decomposition (SVD) and Eigen Value Decomposition (EVD). To consider their extensions to higher order tensors, there is no single generalization of singular value or eigenvalue that preserves all the properties of the matrix case [63],[64]. The most commonly used generalization of the matrix singular value decomposition is known as the Higher order Singular Value Decomposition (HOSVD) which is basically the same as Tucker decomposition for higher order tensors [65]. Similarly as a generalization to matrix eigenvalues to tensors, several definitions exist in literature for tensor eigenvalues [66]. But most of these definitions apply to super-symmetric tensors which restricts to a class of tensors that are invariant under any permutation of their indices. Such an approach has applications in Physics and Mechanics [67]. More recently, in order to solve a set of multi-linear equations using tensor inversion, a specific notion of tensor SVD and EVD was introduced in [48] using the Einstein product of tensors. It is shown in [48] that a tensor group endowed with the Einstein product is structurally similar or isomorphic to a general linear group of matrices. Through this property, [48] establishes a singular value decomposition that decomposes the tensor into the Einstein product of a core tensor along with two unitary tensors. The tensor SVD and EVD proposed in [48] can be seen as a specific case of the Tucker decomposition. While strict diagonalization of the core tensor is not possible in Tucker decomposition [32], the proposed tensor SVD in [48] generates a core tensor which has certain interesting structural properties that can be referred to as pseudo-diagonal. In [48], the idea of tensor SVD and EVD is presented only for fourth order tensors with symmetric mode lengths. The idea of SVD from [48] is further generalized for any even order tensor irrespective of symmetry in mode length in [68]. Similarly, [69] generalizes the idea of tensor EVD to any even order tensor with applications in image processing. The notion of equivalence between the Einstein product of tensors and the corresponding matrix product of the transformed tensors is very crucial and relevant as it helps in developing several tools and concepts from linear algebra such as matrix inverse, ranks, determinants, etc., to tensors without having to explicitly transform a tensor into a matrix or a vector [70].

In the past two decades, the idea to use tensors for modelling wireless communication systems has gained much attention to improve system performance and analysis as it allows a consolidated representation of multiple signalling domains. In the spirit of multi-user, multi-carrier, multi-antenna systems along with different signal processing strategies across time, code, etc., more domains need to be introduced in the line of transmission and reception to model and counter the effect of all the possible sources of interference. A common approach has been to combine multiple domains and treat the transmit and receive signals as concatenated vectors and the channel as a matrix. However, in doing so the natural structure of the signal becomes obscured in the model and we lose the distinction between domains. Thus it makes more sense to use tensors for signal representation and employ tools from multi-linear algebra for the design and analysis of transceiver schemes. One of the initial applications of tensor in wireless communications was proposed for a Direct Sequence- Code Division Multiple Access (DS-CDMA) system in [71] where the received signal is mapped into a third order tensor with code, spatial, and temporal domains and information is extracted from the received signal using PARAFAC decomposition. It was shown in [72] that the blind multi-user separation and equalization can be solved using tensor decompositions of third order tensors for Code Division Multiple Access (CDMA) and over-sampled systems. Another application of third order tensor decomposition has been shown for channel estimation and data recovery in MIMO spread spectrum systems [73]. A tensor space-time-frequency coding technique for a MIMO OFDM CDMA system is developed in [74] where a fifth order coding tensor is used to generate a fifth order transmit signal. The five domains correspond to transmit antennas, data stream, sub-carriers, time blocks, and chips. The baseband equivalent received signal is also represented as a fifth order tensor which under some suitable assumptions on the channel admits a generalized PARATUCK model. Based on this model, semi-blind receivers are then used for joint symbol and channel estimation. Further, a unified model is proposed in [75] to represent eight different schemes of tensor based MIMO wireless communication systems by showing them to be the specific cases of the Tensor Space-Time-Frequency (TSTF) scheme. Tensors have also found applications in cooperative communications where tensor based semi-blind receivers have been developed for channel estimation and symbol detection in relay assisted communication systems [76],[77]. More recently, tensors are also being considered for mathematical modelling of several key technologies associated with beyond 5G and 6G communication systems such as Intelligent Reflecting Surfaces (IRS) [78], massive MIMO [79, 80], millimeter wave [81, 82], MIMO relay systems [83]. Recently, the contracted product of tensors and tensor inversion has been used in [84] to develop joint-domain equalization at the receiver to combat inter-domain interferences in multi-domain communication systems.

A short summary of the commonly used tensor tools along with some of their applications is provided in Table 1.1.

Table 1.1: Tensor Tools and Applications.

Tensor Tool	**Example of Applications**
PARAFAC (CP)	Model received signal in DS-CDMA and develop blind receiver methods [71]
Tucker Decomposition	Data mining, Computer Vision, finding low rank structures in high dimensional data [57]
PARATUCK	Semi blind receivers for joint channel estimation and data detection in MIMO OFDM CDMA systems [74]
Tensor Train Decomposition	Reducing storage complexity in Big Data applications [57], space-time coding for MIMO OFDM relay systems [85]
Tensor Inversion	Joint multi-domain equalization in systems such as MIMO GFDM [84]
Tensor EVD using the Einstein Product	Multi-linear controls system theory [46], Image Processing [69]
Block Constrained PARAFAC	Blind multi-user detection and equalization for oversampled, DS-CDMA and OFDM systems [86]
PARAFAC with Linear Dependencies (PARALIND)	Blind receiver for MIMO OFDM in the presence of carrier frequency offset [87]

With all these applications and many more [88], the use of tensors for signal processing in wireless communication systems has garnered a lot of attention in the past few years. However, most of the focus has been around signal processing and to the best of our knowledge, not much attention has been given to characterizing the channel in a communication system as a higher order tensor and approaching the subject matter from an information theoretic point of view which is the main thrust of this thesis. We aim to quantify capacity for higher order tensor channels. The initial motivation to use tensors for our purpose stems from their unique suitability to retain the distinction between multiple domains in the system model, thereby allowing a convenient representation of a variety of power constraints across domains. Even though tensor entities occur naturally in multi-domain communication systems, it should be noted that in principle a tensor can be represented using a matrix or a vector. For instance, the slices of a third order tensor can be stacked together to form a bigger matrix. Such matrix representations are sometimes used in order to leverage the well established linear algebra concepts for analysis. However, representing a naturally occurring higher order tensor using a lower order array such as a matrix or vector collapses the distinct multiple indices which are used to identify the domains. Thus in order to restore the identifiability of domains, it becomes imperative to use tensors [89]. Retaining the identifiability of domains is crucial such that any domain specific constraints can be incorporated in the mathematical framework as demonstrated in this thesis.

In the existing literature, capacity analysis for a MIMO matrix channel which characterizes mostly space domain has been extensively studied. The MIMO matrix based system model can be seen a specific case of the more general tensor based system model that is proposed in this thesis. Therefore, it is important to look at the different results available in the literature for the capacity of MIMO matrix channels, so as to ensure that we can show those results as a degenerate case of the results corresponding to tensor channels. Hence comparing the tensor and matrix channel results would not only emphasize the advantages of the tensor based schemes, but would also act as a sanity check and validation for the results presented in this thesis. For this purpose, in the next section, we present a brief review of some important information theoretic results associated with the MIMO matrix channels.

1.1.3 Capacity for MIMO channels

Initial results predicting the gain in channel capacity by employing multiple antennas at the transmitter and receiver were presented by Telatar [90] and Foschnini [91] in the late 90's. Telatar proposed in [90] that a MIMO matrix channel can be converted into parallel, non-interfering Single Input Single Output (SISO) channels through a singular value decomposition of the channel matrix with gains corresponding to the singular values of the channel matrix. It was also established in [90] that the optimum input distribution which achieves the capacity is circularly symmetric complex Gaussian and the optimum power allocation is performed using the classical water-filling approach on the decomposed channels. Telatar's work has acted as a stepping stone over the late 90's and early 2000's for several other contributions in this field. Since then, a significant research effort has been invested in extending Telatar's work and developing its variations for more practical scenarios, and under different assumptions on Channel State Information (CSI).

Telatar considered the cases for fixed channels with perfect Channel State Information at the Transmitter (CSIT) and for flat Rayleigh fading channels with perfect Channel State Information at the Receiver (CSIR) [90]. Majority of the ensuing work in this field after [90] considered variations of the channel state information availability at the transmitter or receiver. If the transmitter or the receiver does not know the CSI perfectly but rather have only a partial knowledge of the channel statistics, then such a case is referred to as having Channel Distribution Information at the Transmitter (CDIT) or Channel Distribution Information at the Receiver (CDIR) respectively. Telatar assumed a zero-mean spatially white channel model at the transmitter for the CSIT and CSIR case. Later, the mutual information optimization problems for CSIR and CDIT situation where the channel mean and channel covariance feedbacks are separately available from the receiver to transmitter were solved in [92]. However [92] assumed only a single receive antenna. The work in [92] was later extended in [93] for multiple transmit and multiple receive antenna case with covariance feedback, where it was concluded that the optimum transmit strategy involved transmitting along the eigenvectors of the channel covariance matrix. However, in its system model, [93] considered breaking the channel matrix into the product of a spatially white zero mean channel matrix and a transmit correlation matrix only, thereby assuming one-sided correlation. Later on, [94] considered the same case but assuming receive correlation as well and showed that the receive correlation matrix does not affect the eigenvectors of the transmit covariance matrix. It is interesting to note that [92], [93] and [94], under incremental assumptions in the system model, reached at similar conclusions. A two-sided correlated MIMO channel was handled in [95] by expressing the ergodic capacity of the channel in terms of a hyper-geometric function with matrix arguments providing close form solutions for some asymptotic cases. Their work generalizes the capacity expression by Telatar from [90] and water-filling power allocation for correlated MIMO channels. For the CDIT and CDIR case, [96] considered the zero-mean spatially white model and it was concluded that increasing the number of antennas beyond the number of symbol periods in the channel coherence interval does not help with increasing the capacity. However, later it was contradicted in [97] which showed that if we do not assume zero mean spatially white channel, then in CDIT and CDIR case, spatial correlation at the transmit side can actually benefit the capacity.

Several MIMO communication systems often come along with large transmission bandwidth in which case it is not very realistic to assume narrowband MIMO transmissions. For frequency selective transmissions, where the channel coherence bandwidth is assumed smaller than the typical transmission bandwidth, the key idea is to divide the channel bandwidth into parallel flat fading channels and construct an overall block-diagonal channel matrix with the diagonal blocks given by the channel matrices corresponding to each of the sub-channels. This approach essentially concatenates the transmit vector corresponding to the different antennas and sub-carriers. Using this approach, an ergodic capacity expression for OFDM based spatial multiplexing system is provided in [98] by resorting to asymptotic analysis. It is also established in [98] that unlike the SISO case, frequency selectivity may give advantage in terms of ergodic capacity as compared to the flat fading assuming that the delay paths increase the total angle spread. In contrast with the Rayleigh flat fading case, closed form expressions are not available for frequency selective case, and the capacity is generally computed numerically. In [99], instead of relying on

Monte-Carlo simulations, another iterative water-filling technique is proposed to optimize the approximation of the average mutual information in frequency selective channels for large number of transmit and receive antennas.

Within the CSIT scenario, one important consideration has been finding the channel capacity under per antenna power constraints instead of sum power constraint. Such per antenna power constraints emerge in several practical MIMO systems since each antenna may be connected to a separate power amplifier with finite dynamic range on the individual RF chain. Another scenario where such constraints are common is distributed MIMO which has transmit antennas located at different physical locations and do not share the same power source [100]. For MIMO channels, capacity with per antenna power constraints was first considered in [101] where the capacity optimization problem was formulated in a semi-definite program framework and a solution was proposed using the Karush-Kuhn-Tucker (KKT) conditions for optimality. Furthermore, using similar approach mixed power constraints were considered in [102] which leads to a matrix field water-filling solution for optimum power allocation.

With no constraints on the choice of input distribution, most of the work for channel capacity takes input to be circularly symmetric complex Gaussian distributed as this distribution is known to be the entropy maximizer for a given covariance [103, 90]. However, in a practical communication system, more often the input is drawn from discrete signalling constellations and hence is not Gaussian. Such cases are often handled by exploiting the Mutual Information - Minimum Mean Squared Error (I-MMSE) relationship proposed in [104] for scalar and vector inputs. In [105], a power allocation policy was proposed that maximizes the mutual information for any arbitrary input distribution for diagonal, i.e. parallel non-interfering MIMO Gaussian channels leading to a mercury water-filling approach of power allocation. The mercury level corresponds to the non-Gaussianity of the input distribution. For Gaussian distributed signal, the mercury level is zero and the approach reduces to classical water-filling. This work was further extended in [106] for interfering channels where the mercury level accounts not only for the non-Gaussian input distributions, but also for the interference among the input terms. A linear precoder was proposed in [106] using the relation between the gradient of mutual information with respect to the channel and the minimum mean squared error matrix derived in [107]. Further, [108] presents iterative algorithms using a quadratic function of the precoder matrix to compute a linear precoder which maximizes the mutual information for finite alphabet input. Instead of finding a precoder, [109] presents an iterative method to directly find the optimum input covariance and characterize it in high and low Signal to Noise Ratio (SNR) regimes. More recently, [110] considers the power allocation problem for parallel Gaussian channels with input distributions which are close to Gaussian in the Kullback-Leibler divergence, leading to a robust water-filling. Also, [111] presents MIMO transmission strategy under discrete input signal constraints where a linear precoder and non-uniformly distributed input signals are jointly optimized. A more detailed summary of results pertaining to capacity limits of MIMO channels where input signals are drawn from finite constellations can be found in [112].

Having reviewed the various aspects of multi-domain communication systems, tensors, and the capacity of MIMO channels, we now present the thesis statement in the following section.

1.2 Thesis Statement

In this section, we present the objectives that motivated the research presented in this thesis, our original contributions, and an organization of this thesis.

1.2.1 Thesis Motivation and Objectives

The primary objective of this research is to derive limits of information transmission capabilities for multi-domain communications channels characterized by tensors. To this end, the presented research aims to introduce a unifying tensor-based mathematical framework for multi-domain communication systems. The channel when treated as a higher order tensor generalizes the notion of MIMO channels to higher domains. Through an information theoretic analysis of such a tensor channel, this research aims to explore the impact of different channel sizes, and several input power and constellation constraints, on the channel capacity. Note that the analysis in this thesis focusses on the case where the channel is deterministic, and is known at both the transmitter and the receiver. Since a tensor representation preserves the natural structure of the signals and systems involved, we aim to exploit the multi-domain structure of the input signals to mathematically describe a family of input power constraints, such as sum power or per antenna or per element power constraints. The presented research intends to highlight the benefits involved in encompassing all the system parameters in a single framework using tensors which gives a convenient way of handling inter-domain interferences. Also, this research aims to develop a framework for estimation of multi-domain signals and data, which can be used to characterize the covariance of the estimation error as a higher order tensor. By exploiting a connection between the mutual information and the error covariance tensor of a Minimum Mean Squared Error (MMSE) estimator, this research aims to find the capacity of tensor channels under arbitrary input distributions.

1.2.2 Original Contributions

This thesis proposes a novel mathematical framework using tensors for modelling multi-domain communication systems. Through several examples of multi-user, multi-carrier, and multi-antenna systems, the utility of the proposed framework is established. In particular, this thesis characterizes the multi-linear channel in a multi-domain communication system as a higher order tensor and finds its Shannon capacity. The tensor approach allows to find the channel capacity under a family of power constraints such as per element or per domain constraints. It is shown that the capacity multiplexing gain associated with a tensor channel can increase exponentially with increase in the number of domains. The tensor technique also allows to characterize the capacity and rate regions for multi-user MIMO systems. It is shown that for multi-user systems, the tensor technique allows to treat inter-user interferences as information bearing entities, thereby providing larger possible sum rates as compared to the matrix techniques which often treat interference as noise. Also, a tensor framework for estimation has been proposed which includes the multi-linear estimator, and the widely multi-linear estimator. It is shown that when the signal or data to be estimated is inherently multi-domain in nature, then joint estimation of the signals across all the domains achieved through the tensor framework can lead to much better mean square error performance at the receiver as compared to the per domain estimators often used in a matrix/vector setting. The tensor framework gives a structured mathematical formalism to treat multi-domain interferences and thereby captures the mutual effect of various domain parameters. This thesis also generalizes the vector I-MMSE relation to a tensor setting, and uses it to find the tensor channel capacity when the input is constrained to be drawn from discrete signalling constellations with given input distribution. Once again, the tensor approach allows to solve this problem for not just the sum power constraint, but for a family of power constraints spanning multiple domains.

This thesis has resulted into the following published contributions :

Journals :

[J1] D. Pandey, and H. Leib, "The Tensor Multi-linear Channel and its Shannon Capacity", IEEE Access, pp 1-36, vol 10, March 2022.

[J2] D. Pandey, A. Venugopal, and H. Leib, "Multi-Domain Communication Systems and Networks: A Tensor-Based Approach", MDPI Network vol 1(2), pp 50-74, July 2021 (*Invited Paper*).

[J3] D. Pandey, and H. Leib, "A Tensor Framework for Multi-Linear Complex MMSE Estimation", IEEE Open Journal of Signal Processing, vol 2, pp 336-358, May 2021.

Conference Proceedings :

[C1] D. Pandey, and H. Leib, "Shannon Capacity of Tensor Channels under a Family of Power Constraints", in 2021 Biennial Symposium on Communications (BSC 2021), held between Jun 29, 2021 – Jun 30, 2021.

[C2] D. Pandey, and H. Leib, "A Tensor based Precoder and Receiver for MIMO GFDM systems", in ICC 2021-IEEE International Conference on Communications (pp. 1-6), held between Jun 14, 2021 – Jun 23, 2021.

[C3] D. Pandey, and H. Leib, “Tensor multi-linear MMSE estimation using the Einstein product”, in Future of Information and Communication Conference (FICC) held between April 29, 2021- April 30, 2021, (pp. 47-64), Springer, Cham.
(This paper won the **Best Student Paper award** at the conference)

Poster Presentations :

[P1] D. Pandey, and H. Leib, “The Tensor Channel for Multi-domain Communications and its Information Transmission Capacity”, STARaCom Annual Meeting, McGill University, Montreal, Feb 2019.

[P2] D. Pandey, and H. Leib, “Capacity Achieving Multi-linear Precoder for Tensor Channels with Arbitrary Input Distributions”, STARaCom Annual Meeting, McGill University, Montreal, May 2022.

Author Contributions Statement : The research for all the published/presented work listed here, except [J2], was carried out by the first author, D. Pandey under the guidance and supervision of H. Leib. The research published in [J2] was carried out equally by both the first and second authors, D.Pandey and A.Venugopal (fellow graduate student), under the supervision of H. Leib. The drafts for all the papers were written by D. Pandey, and reviewed by H. Leib.

The research presented in [J1], [J2], [C1], [C2], and [P1] is mostly based on Chapters 2 and 3 of this thesis. The contributions [J3] and [C3] are based on Chapters 2 and 4 of this thesis, while [P2] is based on some initial results from Chapter 5. A journal draft based on results from Chapter 5 is under preparation.

1.2.3 Outline of the Thesis

The organization of this thesis is as follows :

Chapter 2 introduces the notion of tensors along with some relevant tensor algebra results and properties. We present several tensor operations such as tensor SVD, EVD, inversion, Hermitian, based on the Einstein product definition. A new numerical approach for calculating tensor inversion based on Newton's Method is also presented. Complex random tensors and their second order characteristics are also defined. Furthermore, we use the tensor properties to develop and present our generic framework which models several multi-domain communication systems. Various examples of practical multi-domain communication systems where channel could be modelled as a higher order tensor based on our system model are also presented.

Chapter 3 introduces information theoretic notions such as Entropy, Mutual Information, associated with the tensor based multi-domain communication systems. In particular, we harness the tensor framework developed in Chapter 2 to mathematically define a family of input power constraints spanning multiple domains. We find the Shannon Capacity of higher order tensor channels under the family of power constraints. This chapter also presents an algorithmic approach to find the optimal input covariance, along with a discussion on its computational complexity. Furthermore, this chapter presents several numerical examples illustrating the capacity results for various tensor channel sizes, and different constraints. We also present application of our work to MIMO GFDM systems. Also, as an application, we consider the capacity of multi-user MIMO multiple access channels and interference channels through the proposed tensor framework.

Chapter 4 focusses on the MMSE estimation problem in the context of tensor signals. We present the best MMSE estimator followed by a widely multi-linear MMSE and multi-linear MMSE estimators. A comparison between the proposed estimator using the Einstein product and the Tucker product approach is also presented. We present several numerical examples with applications of the tensor framework for MMSE estimation of Gaussian signals, tensors stored in TT format and multi-domain communication systems. Simulation results for MIMO OFDM systems are presented to illustrate the performance of various tensor estimation techniques.

Chapter 5 employs the results from Chapters 3 and 4, to find the capacity of tensor channels when the input is drawn from discrete signalling constellations. A mathematical relation between the gradient of the mutual information presented in Chapter 3, and the MMSE error covariance tensor derived in Chapter 4 is presented which generalizes a similar relation from [107] to tensor settings. Furthermore, this relation is used to find the capacity achieving precoding scheme at the transmitter under discrete input constellation constraints.

Chapter 6 presents the conclusions from all the previous chapters, and shows direction for future works.

Appendices A, B present some supplementary mathematical results and derivations required for a more detailed understanding of several key results in the thesis. **Appendix C** is a guide for computer simulation codes required to reproduce the results presented in this thesis.

Chapter 2

Tensor Algebra and the System Model

This chapter presents relevant tensor definitions and properties needed for this thesis. More details on tensor algebra can be found in [32, 41, 84, 48, 70, 49]. Also, this chapter includes the tensor based system model used to represent multi-domain communication systems, along with examples of various practical systems modelled through the proposed framework.

2.1 Tensor Algebra

2.1.1 Preliminary Definitions

Tensors are multi-way arrays with components indexed by N indices also known as *modes*. The number of modes, N is called the *order* of the tensor. Tensors can be seen as a generalization of matrices which have only two modes (order 2 tensor) or vectors which have only one mode (order 1 tensor) [41]. A *fiber* is defined by fixing every index in a tensor except one, and can be considered as the higher order analogue of matrix rows and columns. Similarly, a *slice* is a two dimensional section of a tensor defined by fixing all but two indices [32].

Definition 1. Tensor Linear Space : The set of all tensors of size $I_1 \times \ldots \times I_K$ over $\mathbb{C}$ forms a linear space, denoted as $\mathbb{T}_{I_1,\ldots,I_K}(\mathbb{C})$. For $\mathcal{A}, \mathcal{B} \in \mathbb{T}_{I_1,\ldots,I_K}(\mathbb{C})$ and $\alpha \in \mathbb{C}$, the sum $\mathcal{A} + \mathcal{B} = \mathcal{C} \in \mathbb{T}_{I_1,\ldots,I_K}(\mathbb{C})$ where $\mathcal{C}_{i_1,\ldots,i_k} = \mathcal{A}_{i_1,\ldots,i_k} + \mathcal{B}_{i_1,\ldots,i_k}$, and scalar multiplication $\alpha \cdot \mathcal{A} = \mathcal{D} \in \mathbb{T}_{I_1,\ldots,I_K}(\mathbb{C})$ where $\mathcal{D}_{i_1,\ldots,i_k} = \alpha \mathcal{A}_{i_1,\ldots,i_k}$.

Definition 2. Matricization Transformation [48]: Let us denote the linear space of $P \times Q$ matrices over $\mathbb{C}$ as $\mathbb{M}_{P,Q}(\mathbb{C})$. For an order $K = N + M$ tensor $\mathcal{A} \in \mathbb{C}^{I_1 \times \ldots \times I_N \times J_1 \times \ldots \times J_M}$, the transformation $f_{I_1,\ldots,I_N|J_1,\ldots,J_M} : \mathbb{T}_{I_1,\ldots,I_N,J_1,\ldots,J_M}(\mathbb{C}) \Rightarrow \mathbb{M}_{I_1 \cdot I_2 \cdots I_{N-1} \cdot I_N, J_1 \cdot J_2 \cdots J_{M-1} \cdot J_M}(\mathbb{C})$ with $f_{I_1,\ldots,I_N|J_1,\ldots,J_M}(\mathcal{A}) = \mathrm{A}$ is defined component-wise as :

$$\mathcal{A}_{i_1,i_2,\ldots,i_N,j_1,j_2,\ldots,j_M} \underset{f^{-1}_{I_1,\ldots,I_N|J_1,\ldots,J_M}}{\overset{f_{I_1,\ldots,I_N|J_1,\ldots,J_M}}{\rightleftarrows}} \mathrm{A}_{i_1 + \sum_{k=2}^{N}(i_k - 1)\prod_{l=1}^{k-1} I_l,\, j_1 + \sum_{k=2}^{M}(j_k - 1)\prod_{l=1}^{k-1} J_l} \tag{2.1}$$

The transformation in (2.1) is also called matricization, or matrix unfolding of tensor by partitioning the indices into two disjoint subsets [47]. The vectorization operation as defined in [113] can be seen as a special case of (2.1) where $J_1 = \cdots = J_M = 1$. The bar in subscript of $f_{I_1,\ldots,I_N|J_1,\ldots,J_M}$ represents the partitioning after N modes of an $N + M$ order tensor where first N modes correspond to the rows of the representing matrix, and the last M modes correspond to the columns of the representing matrix. This mapping is bijective [70], and it preserves addition and scalar multiplication operations i.e., for $\mathcal{A}, \mathcal{B} \in \mathbb{T}_{I_1,\ldots,I_N,J_1,\ldots,J_M}(\mathbb{C})$ and any scalar $\alpha \in \mathbb{C}$, we have $f_{I_1,\ldots,I_N|J_1,\ldots,J_M}(\mathcal{A} + \mathcal{B}) = f_{I_1,\ldots,I_N|J_1,\ldots,J_M}(\mathcal{A}) + f_{I_1,\ldots,I_N|J_1,\ldots,J_M}(\mathcal{B})$ and $f_{I_1,\ldots,I_N|J_1,\ldots,J_M}(\alpha\mathcal{A}) = \alpha f_{I_1,\ldots,I_N|J_1,\ldots,J_M}(\mathcal{A})$. Hence the linear spaces $\mathbb{T}_{I_1,\ldots,I_N,J_1,\ldots,J_M}(\mathbb{C})$ and $\mathbb{M}_{I_1 \cdot I_2 \cdots I_{N-1} \cdot I_N, J_1 \cdot J_2 \cdots J_{M-1} \cdot J_M}(\mathbb{C})$ are isomorphic and the transformation $f_{I_1,\ldots,I_N|J_1,\ldots,J_M}$ is an isomorphism between the linear spaces. For a matrix, the transformation (2.1) creates a column vector when $N = 2, M = 0$, a row vector when $N = 0, M = 2$ does not change when $N = M = 1$.

Definition 3. Tensor Contracted product [47]: A contraction between tensors $\mathcal{X} \in \mathbb{C}^{I_1 \times \ldots \times I_M \times J_1 \times \ldots \times J_N}$ and $\mathcal{Y} \in \mathbb{C}^{I_1 \times \ldots \times I_M \times K_1 \times \ldots \times K_P}$ along their M common modes denoted by $\langle \mathcal{X}, \mathcal{Y} \rangle_{\{1,\ldots,M;1,\ldots,M\}}$ leads to a resulting tensor, $\mathcal{Z} \in \mathbb{C}^{J_1 \times \ldots \times J_N \times K_1 \times \ldots \times K_P}$ given by :

$$\mathcal{Z}_{j_1,\ldots,j_N,k_1,\ldots,k_P} = \sum_{i_1,\ldots,i_M} \mathcal{X}_{i_1,\ldots,i_M,j_1,\ldots,j_N} \mathcal{Y}_{i_1,\ldots,i_M,k_1,\ldots,k_P} \tag{2.2}$$

In general, the modes to be contracted need not be consecutive. However, the dimensions of the corresponding modes being contracted must be same. For instance, tensors $\mathcal{X} \in \mathbb{C}^{I\times J\times K\times L}$ and $\mathcal{Y} \in \mathbb{C}^{J\times P\times L\times Q}$ can be contracted as $\mathcal{Z} = \langle \mathcal{X}, \mathcal{Y}\rangle_{\{2,4;1,3\}}$ to generate $\mathcal{Z} \in \mathbb{C}^{I\times K\times P\times Q}$ with elements $\mathcal{Z}_{i,k,p,q} = \sum_{j,l} \mathcal{X}_{i,j,k,l}\mathcal{Y}_{j,p,l,q}$. Other tensor products, such as the Einstein product or the mode-n product of tensor with matrices can be seen as special cases of the tensor contracted product.

Definition 4. Einstein product [48]: For any N, the Einstein product is defined using the operation $*_N$ by :

$$(\mathcal{A} *_N \mathcal{B})_{i_1,\ldots,i_P,j_1,\ldots,j_M} = \sum_{k_1,\ldots,k_N} \mathcal{A}_{i_1,i_2,\ldots,i_P,k_1,\ldots,k_N}\mathcal{B}_{k_1,\ldots k_N,j_1,j_2,\ldots,j_M} \tag{2.3}$$

where $\mathcal{A} \in \mathbb{C}^{I_1\times\ldots\times I_P\times K_1\times\ldots\times K_N}$ and $\mathcal{B} \in \mathbb{C}^{K_1\times\ldots\times K_N\times J_1\times\ldots\times J_M}$.

Einstein product is a special case of tensor contracted product where contraction is over N consecutive modes. Both outer product and inner product can be seen as special cases of the Einstein product. For tensors $\mathcal{X}, \mathcal{Y} \in \mathbb{C}^{I_1\times I_2\times\ldots\times I_N}$ and $\mathcal{Z} \in \mathbb{C}^{J_1\times J_2\times\ldots\times J_M}$, we define the *Inner Product* as :

$$\langle \mathcal{X}, \mathcal{Y}\rangle = \sum_{i_1=1}^{I_1}\sum_{i_2=1}^{I_2}\cdots\sum_{i_N=1}^{I_N} \mathcal{X}_{i_1,i_2,\ldots,i_N}\mathcal{Y}^*_{i_1,i_2,\ldots,i_N} = \mathcal{X} *_N \mathcal{Y}^* \tag{2.4}$$

where $\langle \mathcal{X}, \mathcal{Y}\rangle$ is a scalar, and the *Outer Product* as :

$$(\mathcal{X} \circ \mathcal{Z})_{i_1,i_2,\ldots,i_N,j_1,j_2,\ldots,j_M} = \mathcal{X}_{i_1,i_2,\ldots,i_N}\mathcal{Z}_{j_1,j_2,\ldots,j_M} = \mathcal{X} *_0 \mathcal{Z} \tag{2.5}$$

where $(\mathcal{X} \circ \mathcal{Z}) \in \mathbb{C}^{I_1\times\ldots\times I_N\times J_1\times\ldots\times J_M}$. Also, the *norm* of $\mathcal{X}$ is defined as :

$$||\mathcal{X}|| = \sqrt{\sum_{i_1=1}^{I_1}\sum_{i_2=1}^{I_2}\ldots\ldots\sum_{i_N=1}^{I_N} \mid \mathcal{X}_{i_1,i_2,\ldots.,i_N}\mid^2} \tag{2.6}$$

Definition 5. mode-n product[65]: The mode-n product of a tensor $\mathcal{X} \in \mathbb{C}^{I_1\times I_2\times\ldots\times I_N}$ with a matrix $\mathrm{U} \in \mathbb{C}^{J\times I_n}$ is denoted by $\mathcal{X}\times_n\mathrm{U}$ and is defined as :

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

where $(\mathcal{X} \times_n \mathrm{U}) \in \mathbb{C}^{I_1\times\ldots\times I_{n-1}\times J\times I_{n+1}\times\ldots\times I_N}$.

Definition 6. Square tensors [70]: A tensor $\mathcal{A} \in \mathbb{C}^{I_1 \times \ldots \times I_N \times J_1 \times \ldots \times J_M}$ is called a square tensor if $N = M$ and $I_k = J_k$ for $k = 1, \ldots, N$.

For square tensors $\mathcal{A}, \mathcal{B}$ of size $I \times J \times I \times J$, it was first shown in [48] that $f_{I,J|I,J}(\mathcal{A} *_2 \mathcal{B}) = f_{I,J|I,J}(\mathcal{A}) \cdot f_{I,J|I,J}(\mathcal{B})$ where $\cdot$ refers to the usual matrix multiplication. It can be easily generalized to square or non-square tensors of any order and size as shown in [70, 114].

Lemma 1. *For tensors $\mathcal{A} \in \mathbb{C}^{I_1 \times \ldots \times I_N \times J_1 \times \ldots \times J_M}$ and $\mathcal{B} \in \mathbb{C}^{J_1 \times \ldots \times J_M \times K_1 \times \ldots \times K_P}$ under the transformation from (2.1), the following holds:*

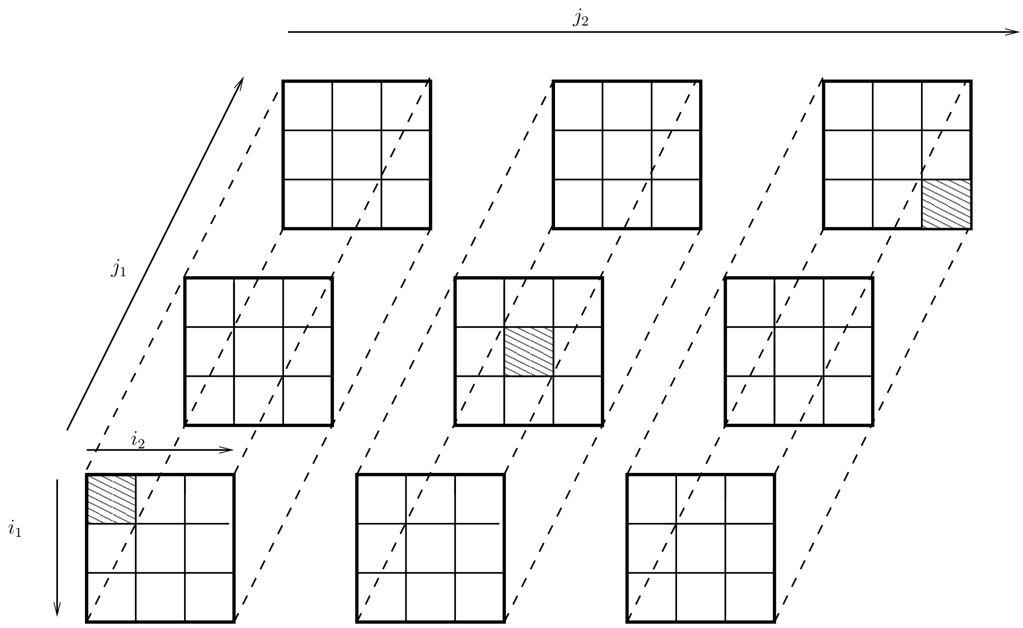
$$f_{I_1,\ldots,I_N|K_1,\ldots,K_P}(\mathcal{A} *_M \mathcal{B}) = f_{I_1,\ldots,I_N|J_1,\ldots,J_M}(\mathcal{A}) \cdot f_{J_1,\ldots,J_M|K_1,\ldots,K_P}(\mathcal{B}) \tag{2.8}$$

The proof for Lemma 1 is provided in [114].

Definition 7. Pseudo-diagonal Tensors : Any tensor $\mathcal{D} \in \mathbb{C}^{I_1 \times \ldots \times I_N \times J_1 \times \ldots \times J_M}$ of order $N + M$ is called pseudo-diagonal if its transformation $f_{I_1,\ldots,I_N|J_1,\ldots,J_M}(\mathcal{D})$ yields a diagonal matrix.

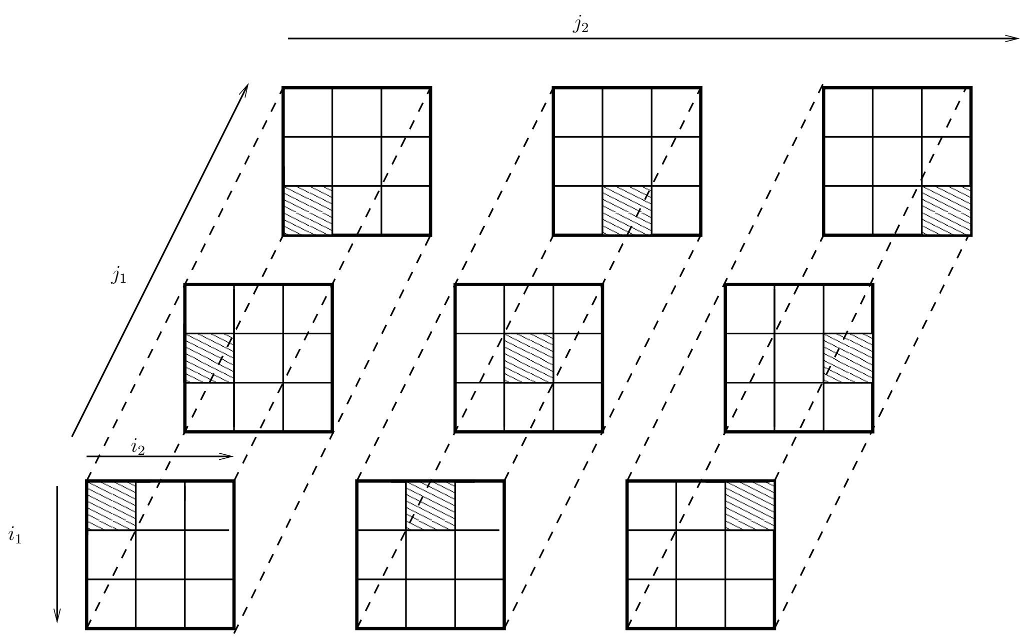
A square tensor $\mathcal{D} \in \mathbb{C}^{I_1 \times \ldots \times I_N \times I_1 \times \ldots \times I_N}$ is pseudo-diagonal if all its entries $\mathcal{D}_{i_1,\ldots,i_N,j_1,\ldots,j_N}$ are zero except when $i_1 = j_1, i_2 = j_2, \ldots, i_N = j_N$. Such a tensor under transformation $f_{I_1,\ldots,I_N|I_1,\ldots,I_N}(\mathcal{D})$ always yields a square diagonal matrix. In [48, 68] such a tensor is termed as diagonal tensor, and in [45] it is termed as U-diagonal. However, we will refer to it as pseudo-diagonal in this thesis, so as to make a distinction from the diagonal tensor definition more widely found in literature which states that a diagonal tensor has all entries $\mathcal{D}_{i_1,\ldots,i_N}$ zero except when $i_1 = i_2 = \cdots = i_N$ [32]. This can be seen as a strict diagonal condition as non-zero elements exist only when all the modes have same index. In a pseudo-diagonal tensor, say of order $2N$, elements are non-zero when every ith and $(i + N)$th mode have same index for $i = 1, \ldots, N$. An example of order 4 tensor showing the difference between diagonal and pseudo-diagonal structures is presented in Figure 2.1 where the empty squares represent zero elements.

For a matrix, which has just two modes, the diagonal and pseudo-diagonal structures are the same. Note that pseudo-diagonality is defined with respect to a partition after N modes. For instance, if we refer to a third order tensor as pseudo-diagonal, it is important



Strictly diagonal tensor $\mathcal{A}$ of order 4, size $3 \times 3 \times 3 \times 3$

Non-zero elements occur only at $\mathcal{A}_{1,1,1,1}, \mathcal{A}_{2,2,2,2}, \mathcal{A}_{3,3,3,3}$, i.e. when $i_i = i_2 = j_1 = j_2$



Pseudo-diagonal tensor $\mathcal{A}$ of order 4, size $3 \times 3 \times 3 \times 3$, with respect to partition after 2 modes

Non-zero elements occur at $\mathcal{A}_{1,1,1,1}, \mathcal{A}_{1,2,1,2}, \mathcal{A}_{2,1,2,1}, \mathcal{A}_{2,2,2,2}, \mathcal{A}_{1,3,1,3}, \mathcal{A}_{3,1,3,1}, \mathcal{A}_{3,2,3,2}, \mathcal{A}_{2,3,2,3}, \mathcal{A}_{3,3,3,3}$ i.e. when $i_1 = j_1, i_2 = j_2$

Fig. 2.1: Order 4 diagonal and pseudo-diagonal tensors.

to specify whether it is pseudo-diagonal with respect to partition after the first mode or the second mode. Hence to avoid overload of notation in this thesis, whenever we refer to a tensor of order $N+M$ or order $2N$ as pseudo-diagonal, then the tensor is pseudo-diagonal with respect to a partition after N modes.

Definition 8. An *identity tensor*, $\mathcal{I} \in \mathbb{C}^{I_1\times\ldots\times I_N\times I_1\times\ldots\times I_N}$ is a square pseudo-diagonal tensor such that for any other square tensor $\mathcal{A} \in \mathbb{C}^{I_1\times\ldots\times I_N\times I_1\times\ldots\times I_N}$, we have $\mathcal{A}*_N\mathcal{I} = \mathcal{I}*_N\mathcal{A} = \mathcal{A}$. For non-square tensor $\mathcal{B} \in \mathbb{C}^{J_1\times\ldots\times J_M\times I_1\ldots\times\ldots\times I_N}$, we have $\mathcal{B} *_N \mathcal{I}_N = \mathcal{B}$ and $\mathcal{I}_M *_M \mathcal{B} = \mathcal{B}$ where $\mathcal{I}_N$ and $\mathcal{I}_M$ are identity tensors of order $2N$ and $2M$ respectively.

Definition 9. The tensor $\mathcal{A}^{-1} \in \mathbb{C}^{I_1\times\ldots\times I_N\times I_1\times\ldots\times I_N}$ is an *inverse* of a square tensor of same size, $\mathcal{A} \in \mathbb{C}^{I_1\times\ldots\times I_N\times I_1\times\ldots\times I_N}$ if $\mathcal{A} *_N \mathcal{A}^{-1} = \mathcal{A}^{-1} *_N \mathcal{A} = \mathcal{I}_N$ [70].

Definition 10. The *Hermitian* of a tensor $\mathcal{A} \in \mathbb{C}^{I_1\times\ldots\times I_N\times J_1\times\ldots\times J_M}$ is a tensor $\mathcal{B} \in \mathbb{C}^{J_1\times\ldots\times J_M\times I_1\times\ldots\times I_N}$ which has entries $\mathcal{B}^*_{j_1,j_2,\ldots,j_M,i_1,i_2,\ldots,i_N} = \mathcal{A}_{i_1,i_2,\ldots,i_N,j_1,j_2,\ldots,j_M}$ and is denoted as $\mathcal{A}^H$, where $()^*$ denotes complex conjugate. Note that since tensors have more than two modes, so there can be multiple ways to define a tensor Hermitian or transpose using permutations [115]. For simplicity of notation, in this thesis wherever we write a tensor explicitly as $N+M$ or $2N$ order tensor, then $\mathcal{A}^H$ is always with respect to partition after N modes as explained in this definition. Also, a square tensor $\mathcal{X} \in \mathbb{C}^{I_1\times\ldots\times I_N\times I_1\times\ldots\times I_N}$ is called *Hermitian tensor* if $\mathcal{X} = \mathcal{X}^H$.

Definition 11. A square tensor, $\mathcal{U} \in \mathbb{C}^{I_1\times\ldots\times I_N\times I_1\times\ldots\times I_N}$ is called *unitary* if $\mathcal{U}^H *_N \mathcal{U} = \mathcal{U} *_N \mathcal{U}^H = \mathcal{I}$.

Some important properties of the Einstein product

Based on the definitions of tensor inverse, Hermitian, the Einstein product, and Lemma 1, several tensor algebra relations and properties can be derived. Here we list a few properties which are often used and can be proven by expanding the tensor operations element-wise or using Lemma 1.

1. *Associativity* : For tensors $\mathcal{A} \in \mathbb{C}^{I_1\times\ldots\times I_P\times J_1\times\ldots\times J_N}$, $\mathcal{B} \in \mathbb{C}^{J_1\times\ldots\times J_N\times K_1\times\ldots\times K_M}$ and $\mathcal{C} \in \mathbb{C}^{K_1\times\ldots\times K_M\times T_1\times\ldots\times T_Q}$, we have
$$(\mathcal{A} *_N \mathcal{B}) *_M \mathcal{C} = \mathcal{A} *_N (\mathcal{B} *_M \mathcal{C}) \tag{2.9}$$

2. *Commutativity* : Einstein product is not commutative in general. However for the specific case where the product is taken over all the N modes of one of the tensors, say for tensors $\mathcal{A} \in \mathbb{C}^{I_1\times\ldots\times I_P\times J_1\times\ldots\times J_N}$ and $\mathcal{B} \in \mathbb{C}^{J_1\times\ldots\times J_N}$, the following can be established :
$$\mathcal{A} *_N \mathcal{B} = \mathcal{B} *_N \mathcal{A}^T \tag{2.10}$$

3. *Distributivity* : For tensors, $\mathcal{A}, \mathcal{B} \in \mathbb{C}^{I_1\times\ldots\times I_P\times J_1\times\ldots\times J_N}$ and $\mathcal{C} \in \mathbb{C}^{J_1\times\ldots\times J_N\times K_1\times\ldots\times K_M}$, we have :
$$(\mathcal{A} + \mathcal{B}) *_N \mathcal{C} = (\mathcal{A} *_N \mathcal{C}) + (\mathcal{B} *_N \mathcal{C}) \tag{2.11}$$

4. For tensors $\mathcal{A} \in \mathbb{C}^{I_1\times\ldots\times I_M\times J_1\times\ldots\times J_N}$ and $\mathcal{B} \in \mathbb{C}^{J_1\times\ldots\times J_N\times K_1\times\ldots\times K_P}$, we have :
$$(\mathcal{A} *_N \mathcal{B})^H = \mathcal{B}^H *_N \mathcal{A}^H \tag{2.12}$$

5. For square tensors $\mathcal{A}$ and $\mathcal{B} \in \mathbb{C}^{I_1\times\ldots\times I_N\times I_1\times\ldots\times I_N}$, we have :
$$(\mathcal{A} *_N \mathcal{B})^{-1} = \mathcal{B}^{-1} *_N \mathcal{A}^{-1} \tag{2.13}$$

2.1.2 Tensor SVD and EVD

Tucker decomposition of a tensor can be seen as a higher order SVD [65] and has found many applications particularly in extracting low rank structures in higher dimensional data [116]. A more specific version of tensor SVD is presented in [48] as a tool for finding tensor inversion and solving multi-linear systems. Note that [48] presents SVD for square tensors only. The idea of SVD from [48] is further generalized for any even order tensor in [68]. However, it can be further extended for any arbitrary order and size of tensor in the form of the following theorem.

Theorem 1. *For a tensor,* $\mathcal{A} \in \mathbb{C}^{I_1\times\ldots\times I_N\times J_1\times\ldots\times J_M}$*, the SVD of* $\mathcal{A}$ *has the form :*
$$\mathcal{A} = \mathcal{U} *_N \mathcal{D} *_M \mathcal{V}^H \tag{2.14}$$

where $\mathcal{U} \in \mathbb{C}^{I_1 \times \ldots \times I_N \times I_1 \times \ldots \times I_N}$ *and* $\mathcal{V} \in \mathbb{C}^{J_1 \times \ldots \times J_M \times J_1 \times \ldots \times J_M}$ *are unitary tensors and* $\mathcal{D} \in \mathbb{C}^{I_1 \times \ldots \times I_N \times J_1 \times \ldots \times J_M}$ *is a pseudo-diagonal tensor whose non-zero values are the singular values of* $\mathcal{A}$*.*

Note that for a tensor of order $N + M$, we will get different valid SVDs for different values of N and M. For a given N and M, the existence of a tensor SVD can be shown using Lemma 1 [48]. A proof of this theorem for $2N$ order tensors with $N = M$ using transformation defined in (2.1) is provided in [48]. The proof of the general case is included in Appendix B.1.

Several different definitions of tensor eigenvalues exist in the literature [66] which intend to generalize the properties of matrix eigenvalues to higher order. Motivated by their applications in physics and mechanics, several of such definitions apply to the case of only super-symmetric tensors which are defined as a class of tensors that are invariant under any permutation of their indices [67]. However, there is no single generalization of eigenvalues to tensor case that preserves all the properties of the matrix eigenvalues [63]. For our purposes, we present a particular generalization from [69] which can be easily extended to any square tensor, irrespective of symmetry in its elements.

Definition 12. Let $\mathcal{A} \in \mathbb{C}^{I_1 \times \ldots \times I_N \times I_1 \times \ldots \times I_N}$, $\mathcal{X} \in \mathbb{C}^{I_1 \times \ldots \times I_N}$, $\lambda \in \mathbb{C}$, where $\mathcal{X}$ and λ satisfy $\mathcal{A} *_N \mathcal{X} = \lambda \mathcal{X}$, then we call $\mathcal{X}$ and λ as *eigentensor* and *eigenvalue* of $\mathcal{A}$ respectively [69].

Lemma 2. *Eigenvalues* λ *of a Hermitian tensor* $\mathcal{A} \in \mathbb{C}^{I_1 \times \ldots \times I_N \times I_1 \times \ldots \times I_N}$ *are real, i.e* $\lambda \in \mathbb{R}$.

Proof. Consider $\mathcal{X} \in \mathbb{C}^{I_1 \times \ldots \times I_N}$ and $\lambda \in \mathbb{C}$. Since $\mathcal{A}$ is Hermitian, we have $\mathcal{A}^H = \mathcal{A}$

$$\mathcal{A} *_N \mathcal{X} = \lambda \mathcal{X}$$
$$\mathcal{X} *_N \mathcal{A}^T = \lambda \mathcal{X} \quad \text{(from (2.10))}$$
$$(\mathcal{X} *_N \mathcal{A}^T)^* = (\lambda \mathcal{X})^*$$
$$\mathcal{X}^* *_N \mathcal{A}^H = \lambda^* \mathcal{X}^*$$
$$\mathcal{X}^* *_N \mathcal{A} = \lambda^* \mathcal{X}^* \quad \text{(since } \mathcal{A} = \mathcal{A}^H\text{)}$$
$$\mathcal{X}^* *_N \mathcal{A} *_N \mathcal{X} = \lambda^* \mathcal{X}^* *_N \mathcal{X}$$
$$\mathcal{X}^* *_N (\lambda \mathcal{X}) = \lambda^* \mathcal{X}^* *_N \mathcal{X} \quad \text{(as } \mathcal{A} *_N \mathcal{X} = \lambda \mathcal{X}\text{)}$$
$$\lambda \mathcal{X}^* *_N \mathcal{X} = \lambda^* \mathcal{X}^* *_N \mathcal{X} \quad \text{(as } \lambda \text{ is a scalar)}$$
$$\lambda = \lambda^* \Rightarrow \lambda \in \mathbb{R} \quad \text{(since } \mathcal{X} \text{ is non-zero)}$$

□

Theorem 2. *The EVD of a Hermitian tensor* $\mathcal{A} \in \mathbb{C}^{I_1 \times \ldots \times I_N \times I_1 \times \ldots \times I_N}$ *is given as* [48] *:*

$$\mathcal{A} = \mathcal{U} *_N \mathcal{D} *_N \mathcal{U}^H \tag{2.15}$$

where $\mathcal{U} \in \mathbb{C}^{I_1 \times \ldots \times I_N \times I_1 \times \ldots \times I_N}$ *is a unitary tensor and* $\mathcal{D} \in \mathbb{C}^{I_1 \times \ldots \times I_N \times I_1 \times \ldots \times I_N}$ *is a square pseudo-diagonal tensor, i.e.* $\mathcal{D}_{i_1,\ldots,i_N,j_1,\ldots,j_N} = 0$ *if* $(i_1, \ldots, i_N) \neq (j_1, \ldots, j_N)$ *with its non-zero values being the eigenvalues of* $\mathcal{A}$ *and* $\mathcal{U}$ *containing the eigentensors of* $\mathcal{A}$*.*

This theorem can be proven using Lemma 1, details are provided in [48, 117]. The eigenvalues of $\mathcal{A}$ are same as the eigenvalues of $f_{I_1,\ldots,I_N|I_1,\ldots,I_N}(\mathcal{A})$ [117].

Definition 13. Positive semi-definite and definite tensors: A square tensor $\mathcal{A} \in \mathbb{C}^{I_1 \times \ldots \times I_N \times I_1 \times \ldots \times I_N}$ is *positive semi-definite*, denoted by $\mathcal{A} \succeq 0$ if all its eigenvalues are non-negative, which is the same as $f_{I_1,\ldots,I_N|I_1,\ldots,I_N}(\mathcal{A})$ being a positive semi-definite matrix. A tensor is *positive definite*, $\mathcal{A} \succ 0$, if all its eigenvalues are strictly greater than zero.

A positive semi-definite pseudo-diagonal tensor $\mathcal{D}$, will have all its components non-negative and its square root can be denoted as $\mathcal{D}^{1/2}$ which is also pseudo-diagonal positive semi-definite whose elements are the square root of the elements of $\mathcal{D}$ such that $\mathcal{D}^{1/2} *_N \mathcal{D}^{1/2} = \mathcal{D}$. Similarly, if $\mathcal{D}$ is positive definite, its inverse can be denoted as $\mathcal{D}^{-1}$ which is also pseudo-diagonal whose non zero elements are the reciprocal of the corresponding elements of $\mathcal{D}$. Based on the tensor EVD, we can also write the square root of any Hermitian positive semi-definite tensor as $\mathcal{A}^{1/2} = \mathcal{U} *_N \mathcal{D}^{1/2} *_N \mathcal{U}^H$ and inverse of any Hermitian positive definite tensor as $\mathcal{A}^{-1} = \mathcal{U} *_N \mathcal{D}^{-1} *_N \mathcal{U}^H$.

Definition 14. Trace: The trace of a tensor $\mathcal{A} \in \mathbb{C}^{I_1 \times \ldots \times I_N \times I_1 \times \ldots \times I_N}$ is defined as the sum of its pseudo-diagonal entries :

$$\operatorname{tr}(\mathcal{A}) = \sum_{i_1,\ldots,i_N} \mathcal{A}_{i_1,i_2,\ldots,i_N,i_1,i_2,\ldots,i_N} \tag{2.16}$$

Definition 15. Determinant: The determinant of a tensor $\mathcal{A} \in \mathbb{C}^{I_1 \times \ldots \times I_N \times I_1 \times \ldots \times I_N}$ is defined as the product of its eigenvalues i.e., if $\mathcal{A} = \mathcal{U} *_N \mathcal{D} *_N \mathcal{U}^H$, then

$$\det(\mathcal{A}) = \prod_{i_1,\ldots,i_N} \mathcal{D}_{i_1,i_2,\ldots,i_N,i_1,i_2,\ldots,i_N} \tag{2.17}$$

The eigenvalues of $\mathcal{A}$ are the same as that of its matrix transformation, hence $\det(\mathcal{A}) = \det(f_{I_1,\ldots,I_N|I_1,\ldots,I_N}(\mathcal{A}))$. Note that a few other definitions for determinant exist in literature based on how one chooses to define the eigenvalues of tensors [118]. The definition we have presented is the one corresponding to the eigenvalue definition used in this thesis. This definition is the same as the unfolding determinant in [70].

Some properties of trace and determinant

The following properties can be shown by writing the tensors component wise or using Lemma 1.

1. For two tensors $\mathcal{A} \in \mathbb{C}^{I_1 \times \ldots \times I_N}$ and $\mathcal{B} \in \mathbb{C}^{I_1 \times \ldots \times I_N}$ of same size and order N,

$$\mathcal{A} *_N \mathcal{B} = \mathcal{B} *_N \mathcal{A} = \operatorname{tr}(\mathcal{A} \circ \mathcal{B}) = \operatorname{tr}(\mathcal{B} \circ \mathcal{A}) \tag{2.18}$$

   Note that (2.18) can be seen as a specific case of (2.10) by setting $P = 0$. In (2.10), if $P = 0$ then $\mathcal{A}^T = \mathcal{A}$ and thus (2.10) reduces to $\mathcal{A} *_N \mathcal{B} = \mathcal{B} *_N \mathcal{A}$. Also, $\mathcal{A} *_N \mathcal{B}$ is a scalar in this case which is given by $\sum_{i_1,\ldots,i_N} \mathcal{A}_{i_1,\ldots,i_N} \mathcal{B}_{i_1,\ldots,i_N}$. Hence the second part of the equality in (2.18) follows from the definition of outer product and trace which gives $\operatorname{tr}(\mathcal{A} \circ \mathcal{B}) = \operatorname{tr}(\mathcal{B} \circ \mathcal{A}) = \sum_{i_1,\ldots,i_N} \mathcal{A}_{i_1,\ldots,i_N} \mathcal{B}_{i_1,\ldots,i_N}$.

2. For tensors $\mathcal{A} \in \mathbb{C}^{I_1\times\ldots\times I_N\times J_1\times\ldots\times J_M}$ and $\mathcal{B} \in \mathbb{C}^{J_1\times\ldots\times J_M\times I_1\times\ldots\times I_N}$, we have :
$$\operatorname{tr}(\mathcal{A} *_M \mathcal{B}) = \operatorname{tr}(\mathcal{B} *_N \mathcal{A}) \tag{2.19}$$
$$\det(\mathcal{I}_N + \mathcal{A} *_M \mathcal{B}) = \det(\mathcal{I}_M + \mathcal{B} *_N \mathcal{A}) \tag{2.20}$$
where $\mathcal{I}_N$ and $\mathcal{I}_M$ are identity tensors of order $2N$ and $2M$ respectively. To prove (2.20), we can use Lemma 1 and Sylvester's matrix determinant identity [119].

3. For tensors $\mathcal{A}, \mathcal{B} \in \mathbb{C}^{I_1\times\ldots\times I_N\times I_1\times\ldots\times I_N}$, we have [70]:
$$\det(\mathcal{A} *_N \mathcal{B}) = \det(\mathcal{B} *_N \mathcal{A}) = \det(\mathcal{A}) \cdot \det(\mathcal{B}) \tag{2.21}$$

4. Trace of a positive semi-definite tensor is the sum of its eigenvalues.

5. The determinant of a unitary tensor is 1 and the determinant of a square pseudo-diagonal tensor is the product of its pseudo-diagonal entries.

2.1.3 Tensor Train Decomposition

The TT decomposition represents a higher order tensor through a set of sparsely connected lower order tensors called *cores*. For a tensor $\mathcal{T}$ of size $I_1 \times I_2 \times \ldots \times I_N$, the TT decomposition is written as [62] :
$$\mathcal{T}_{i_1,\ldots,i_N} = \sum_{r_0,r_1,\ldots,r_N} \mathcal{T}^{(1)}_{r_0,i_1,r_1} \cdot \mathcal{T}^{(2)}_{r_1,i_2,r_2} \cdots \mathcal{T}^{(N)}_{r_{N-1},i_N,r_N} \tag{2.22}$$
where each $\mathcal{T}^{(i)}$ is a third order tensor of size $R_{i-1} \times I_i \times R_i$ and R_i denote the TT ranks with $R_0 = R_N = 1$ and $R_i \geq 1$ for $i = 1, \ldots, N-1$ [62]. In TT decomposition, the low rank structure of the core tensors is harnessed for reducing the storage complexity. Instead of storing the whole tensor, only the cores $\mathcal{T}^{(i)}$ are stored. Different cores are also sometimes stored in distributed storage systems. This imposes a requirement that any mathematical operation to be performed on the tensor should avoid a full reconstruction or rearranging of the data but should be able to act on the cores themselves. The mechanism with which tensor operations act on the cores can be best understood using a graphical representation of the TT decomposition through a Tensor Network (TN) containing nodes and edges.

A TN is a graphical illustration of tensor operations where each node represents a tensor and each outgoing edge from a node represents a mode of the tensor. Hence a vector can be represented in a TN using a node with a single edge, a matrix with a node and two edges, and a tensor of order N with a node and N edges. An edge connecting two nodes in a TN represents contraction between the two tensors along the connected edge. The free edges correspond to the modes which are not contracted. The total number of free edges represent the order of the resulting tensor. Subsequently, the TT decomposition from (2.22) can be represented using a TN as shown in Figure 2.2 where each core is represented by a node.

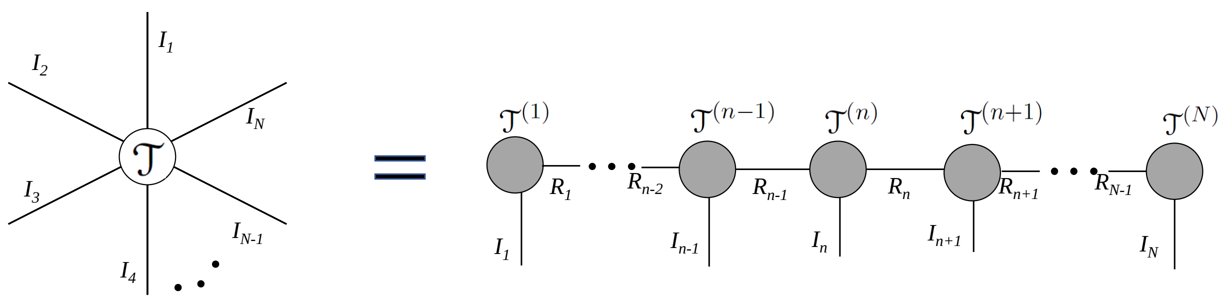


Fig. 2.2: TN representation of Nth order tensor $\mathcal{T} \in \mathbb{C}^{I_1 \times I_2 \times \ldots \times I_N}$ in TT format.

The TT ranks of a tensor determine the storage consumption of the tensor. In several cases, the exact TT decomposition of a tensor may lead to high TT ranks. Hence often an approximation of the TT decomposition of a tensor is computed with a given accuracy ϵ to fit a desired set of low TT ranks for reduced storage [62]. If we denote the TT approximation of a tensor $\mathcal{A}$ as $\mathcal{A}_T$, the computed approximation satisfies $||\mathcal{A} - \mathcal{A}_T|| \leq \epsilon ||\mathcal{A}||$. A sequential SVD based algorithm to compute such TT decomposition with given accuracy ϵ is presented in [62].

The Einstein product between two different tensors, $\mathcal{Q} *_K \mathcal{P}$ stored in TT format can be represented using the TN as shown in Figure 2.3. The gray nodes represent the cores in the tensor train of $\mathcal{P} \in \mathbb{C}^{L_1 \times \ldots \times L_K \times J_1 \times \ldots \times J_M}$ and the white nodes represent the cores in the tensor train of $\mathcal{Q} \in \mathbb{C}^{I_1 \times \ldots \times I_N \times L_1 \times \ldots \times L_K}$. An advantage of computing the Einstein product using the TT format is that the resulting tensor can be directly obtained in terms of its TT cores. An algorithm to compute the Einstein product in the TT format without reconstructing the whole tensor has been provided in [120].

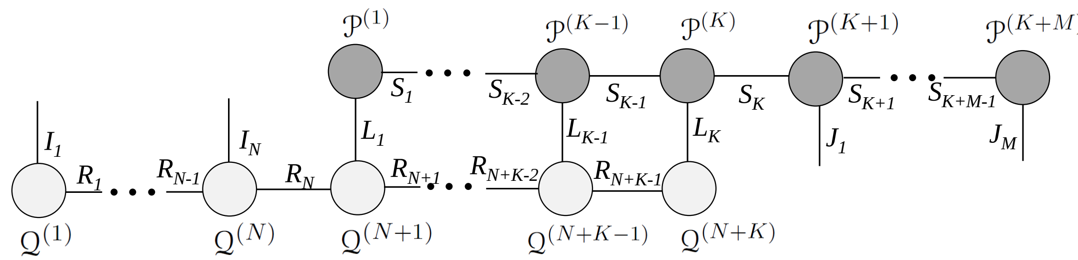


Fig. 2.3: TN representation of Einstein Product $\mathcal{Q} *_K \mathcal{P}$ for tensors $\mathcal{Q} \in \mathbb{C}^{I_1 \times \ldots \times I_N \times L_1 \times \ldots \times L_K}$ and $\mathcal{P} \in \mathbb{C}^{L_1 \times \ldots \times L_K \times J_1 \times \ldots \times J_M}$ in TT format where R_i and S_j represent TT ranks for $\mathcal{Q}$ and $\mathcal{P}$ respectively.

2.1.4 Complex Random Tensors

A tensor is said to be *random* if its components are random variables. Expectation of a random tensor $\boldsymbol{\mathcal{X}} \in \mathbb{C}^{I_1 \times \ldots \times I_N}$, denoted by $\bar{\mathcal{M}} = \mathbb{E}[\boldsymbol{\mathcal{X}}] \in \mathbb{C}^{I_1 \times \ldots \times I_N}$, is a tensor with each component consisting of the expected value of the corresponding component of $\boldsymbol{\mathcal{X}}$.

Covariance and Pseudo-covariance

The covariance of a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{C}^{I_1 \times I_2 \times \ldots \times I_N}$ can be defined as a tensor of size $I_1 \times I_2 \times \ldots \times I_N \times I_1 \times I_2 \times \ldots \times I_N$ represented by $\mathcal{Q} = \mathbb{E}[(\boldsymbol{\mathcal{X}} - \bar{\mathcal{M}}) \circ (\boldsymbol{\mathcal{X}} - \bar{\mathcal{M}})^*]$ where $\bar{\mathcal{M}} = \mathbb{E}[\boldsymbol{\mathcal{X}}]$ is the mean tensor. However, a complete second-order characterization also requires defining the pseudo-covariance which is given as $\tilde{\mathcal{Q}} = \mathbb{E}[(\boldsymbol{\mathcal{X}} - \bar{\mathcal{M}}) \circ (\boldsymbol{\mathcal{X}} - \bar{\mathcal{M}})]$. A complex random tensor is said to be *proper* if its pseudo-covariance is $0_{\mathcal{T}}$, i.e $\tilde{\mathcal{Q}} = 0_{\mathcal{T}}$, where $0_{\mathcal{T}}$ denotes an all zero tensor. Corresponding definitions for vector and matrix cases can be found in [121, 122]. For vectors, an augmented representation is used to completely define the second order characteristics where the augmented vector is written as $\ddot{\underline{\mathbf{x}}} = \begin{bmatrix} \underline{\mathbf{x}} \\ \underline{\mathbf{x}}^* \end{bmatrix}$ [123, 103]. Next, we will exploit the multi-domain nature of tensors to develop an augmented representation of complex tensors.

Unlike a vector, a tensor has more than one mode. Therefore concatenation of the conjugate tensor can be done across any mode. However, given that the primary advantage of a tensor is its ability to maintain the distinction between different domains, we suggest that for a complex valued tensor $\boldsymbol{\mathcal{X}} \in \mathbb{C}^{I_1 \times \ldots \times I_N}$, the augmented tensor can be created by adding another domain of size 2 such that $\ddot{\boldsymbol{\mathcal{X}}} \in \mathbb{C}^{I_1 \times \ldots \times I_N \times 2}$ where $\ddot{\boldsymbol{\mathcal{X}}}_{i_1,\ldots,i_N,1} = \boldsymbol{\mathcal{X}}_{i_1,\ldots,i_N}$ and $\ddot{\boldsymbol{\mathcal{X}}}_{i_1,\ldots,i_N,2} = \boldsymbol{\mathcal{X}}^*_{i_1,\ldots,i_N}$. The augmented mean tensor will be $\ddot{\mathcal{M}} = \mathbb{E}[\ddot{\boldsymbol{\mathcal{X}}}]$. The augmented covariance tensor will be given as $\ddot{\mathcal{Q}} = \mathbb{E}[(\ddot{\boldsymbol{\mathcal{X}}} - \ddot{\mathcal{M}}) \circ (\ddot{\boldsymbol{\mathcal{X}}} - \ddot{\mathcal{M}})^*]$ of size $I_1 \times \ldots \times I_N \times 2 \times I_1 \times \ldots \times I_N \times 2$.

The augmented covariance tensor $\ddot{\mathcal{Q}}$ contains the covariance $\mathcal{Q}$ and the pseudo-covariance tensor $\tilde{\mathcal{Q}}$ along with their conjugates as $\ddot{\mathcal{Q}}[\underbrace{:,\ldots,:}_{N},1,\underbrace{:,\ldots,:}_{N},1] = \mathcal{Q}$, $\ddot{\mathcal{Q}}[:,\ldots,:,1,:,\ldots,:,2] = \tilde{\mathcal{Q}}$, $\ddot{\mathcal{Q}}[:,\ldots,:,2,:,\ldots,:,1] = \tilde{\mathcal{Q}}^*$ and $\ddot{\mathcal{Q}}[:,\ldots,:,2,:,\ldots,:,2] = \mathcal{Q}^*$.

Tensor Gaussian Distribution

Definition 16. The pdf of a general Gaussian distributed complex-valued tensor $\boldsymbol{\mathcal{X}} \in \mathbb{C}^{I_1 \times I_2 \times \ldots \times I_N}$ of order N is given by :

$$p_{\boldsymbol{\mathcal{X}}}(\underline{\mathrm{x}}) = \frac{1}{(\pi)^{I_1 I_2 \ldots I_N} (\det(\ddot{\mathrm{Q}}))^{1/2}} \times \exp\Big\{ -\frac{1}{2}(\underline{\ddot{\mathrm{x}}} - \underline{\ddot{\mathrm{m}}})^H \ddot{\mathrm{Q}}^{-1} (\underline{\ddot{\mathrm{x}}} - \underline{\ddot{\mathrm{m}}}) \Big\} \tag{2.23}$$

where $\underline{\ddot{\mathrm{x}}} = \mathrm{vec}(\ddot{\mathcal{X}})$, $\underline{\ddot{\mathrm{m}}} = \mathrm{vec}(\mathbb{E}[\ddot{\boldsymbol{\mathcal{X}}}])$ and $\ddot{\mathrm{Q}}$ is the covariance matrix of the vectorized augmented tensor where $f_{I_1,\ldots,I_N,2|I_1,\ldots,I_N,2}(\ddot{\mathcal{Q}}) = \ddot{\mathrm{Q}}$. We can re-write the pdf as :

$$p_{\boldsymbol{\mathcal{X}}}(\mathcal{X}) = \frac{1}{(\pi)^{I_1 I_2 \ldots I_N} (\det(\ddot{\mathcal{Q}}))^{1/2}} \times \exp\Big\{ -\frac{1}{2}(\ddot{\mathcal{X}} - \ddot{\mathcal{M}})^* *_{N+1} \ddot{\mathcal{Q}}^{-1} *_{N+1} (\ddot{\mathcal{X}} - \ddot{\mathcal{M}}) \Big\} \tag{2.24}$$

Notationally, we write $\boldsymbol{\mathcal{X}} \sim \mathcal{CN}(\bar{\mathcal{M}}, \mathcal{Q}, \tilde{\mathcal{Q}})$. The equivalence of (2.23) and (2.24) can be directly established based on the properties of Einstein product and Lemma 1. For proper complex Gaussian tensor, with zero pseudo-covariance the pdf simplifies as :

$$p_{\boldsymbol{\mathcal{X}}}(\mathcal{X}) = \frac{1}{(\pi)^{I_1 I_2 \ldots I_N} \det(\mathcal{Q})} \times \exp\Big\{ -(\mathcal{X} - \bar{\mathcal{M}})^* *_N \mathcal{Q}^{-1} *_N (\mathcal{X} - \bar{\mathcal{M}}) \Big\} \tag{2.25}$$

where $\bar{\mathcal{M}} = \mathbb{E}[\boldsymbol{\mathcal{X}}]$ is the order-N mean tensor and $\mathcal{Q} = \mathbb{E}[(\boldsymbol{\mathcal{X}} - \bar{\mathcal{M}}) \circ (\boldsymbol{\mathcal{X}} - \bar{\mathcal{M}})^*]$ is the order-$2N$ covariance tensor.

2.1.5 Tensor Gradients and Integrals

Any real valued scalar function $g(x)$ of a complex variable x can be seen as a function of its real and imaginary components or equivalently as a function of the complex variable and its conjugate, i.e. $g(x) = f(x, x^*)$. Thus to find a stationary point of $f(x, x^*)$, it is shown in Appendix A.7.4 of [124] that one can use the partial derivative with respect to either the complex variable, $\frac{\partial f}{\partial x} = 0$ or its conjugate $\frac{\partial f}{\partial x^*} = 0$. Either one of the two conditions would be necessary and sufficient to determine a stationary point. Furthermore, since $g(x)$ is real-valued, stationary point can simply be found by taking the derivative of the function $g(x)$ with respect to x^* and setting it to zero [125].

The gradient with respect to a matrix is defined as $\nabla_{\mathrm{X}} f \triangleq \partial f / \partial \mathrm{X}^*$, where $[\nabla_{\mathrm{X}} f]_{i,j} = \partial f / \partial \mathrm{X}^*_{i,j}$ [107]. We similarly define the gradient with respect to a tensor as $\nabla_{\mathcal{X}} f \triangleq \partial f / \partial \mathcal{X}^*$ where the gradient is a tensor of the same size as $\mathcal{X}$ whose individual components are the derivatives with respect to the components of $\mathcal{X}^*$, i.e. $[\nabla_{\mathcal{X}} f]_{i_1,\dots,i_N} = \partial f / \partial \mathcal{X}^*_{i_1,\dots,i_N}$ where the complex derivative is defined as [107] :

$$\frac{\partial f}{\partial \mathcal{X}^*_{i_1,\dots,i_N}} \triangleq \frac{1}{2}\Big(\frac{\partial f}{\partial \Re\{\mathcal{X}_{i_1\dots,i_N}\}} + j\frac{\partial f}{\partial \Im\{\mathcal{X}_{i_1,\dots,i_N}\}}\Big). \tag{2.26}$$

Using this definition, we can extend several results on matrix complex gradients from [107], [126], [124, Appendix A.7] to a tensor setting. For instance, given $\mathcal{A} \in \mathbb{C}^{J_1\times\dots\times J_M\times I_1\times\dots\times I_N}$ and Hermitian positive semi-definite tensors $\mathcal{B}, \mathcal{C} \in \mathbb{C}^{I_1\times\dots\times I_N\times I_1\times\dots\times I_N}$, using results from [107], [126] and Lemma 1, we can write :

$$\nabla_{\mathcal{C}} \log[\det(\mathcal{I}_M + \mathcal{A} *_N \mathcal{C} *_N \mathcal{A}^H)] = \mathcal{A}^H *_M (\mathcal{A} *_N \mathcal{C} *_N \mathcal{A}^H + \mathcal{I}_M)^{-1} *_M \mathcal{A}, \tag{2.27}$$

$$\nabla_{\mathcal{C}} \operatorname{tr}(\mathcal{B} *_N \mathcal{C}) = \mathcal{B}. \tag{2.28}$$

Further, we represent the integration of a scalar function $f(\mathcal{A})$ over the tensor variable $\mathcal{A} \in \mathbb{C}^{I_1\times\dots\times I_N}$ as follows :

$$\int f(\mathcal{A})\partial\mathcal{A} = \int\cdots\int f(\mathcal{A})\partial\mathcal{A}_{1,\dots,1,1}\partial\mathcal{A}_{1,\dots1,2}\partial\mathcal{A}_{1,\dots1,3}\cdots\partial\mathcal{A}_{I_1,\dots,I_N}. \tag{2.29}$$

The result is a scalar entity obtained by a multi variable integration of the scalar function over all the components of the tensor $\mathcal{A}$. Now let us consider a tensor valued function g of the tensor $\mathcal{A}$ such that $g(\mathcal{A}) \in \mathbb{C}^{J_1\times\dots\times J_M}$. In such a case, integrating the tensor function over the tensor variable denoted as $\int g(\mathcal{A})\mathtt{d}\mathcal{A}$ would be a tensor of same size as $g(\mathcal{A})$ with each element given as :

$$\left[\int g(\mathcal{A})\mathtt{d}\mathcal{A}\right]_{j_1,\ldots,j_M} = \int \left[g(\mathcal{A})\right]_{j_1,\ldots,j_M} \partial\mathcal{A}. \tag{2.30}$$

Thus in this thesis, we use the notation of $\int()\partial\mathcal{A}$ when expressing an integration of a scalar function of a tensor $\mathcal{A}$ resulting into a scalar, and $\int()\mathtt{d}\mathcal{A}$ when expressing an integration of a tensor function of tensor $\mathcal{A}$ which results into a tensor. For instance, for a random variable $\boldsymbol{\mathcal{X}} \in \mathbb{C}^{I_1\times\ldots\times I_N}$ with pdf given by $p_{\boldsymbol{\mathcal{X}}}(\mathcal{X})$, the expected value (which is a tensor) would be written as :

$$\mathbb{E}[\boldsymbol{\mathcal{X}}] = \int \mathcal{X} p_{\boldsymbol{\mathcal{X}}}(\mathcal{X})\mathtt{d}\mathcal{X} \tag{2.31}$$

and to denote the expected value of a scalar function of the random variable, say $h(\boldsymbol{\mathcal{X}})$, we would write :

$$\mathbb{E}[h(\boldsymbol{\mathcal{X}})] = \int p_{\boldsymbol{\mathcal{X}}}(\mathcal{X})h(\mathcal{X})\partial\mathcal{X}. \tag{2.32}$$

2.1.6 Newton's Method for Tensor Inversion

Solving tensor equations without resorting to matrix transformation has become an active area of research in the past few years [48, 70, 68, 114, 127, 128, 129, 130, 131, 132, 133]. The reasons for avoiding a matrix unfolding of the tensors are many folds. In several applications, tensors arise naturally as part of the problem. Hence there is no point in unfolding the tensors for computations, only to revert back to the original structure afterwards, since it adds additional steps going to and from tensor space to matrix space. Also, keeping the tensor structure intact can allow us to leverage the information in the structure for reducing the complexity of operations and storage. In [62], it is shown that tensor train decompositions can be used to store large tensors by exploiting the redundancy in its elements. This is a primary reason why tensors are being so widely used in Big Data applications where even large vector and matrix data are being tensorized for storage [57]. Hence there is a need to express any processing for tensor signals within the tensor framework without changing the structure of the tensor.

Among other tensor operations, one of the more frequently used operations is tensor inversion [48]. Higher Order Bi-conjugate Gradient (HOBG) and Jacobi methods are presented in [48, 46] for tensor inversion. In this thesis, we present another numerical method for approximating tensor inversion, without using any matrix transformation. We extend the Newton's Method (NM) used for matrix inversion from [134] to tensors. To find the inverse of tensor $\mathcal{A} \in \mathbb{C}^{I_1\times\ldots\times I_N\times I_1\times\ldots\times I_N}$, we use the following iterative equation :

$$\mathcal{B}^{(k+1)} = (2\mathcal{I}_N - \mathcal{B}^{(k)} *_N \mathcal{A}) *_N \mathcal{B}^{(k)} \tag{2.33}$$

where the initial guess for the inverse $\mathcal{B}^{(0)}$ can be set to $a \cdot \mathcal{A}^H$. Using similar line of arguments as in Theorem 2 from [135], it can be shown that this method converges if $0 < a < 2/\sigma^2_{max}$ where σ^2_{max} is the largest eigenvalue of $\mathcal{C} = \mathcal{A}^H *_N \mathcal{A}$. To avoid the calculation of σ^2_{max}, we use the bound suggested for matrix inversion in [136]. We define :

$$\lambda = m + s(I_1 \cdot I_2 \cdots I_N - 1)^{1/2} \tag{2.34}$$

where $m = \text{tr}(\mathcal{C})/(I_1 \cdots I_N)$ and $s^2 = \text{tr}(\mathcal{C} *_N \mathcal{C}^H)/(I_1 \cdots I_N) - m^2$, and use $a = 2/\lambda$ to ensure convergence. The constant λ is an upper bound on the tensor eigenvalue, i.e. $\lambda \geq \sigma^2_{max}$. This upper bound for the matrix eigenvalues was derived in [137] and for the tensor eigenvalues has been derived in Appendix B.6.

2.1.7 Complexity of Newton's Method

We analyze the computational complexities in terms of the required number of flops for a given step. A flop is defined as a single floating point operation such as an addition, multiplication, subtraction, division, comparison ($>, <, ==$), or a scalar square root, etc.

Consider the definition of Einstein product across N modes between tensor $\mathcal{A}$ of size $I_1 \times \ldots \times I_P \times K_1 \times \ldots \times K_N$ and tensor $\mathcal{B}$ of size $K_1 \times \ldots \times K_N \times J_1 \times \ldots \times J_M$ from (2.3). Computing a single element in $\mathcal{A} *_N \mathcal{B}$ requires $K_1 \cdot K_2 \cdots K_N$ multiplications and also $K_1 \cdot K_2 \cdots K_N - 1$ additions. There are total $I_1 \cdot I_2 \cdots I_P \cdot J_1 \cdot J_2 \cdots J_M$ elements in the tensor $\mathcal{A} *_N \mathcal{B}$. Let $I = \prod_{i=1}^{P} I_i$, $J = \prod_{j=1}^{M} J_j$, and $K = \prod_{k=1}^{N} K_k$. There are a total IJ elements in $\mathcal{A} *_N \mathcal{B}$ where each element is computed using K multiplications and $K - 1$ additions. Subsequently finding all the elements of $\mathcal{A} *_N \mathcal{B}$ requires IJK multiplications and $(K-1)IJ$ additions. Hence a total of $IJK+(K-1)IJ = (2K-1)IJ$ flops are required for computing all the elements in $\mathcal{A} *_N \mathcal{B}$. If the number of required flops is polynomial in its size n, i.e. the operation requires $a_0 n^p + a_1 n^{p-1} + \cdots + a_p$ flops (for fixed constants $a_0, a_1, \ldots, a_p$), we represent the complexity only using the highest power in n as $\mathcal{O}(n^p)$. Hence the complexity of the Einstein product which requires $(2K-1)IJ$ flops is given as $\mathcal{O}(IKJ)$ which is same as $\mathcal{O}((I_1 \cdots I_P) \cdot (K_1 \cdots K_N) \cdot (J_1 \cdots J_M))$. For the case where both $\mathcal{A}$ and $\mathcal{B}$ are order $2N$ tensors of size $I_1 \times \ldots \times I_N \times I_1 \times \ldots \times I_N$ each, the complexity of $\mathcal{A} *_N \mathcal{B}$ is given as $\mathcal{O}((I_1 \cdots I_N)^3)$.

The complexity of each iteration in the NM for finding tensor inversion mainly depends on the complexity of the Einstein product between tensors of order $2N$. For (2.33), assuming $I_1 = I_2 = \cdots = I_N = L$, the worst case complexity of each iteration in NM is cubic in tensor size, $\mathcal{O}(L^{3N})$ and grows exponentially with the order of the tensor. However, using the approach presented in Appendix B.7, the per iteration complexity of tensor inversion using NM can be lowered to $\mathcal{O}(L^{2N})$. Further, use of parallel processing can reduce the time complexity of such tensor operations significantly. Using parallel processing, the time complexity of each iteration in NM is $\mathcal{O}(\log L^N)$ as shown in Appendix B.7, which is linear in the number of domains and not exponential.

The number of iterations required by NM is significantly lower than other methods such as HOBG and Jacobi methods for large tensors. A comparison of the number of required iterations by HOBG and Jacobi method is presented in [48] which shows that HOBG outperforms Jacobi method. Here, we present a comparison between HOBG and NM by comparing the performance of both the algorithms to find the inverse of tensors, having zero mean unit variance circular complex Gaussian entries of various size. The number of iterations required to find the inverse were averaged over 100 different realizations of tensors of each size. Tolerance was kept as 10^{-6}. Figure 2.4 shows number of iterations against order of the tensor when each dimension has size 3. Clearly, as the order increases, iterations required by NM are significantly lower than HOBG. Similarly, Figure 2.5 shows the number of iterations against dimensions of the individual domains for order 4 tensor, where similar observation can be made.

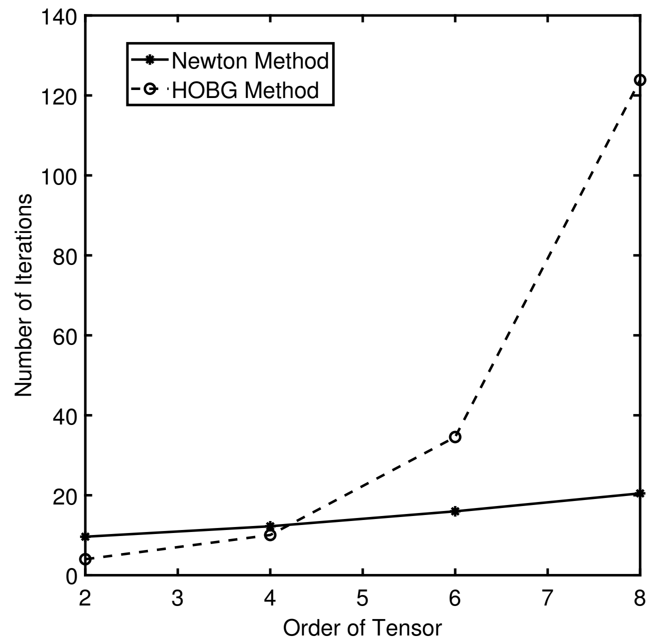


Fig. 2.4: Number of iterations vs Order of tensor with dimension 3 for Tensor Inversion Algorithms.

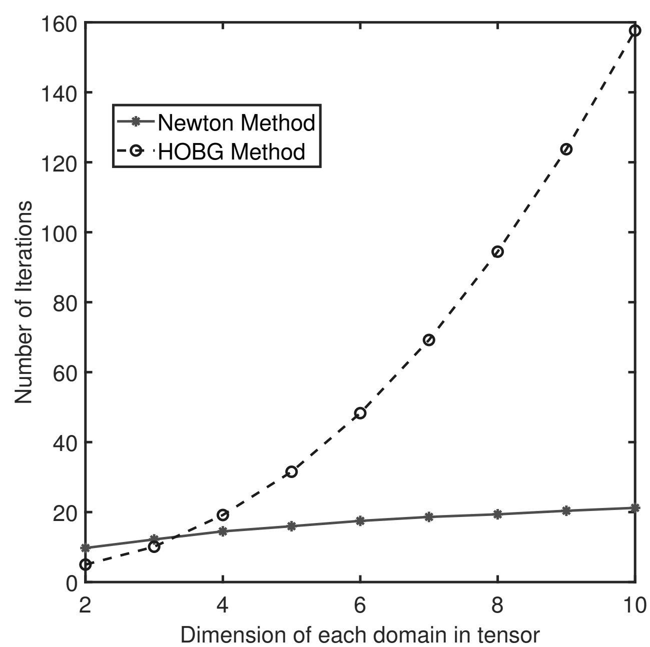


Fig. 2.5: Number of iterations vs Dimension of order 4 tensor for Tensor Inversion Algorithms.

Note that the main thrust of this thesis is not on the computational capabilities of tensors, but on the information theoretic aspects of the channels associated with a tensor based communication system. Thus, all the numerical/simulation results presented in this thesis were generated using MATLAB, which provides a convenient structure to define and operate on multi-way arrays. While MATLAB has several built-in functions to operate directly on higher order arrays, and also allows parallel execution of algorithms, it should be acknowledged that MATLAB may not be the most efficient programming platform for implementation of tensor based operations. There are faster programming platforms which can significantly exploit the tensor structures for more efficient execution of algorithms. Certain Python libraries such as Tensorflow and PyTorch, contain several built-in functionalities which can be used for implementation of various tensor operations, such as tensor products, reshape, concatenation, decompositions, etc. TensorFlow was created by Google as a machine learning implementation library and is widely used for various business and research purposes involving artificial intelligence, machine learning and deep learning [138]. Similarly, PyTorch was developed by Facebook as an open source machine learning and deep learning library [139]. One major advantage that PyTorch provides is its seamless integration with parallel computing platforms such as CUDA that allows using Graphics Processing Unit (GPU) for tensor operations. Several operations, even for large vectors and matrices are often performed by tensorizing the data for parallel and faster implementation on such platforms.

2.2 Tensor System Model for Multi-domain Communication Systems

The input and output in a multi-domain communication system can be defined as tensor symbols of order N and M respectively. Let $\mathcal{X} \in \mathbb{C}^{I_1 \times \ldots \times I_N}$ denote the input (transmitted) tensor symbol with $I_1, I_2, \ldots, I_N$ as the dimensions of its N domains where each component $\mathcal{X}_{i_1,\ldots,i_N}$ is a discrete complex symbol. Similarly, we represent the output (received) tensor symbol by $\mathcal{Y} \in \mathbb{C}^{J_1 \times \ldots \times J_M}$ with $J_1, J_2, \ldots, J_M$ as the dimensions of its M domains. With these input and output, a multi-linear channel between the transmit and the receive side can be defined as a tensor of order $M+N$ represented by $\mathcal{H} \in \mathbb{C}^{J_1 \times \ldots \times J_M \times I_1 \times \ldots \times I_N}$. The system model can be represented using the Einstein product of the channel tensor with the input tensor where we contract along all the N modes of the input. In the presence of a noise tensor, the system model is given as :

$$\boldsymbol{\mathcal{Y}} = \mathcal{H} *_N \boldsymbol{\mathcal{X}} + \boldsymbol{\mathcal{N}} \tag{2.35}$$

where

$$\boldsymbol{\mathcal{Y}}_{j_1,\ldots,j_M} = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} \mathcal{H}_{j_1,\ldots,j_M,i_1,\ldots,i_N} \boldsymbol{\mathcal{X}}_{i_1,\ldots,i_N} + \boldsymbol{\mathcal{N}}_{j_1,\ldots,j_M} \tag{2.36}$$

with $\boldsymbol{\mathcal{N}}$ representing the received noise tensor of same size as $\boldsymbol{\mathcal{Y}}$. The narrowband discrete time MIMO matrix channel model is a special case of the tensor model from (2.35) where input and output are order 1 tensors (vectors), the channel is an order 2 tensor (matrix) and the Einstein product reduces to regular matrix multiplication, $\underline{\mathbf{y}} = \mathrm{H}\underline{\mathbf{x}} + \underline{\mathbf{n}}$. In a matrix representation of a discrete MIMO channel which characterizes only the space domain, each component of the channel matrix $\mathrm{H}_{i,j}$ represents the complex gain (i.e. amounting for both phase change and amplitude gain) of different paths between transmit and receive antennas. In a tensor channel, each component from the channel is a complex gain that couples a component from the order N tensor input symbol to a component of the order M tensor output symbol. The proposed system model is generic and the number of modes along with the physical interpretation of the individual modes is system specific. The modes can represent space, time, frequency, propagation delay, users, spreading sequence, etc., depending on the system.

System models for three different cases (in the absence of noise) are illustrated in Figure 2.6. The first case represents the conventional MIMO system model where the input and output are order 1 tensors (vectors) of size 2, and the channel is an order 2 tensor (matrix) of size 2×2. Such a model can evolve further to incorporate additional domains as illustrated in the second and third cases. In the second case, the input and output are order 2 tensors of size 2×2 each, and the channel is an order 4 tensor of size $2 \times 2 \times 2 \times 2$. Further, in the third case the input and output are order 3 tensors of size $2 \times 2 \times 2$, while the channel is an order 6 tensor of size $2 \times 2 \times 2 \times 2 \times 2 \times 2$. The input indices are denoted using i_1, i_2, i_3 and the output indices using j_1, j_2, j_3. The multi-domain nature of the tensor channel and its coupling with the input through the Einstein product in the system model allow us to visualize (2.35) as an evolution of the MIMO matrix channel model, with latter being just a special case of the former.

For a MIMO communication system having N_R receive and N_T transmit antennas, the continuous time input-output relation in a linear time varying channel is written as [140] :

$$\underline{\mathbf{y}}(t) = \int \mathrm{H}(t,\tau)\underline{\mathbf{x}}(t-\tau)d\tau + \underline{\mathbf{n}}(t) \tag{2.37}$$

where $\underline{\mathbf{x}}(t)$ is the $N_T \times 1$ continuous time input vector, $\underline{\mathbf{n}}(t)$ and $\underline{\mathbf{y}}(t)$ are the $N_R \times 1$ noise and received vectors respectively. The channel matrix $\mathrm{H}(t,\tau)$ of size $N_R \times N_T$, has components $\mathrm{H}_{n_r,n_t}(t,\tau)$ that denote the channel impulse response between the transmit antenna n_t and the receive antenna n_r at a time instant t and delay τ. If the channel is assumed time-invariant with a maximum delay τ_{max}, the discretization of (2.37) at a sampling frequency f_s gives the input/output relation at an instant k as [140]:

$$\underline{\mathbf{y}}[k] = \sum_{d=1}^{D} \mathrm{H}[d]\underline{\mathbf{x}}[k-(d-1)] + \underline{\mathbf{n}}[k], \qquad k = 0, 1, \ldots, (N-1) \tag{2.38}$$

where $\underline{\mathbf{x}}[k]$ is the $N_T \times 1$ channel input at time index k, $\underline{\mathbf{n}}[k]$ and $\underline{\mathbf{y}}[k]$ are the $N_R \times 1$ noise and received vectors respectively. The $N_R \times N_T$ matrix $\mathrm{H}[d]$ has components $\mathrm{H}_{n_r,n_t}[d]$ which represents the length D channel impulse response between transmit antenna n_t and receive antenna n_r at delay d, where $D = \lceil f_s \tau_{max} \rceil$. Delay can be considered as another domain in the system model. So for a time-invariant channel, at any time instant k the relation in (2.38) can be expressed using tensor model as [141]:

$$\underline{\mathbf{y}}[k] = \mathcal{H} *_2 \mathbf{X}[k] + \underline{\mathbf{n}}[k] \tag{2.39}$$

where all the individual vectors $\underline{\mathbf{x}}[k], \underline{\mathbf{x}}[k-1], \ldots, \underline{\mathbf{x}}[k-(D-1)]$ from (2.38) form the columns of the matrix $\mathbf{X}[k]$ of size $N_T \times D$ and all the individual matrices $\mathrm{H}[1], \ldots, \mathrm{H}[D]$ from (2.38) are stacked together where they form the slices of the order-three channel tensor $\mathcal{H}$ of size $N_R \times N_T \times D$. For $D = 1$, $\mathcal{H}$ reduces to a matrix and $\mathbf{X}[k]$ to a vector in (2.39). As D increases, the tensor framework allows capturing the time dispersion in the system

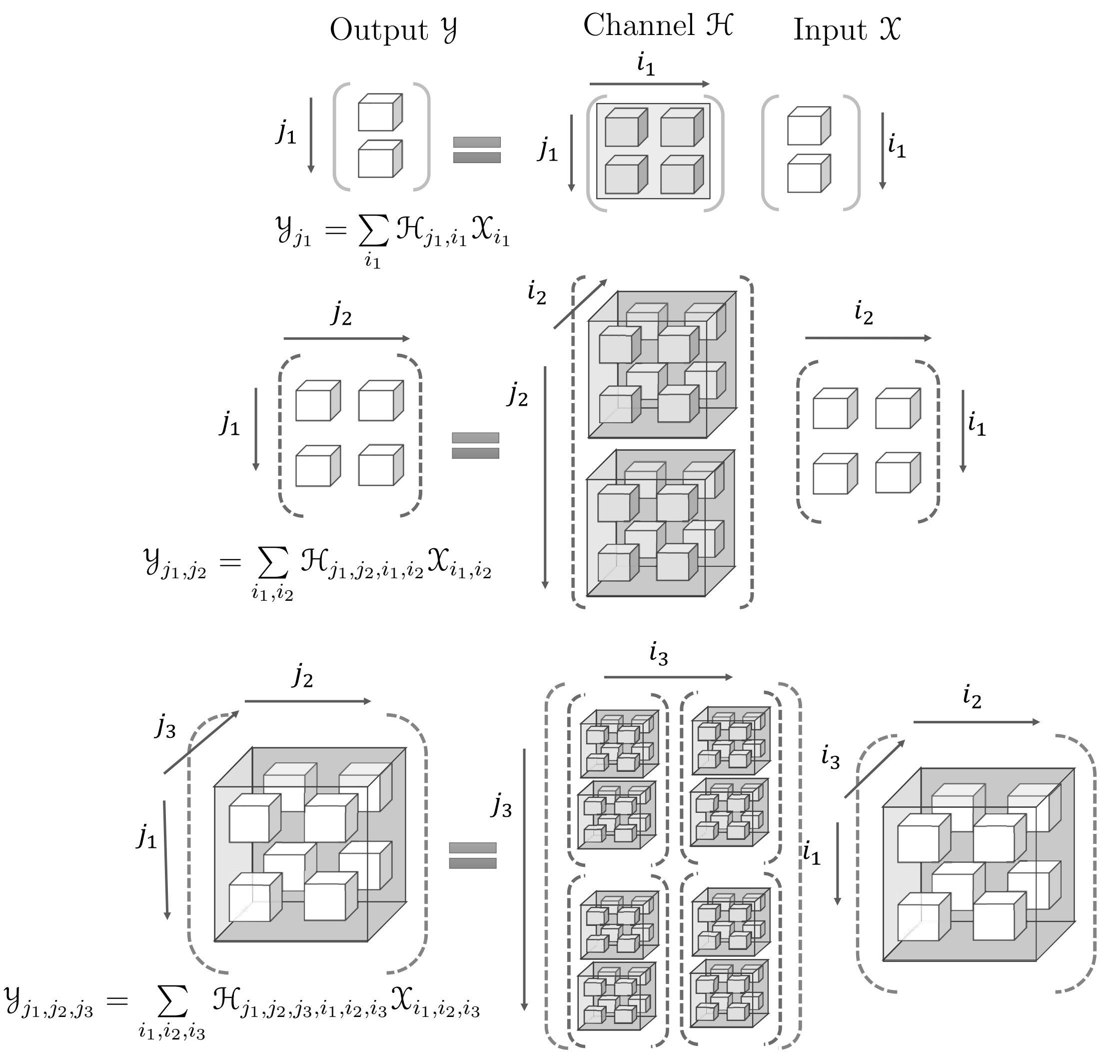


Fig. 2.6: The tensor system model and its evolution with the increase in order.

model by increasing the dimension of the delay domain in the channel and the input. Now assuming that the channel is time variant, the discretized input/output relation can be given as [24]:

$$\underline{\mathbf{y}}[k] = \sum_{d=1}^{D} \mathrm{H}[k,d]\underline{\mathbf{x}}[k-(d-1)] + \underline{\mathbf{n}}[k], \quad k = 0, 1, \ldots, (N-1) \tag{2.40}$$

where each element $\mathrm{H}_{n_r,n_t}[k,d]$ represents complex channel gain between n_tth transmit and n_rth receive antenna for delay d at time instant k. In [24], assuming a cyclic prefix addition to each input block, (2.40) is expressed in matrix notation as $\underline{\mathbf{y}}' = \mathrm{H}'\underline{\mathbf{x}}' + \underline{\mathbf{n}}'$ over a time block of N symbol durations by appending vectors $\underline{\mathbf{y}}[k]$, $\underline{\mathbf{x}}[k-(d-1)]$ and $\underline{\mathbf{n}}[k]$ for different k into vectors $\underline{\mathbf{y}}'$, $\underline{\mathbf{x}}'$ and $\underline{\mathbf{n}}'$ of size $N \cdot N_R$, $N \cdot N_T$ and $N \cdot N_R$ respectively, and the channel matrix $\mathrm{H}[k,d]$ into a larger matrix H' of size $N \cdot N_R \times N \cdot N_T$. However, appending the vectors implies making the two distinct domains indistinguishable in the system formulation. Hence a more obvious and intuitive way to represent such a system would be using tensors where the channel can be expressed as $N_R \times N \times N_T \times D'$ tensor where $D' = N + D - 1$. We do not assume any cyclic prefix addition here. Since output at index k i.e. $\underline{\mathbf{y}}[k]$ will depend on inputs $\underline{\mathbf{x}}[k], \underline{\mathbf{x}}[k-1], \ldots, \underline{\mathbf{x}}[k-(D-1)]$, so corresponding to output being indexed by N time indices $0, 1, \ldots, (N-1)$, the input will be indexed by $N + D - 1$ time indices $-(D-1), \ldots, (N-1)$. So in the system model (2.39), we add a domain of length N time slots to the channel tensor and the output tensor to account for the time variation and increase the delay domain length to $N + D - 1$ in the channel tensor and the input tensor. Thereby, our system model becomes :

$$\mathbf{Y} = \mathcal{H} *_2 \mathbf{X} + \mathbf{N} \tag{2.41}$$

where all the individual vectors $\underline{\mathbf{y}}[k]$ from (2.40) for $k = 0, \ldots, (N-1)$ form the columns of the matrix $\mathbf{Y}$ of size $N_R \times N$. Similarly vectors $\underline{\mathbf{n}}[k]$ from (2.40) form the columns of the matrix $\mathbf{N}$ of size $N_R \times N$ and vectors $\underline{\mathbf{x}}[d']$ where $d' = k - (d-1)$ form the columns of the matrix $\mathbf{X}$ of size $N_T \times D'$. All the individual matrices $\mathrm{H}[k,d]$ from (2.40) are now sub-tensors of the order-four channel tensor $\mathcal{H}$ of size $N_R \times N \times N_T \times D'$ where $\mathcal{H}_{:,k,:,d} = \mathrm{H}[k,d]$. We can see how the tensor system model in (2.39) simply evolved in (2.41) to account for time variation of the channel as well.

2.3 Tensor Model applied to practical systems

The model presented in (2.35) can be used to represent a wide variety of systems, not necessarily confined to communication systems. For instance, the system model used in [69] for image restoration model is a specific case of the system model in (2.35). To emphasize the relevance of the proposed tensor channel model, particularly in multi-domain communication systems, next we present a few examples of practical systems which can be modelled using tensors.

2.3.1 MIMO OFDM and Multi-user MIMO OFDM

OFDM is one of the most popular multi-carrier schemes and has been used extensively with MIMO in 4G standards and Wi-Fi [13]. A conventional model for a MIMO OFDM system in the frequency domain is given by [24] :

$$\check{\underline{\mathbf{y}}}[p] = \check{\mathrm{H}}[p,p]\check{\underline{\mathbf{x}}}[p] + \sum_{q=0,q\neq p}^{N_{sc}-1} \check{\mathrm{H}}[p,q]\check{\underline{\mathbf{x}}}[q] + \check{\underline{\mathbf{n}}}[p] \tag{2.42}$$

where $\check{\underline{\mathbf{y}}}[p]$, $\check{\underline{\mathbf{x}}}[p]$ and $\check{\underline{\mathbf{n}}}[p]$ are the frequency domain received, transmitted and noise symbol vectors at sub-carrier p, while N_{sc} denotes the number of sub-carriers. The model represented by (2.42) can be obtained from (2.40) by taking the $N_{sc}-$point DFT of $\{\underline{\mathbf{y}}[k]\}$ where $DFT\{\underline{\mathbf{y}}[k]\} = \check{\underline{\mathbf{y}}}[p]$, $DFT\{\underline{\mathbf{n}}[k]\} = \check{\underline{\mathbf{n}}}[p]$ and $DFT\{\underline{\mathbf{x}}[k]\} = \check{\underline{\mathbf{x}}}[p]$. The frequency domain $N_R \times N_T$ channel matrix between transmit sub-carrier q and receive sub-carrier p is $\check{\mathrm{H}}[p,q]$, whose individual elements $\check{\mathrm{H}}_{n_r,n_t}[p,q]$ can be obtained from the discrete time varying channel between the n_rth receive antenna and n_tth transmit antenna, $\mathrm{H}_{n_r,n_t}[k,d]$ (based on the DFT of (2.40)) as :

$$\check{\mathrm{H}}_{n_r,n_t}[p,q] = \frac{1}{N_{sc}} \sum_{k=0}^{N_{sc}-1} \sum_{d=0}^{D-1} \mathrm{H}_{n_r,n_t}[k,d] e^{j2\pi k(q-p)/N_{sc}} e^{-j2\pi qd/N_{sc}} \tag{2.43}$$

where $1 \leq n_r \leq N_R$, $1 \leq n_t \leq N_T$ and $0 \leq p,q \leq N_{sc}-1$. Using tensors, we can represent the frequency domain input/output relation in MIMO OFDM of (2.42) as :

$$\check{\mathbf{Y}} = \check{\mathcal{H}} *_2 \check{\mathbf{X}} + \check{\mathbf{N}} \tag{2.44}$$

where each vector $\underline{\check{\mathbf{y}}}[p]$ and $\underline{\check{\mathbf{n}}}[p]$ from (2.42) for $p = 0, \ldots, N_{sc} - 1$ form the columns of matrices $\check{\mathbf{Y}}$ and $\check{\mathbf{N}}$ of size $N_R \times N_{sc}$, and vectors $\underline{\check{\mathbf{x}}}[p]$ form the columns of matrix $\check{\mathbf{X}} \in \mathbb{C}^{N_T \times N_{sc}}$. The input and output are connected by an order 4 tensor channel $\check{\mathcal{H}} \in \mathbb{C}^{N_R \times N_{sc} \times N_T \times N_{sc}}$ where each element $\check{\mathrm{H}}_{n_r,n_t}[p,q]$ from (2.43) is now an element in the channel tensor as $\check{\mathcal{H}}_{n_r,p,n_t,q}$. We can expand the MIMO OFDM system model to include users as an additional domain in the model which will lead to a sixth order tensor channel. In the case of Multi-User (MU) MIMO OFDM, the frequency domain channel matrix is often represented as an $N_R \times N_T$ matrix corresponding to a specific user and a specific sub-carrier [142, 143]. To account for Inter-Carrier Interference (ICI) as well, the channel matrix could be represented as $\check{\mathrm{H}}[u,p,q] \in \mathbb{C}^{N_R \times N_T}$ corresponding to the uth user for transmit sub-carrier q and receive sub-carrier p. Consider a MU MIMO OFDM downlink system where a base station is catering to U users having N_R receive antennas each. The system model is a generalization of (2.42) and it is given by

$$\underline{\check{\mathbf{y}}}[u,p] = \check{\mathrm{H}}[u,p,p]\underline{\check{\mathbf{x}}}[p] + \underbrace{\sum_{q=0, q\neq p}^{N_{sc}-1} \check{\mathrm{H}}[u,p,q]\underline{\check{\mathbf{x}}}[q]}_{\text{ICI for } u\text{th user}} + \underline{\check{\mathbf{n}}}[u,p] \tag{2.45}$$

for $p, q = 0, \ldots, N_{sc} - 1$ and $u = 1, \ldots, U$. The entities $\underline{\check{\mathbf{y}}}[u,p] \in \mathbb{C}^{N_R \times 1}$ and $\underline{\check{\mathbf{n}}}[u,p] \in \mathbb{C}^{N_R \times 1}$ represent the received signal vector and noise vector on sub-carrier p for the uth user, respectively. Also, $\underline{\check{\mathbf{x}}}[q] \in \mathbb{C}^{N_T \times 1}$ denotes the transmit vector from the base station at sub-carrier q, which is given by [144]:

$$\underline{\check{\mathbf{x}}}[q] = \sum_{u'=1}^{U} \mathrm{M}[u',q]\underline{\check{\mathbf{d}}}[u',q] \tag{2.46}$$

where $\mathrm{M}[u',q] \in \mathbb{C}^{N_T \times N_T}$ denotes the precoding matrix used to transmit data vector $\underline{\check{\mathbf{d}}}[u',q] \in \mathbb{C}^{N_T \times 1}$ to user u' on sub-carrier q. Hence the system model of (2.45) becomes

$$\underline{\check{\mathbf{y}}}[u,p] = \sum_{q=0}^{N_{sc}-1} \check{\mathrm{H}}[u,p,q] \Big(\sum_{u'=1}^{U} \mathrm{M}[u',q]\underline{\check{\mathbf{d}}}[u',q] \Big) + \underline{\check{\mathbf{n}}}[u,p]. \tag{2.47}$$

Let $\check{\mathrm{H}}[u,p,q] \cdot \mathrm{M}[u',q] = \mathrm{G}[u,u',p,q] \in \mathbb{C}^{N_R \times N_T}$ denote the equivalent channel between the transmit data vector $\underline{\check{\mathbf{d}}}[u',q]$ and the receive vector $\underline{\check{\mathbf{y}}}[u,p]$, then the input/output relation can be written as :

$$\underline{\check{\mathbf{y}}}[u,p] = \sum_{q=0}^{N_{sc}-1} \sum_{u'=1}^{U} \mathrm{G}[u,u',p,q]\underline{\check{\mathbf{d}}}[u',q] + \underline{\check{\mathbf{n}}}[u,p]. \tag{2.48}$$

In this case, the output and noise elements can be rearranged into order three tensors $\check{\mathcal{Y}}, \check{\mathcal{N}} \in \mathbb{C}^{U\times N_R \times N_{sc}}$. The components $\underline{\check{\mathbf{y}}}_{n_r}[u,p]$ and $\underline{\check{\mathbf{n}}}_{n_r}[u,p]$ are mapped to elements of third order tensors, denoted by $\check{\mathcal{Y}}_{u,n_r,p}$ and $\check{\mathcal{N}}_{u,n_r,p}$ respectively. Similarly, the input can be rearranged as an order three tensor $\check{\mathcal{D}} \in \mathbb{C}^{U\times N_T\times N_{sc}}$ where $\underline{\check{\mathbf{d}}}_{n_t}[u',q]$ is mapped to $\check{\mathcal{D}}_{u',n_t,q}$. Subsequently, the channel can be represented as an order 6 tensor $\check{\mathcal{G}} \in \mathbb{C}^{U\times N_R\times N_{sc}\times U\times N_T\times N_{sc}}$ where each element of matrix $\mathrm{G}_{n_r,n_t}[u,u',p,q]$ from (2.47) is mapped to an element $\check{\mathcal{G}}_{u,n_r,p,u',n_t,q}$ of the sixth order tensor channel. The system model then becomes :

$$\check{\mathcal{Y}} = \check{\mathcal{G}} *_3 \check{\mathcal{D}} + \check{\mathcal{N}}. \tag{2.49}$$

The tensor model represented by (2.49) can be considered as an evolution of the common matrix MIMO model in the space domain only, to a tensor MIMO model that in addition to space encapsulates also the frequency and user domains.

2.3.2 Cellular Networks

In cellular networks, the cell index can also be incorporated as a domain in the system model when represented through the tensor framework. In such a case, the channel tensor will include terms corresponding to the inter-cell interference as well. Assume a K cell MIMO Interfering Broadcast Channel (IBC) as in [145] where each cell consists of a Base Station (BS) with M antennas and U users with L antennas each. The channel between the kth base station and the uth user in the ith cell is denoted by a matrix $\mathbf{H}^{(k,i,u)} \in \mathbb{C}^{L\times M}$. Let $\underline{\mathbf{s}}^{(k)} \in \mathbb{C}^{M\times 1}$ be the broadcast transmitted signal vector by the kth base station, which is intended to be received by all the users within its cell. Then the received signal vector at the uth user in the ith cell can be written as [145]:

$$\underline{\mathbf{y}}^{(i,u)} = \sum_{k=1}^{K} \mathbf{H}^{(k,i,u)}\underline{\mathbf{s}}^{(k)} + \underline{\mathbf{z}}^{(i,u)} \tag{2.50}$$

where $\underline{\mathbf{z}}^{(i,u)} \in \mathbb{C}^{L\times 1}$ denotes the additive noise vector for the uth user in the ith cell. Notice that the summation in (2.50) includes not only the desired term (corresponding to $k = i$), but also the interference terms received by a user from a base station outside its cell (inter-cell interference corresponding to $k \neq i$). The system model in (2.50) can be represented using the tensor model from (2.35). Consider the transmit signal corresponding to K base stations with M antennas each as an order 2 tensor $\mathbf{S} \in \mathbb{C}^{M\times K}$ where each vector $\underline{\mathbf{s}}^{(k)}$ forms a column of the matrix $\mathbf{S}$. Similarly the output and noise received by U users in each of the K cells with L antennas per user can be defined as order 3 tensors $\boldsymbol{\mathcal{Y}} \in \mathbb{C}^{L\times K\times U}$ and $\boldsymbol{\mathcal{Z}} \in \mathbb{C}^{L\times K\times U}$ respectively such that $\boldsymbol{\mathcal{Y}}_{:,i,u} = \underline{\mathbf{y}}^{(i,u)}$ and $\boldsymbol{\mathcal{Z}}_{:,i,u} = \underline{\mathbf{z}}^{(i,u)}$. The channel can be defined as an order 5 tensor $\mathcal{H} \in \mathbb{C}^{L\times K\times U\times M\times K}$ such that $\mathcal{H}_{:,i,u,:,k} = \mathrm{H}^{(k,i,u)}$. Hence the system model from (2.50) can be equivalently expressed using the Einstein product as :

$$\boldsymbol{\mathcal{Y}} = \mathcal{H} *_2 \mathbf{S} + \boldsymbol{\mathcal{Z}}, \tag{2.51}$$

where the channel is expressed as an order-5 tensor.

2.3.3 MIMO GFDM and Multi-user MIMO GFDM

GFDM is a block based multi-carrier modulation scheme where each GFDM symbol consists of complex valued data symbols $d_{k,m}$ distributed over K sub-carriers and M timeslots known as sub-symbols. Hence each GFDM block consists a total of $N = KM$ complex symbols. A GFDM modulated signal for a SISO system is given as [18] :

$$x_n = \sum_{k=0}^{K-1}\sum_{m=0}^{M-1} d_{k,m}\, \underbrace{g[(n-mK)_{\bmod N}]\exp(j2\pi kn/K)}_{g_{k,m}[n]} \tag{2.52}$$

for $n = 0, 1, \ldots, N-1$, where $g_{k,m}[n]$ is the shifted version of the prototype filter impulse response $g[n]$ to the mth sub-symbol which is modulated on the kth sub-carrier. The modulo N operation makes $g_{k,m}[n]$ a circularly shifted version of $g_{k,0}[n]$. Collecting the filter samples in a vector $\underline{\mathrm{g}}_{k,m} = (g_{k,m}[0], g_{k,m}[1], \ldots, g_{k,m}[N-1])^T \in \mathbb{C}^{N\times 1}$, a transmit matrix $\mathrm{A} \in \mathbb{C}^{N\times N}$ can be defined as $\mathrm{A} = (\underline{\mathrm{g}}_{0,0}, \ldots, \underline{\mathrm{g}}_{K-1,0}, \underline{\mathrm{g}}_{0,1}, \ldots, \underline{\mathrm{g}}_{0,M-1}, \ldots, \underline{\mathrm{g}}_{K-1,M-1})$. With $\underline{\mathbf{d}} = (d_{0,0}, \ldots, d_{K-1,0}, d_{0,1}, \ldots, d_{0,M-1}, \ldots, d_{K-1,M-1})^T$, as the $N \times 1$ data vector (2.52) can be written as,

$$\underline{\mathbf{x}} = \mathrm{A}\underline{\mathbf{d}} \tag{2.53}$$

where the vector $\underline{\mathbf{x}} \in \mathbb{C}^{N\times 1}$ is the transmitted signal corresponding to the GFDM data block $\underline{\mathbf{d}}$. After employing cyclic prefix between GFDM blocks to ensure no inter-block interference, the transmitted samples through a wireless channel are modelled as $\underline{\mathbf{y}} = \mathrm{C}\underline{\mathbf{x}} + \underline{\mathbf{n}} = \mathrm{CA}\underline{\mathbf{d}} + \underline{\mathbf{n}}$ [18] where $\mathrm{C} \in \mathbb{C}^{N\times N}$ is the circular channel convolution matrix. It is obtained from channel impulse response $\underline{\mathrm{h}} = [h[0], h[1], \ldots, h[N-1]]$ as $\mathrm{C}_{i,j} = h[(i-j)_{\mathrm{mod}\ N}]$. The vectors $\underline{\mathbf{y}}, \underline{\mathbf{n}} \in \mathbb{C}^{N\times 1}$ represent the received signal and noise vectors respectively. At the receiver, a matrix $\mathrm{B} \in \mathbb{C}^{N\times N}$ is used to process the received signal as

$$\tilde{\underline{\mathbf{d}}} = \mathrm{B}\underline{\mathbf{y}} = \underbrace{\mathrm{BCA}}_{\mathrm{H}}\,\underline{\mathbf{d}} + \mathrm{B}\underline{\mathbf{n}} = \mathrm{H}\underline{\mathbf{d}} + \tilde{\underline{\mathbf{n}}}. \tag{2.54}$$

Some common options for the matrix B includes a zero forcing, matched filter, or MMSE receiver matrix [18].

From (2.52) we see that the data symbols in a GFDM system are indexed by two indices. Hence it is more natural to represent such data as a matrix $\mathbf{D} \in \mathbb{C}^{K\times M}$ with elements $\mathbf{D}_{k,m} = d_{k,m}$. The transmit filtering operation can be expressed using a tensor $\mathcal{A} \in \mathbb{C}^{N\times K\times M}$, where $\mathcal{A}_{n,k,m} = g_{k,m}[n]$. Hence (2.52) can be written in tensor form as

$$\underline{\mathbf{x}} = \mathcal{A} *_2 \mathbf{D} \tag{2.55}$$

and the received signal can be written as $\underline{\mathbf{y}} = \mathrm{C} *_1 \mathcal{A} *_2 \mathbf{D} + \underline{\mathbf{n}}$. At the receiver, a tensor $\mathcal{B} \in \mathbb{C}^{K\times M\times N}$ can be employed to process the received signal as

$$\tilde{\mathbf{D}} = \mathcal{B} *_1 \underline{\mathbf{y}} = \mathcal{B} *_1 \mathrm{C} *_1 \mathcal{A} *_2 \mathbf{D} + \mathcal{B} *_1 \underline{\mathbf{n}} = \mathcal{H} *_2 \mathbf{D} + \mathbf{N} \tag{2.56}$$

where $\tilde{\mathbf{D}}, \mathbf{N} \in \mathbb{C}^{K\times M}$ denote the output data and noise matrices respectively. The tensor $\mathcal{H} \in \mathbb{C}^{K\times M\times K\times M}$ is the equivalent channel between the input and output data matrices which includes a cascade of transmit filter, physical channel and the receive filter.

Such a system model can be further extended to represent a MIMO GFDM system. Let S independent streams of data be transmitted using K sub-carriers, M sub-symbols and N_T transmit antennas. Let N_R denotes the number of receive antennas. The data matrix $\mathbf{D}^{(s)}$ to be transmitted on the sth stream forms a slice of an order-3 input data tensor

$\boldsymbol{\mathcal{D}} \in \mathbb{C}^{S\times K\times M}$ for $s = 1,\ldots,S$. The transmit filtering generates a vector $\mathbf{x}^{(n_t)} \in \mathbb{C}^{N\times 1}$ corresponding to each antenna. All such vectors for $n_t = 1,\ldots,N_T$ can be represented as columns of a matrix $\mathbf{X} \in \mathbb{C}^{N_T\times N}$. The transmit filter tensor that generates $\mathbf{X}$ can now be represented using a tensor $\mathcal{H}_T \in \mathbb{C}^{N_T\times N\times S\times K\times M}$ where $\mathcal{H}_{T n_t,:,s,:,:} = \mathcal{A}$, assuming that same pulse shape filter is used for all the transmit antennas. If $S = N_T$ and transmit data of each stream is mapped directly to a corresponding antenna, then $\mathcal{H}_T$ contains zeros when $s \neq n_t$. The input data tensor is converted into the transmit matrix as $\mathbf{X} = \mathcal{H}_T *_3 \boldsymbol{\mathcal{D}}$. It passes through the channel $\mathcal{H}_C \in \mathbb{C}^{N_R\times N\times N_T\times N}$ where $\mathcal{H}_{C n_r,:,n_t,:} = \mathrm{C}^{(n_r,n_t)}$ and $\mathrm{C}^{(n_r,n_t)} \in \mathbb{C}^{N\times N}$ represents the circular channel convolution matrix obtained from channel impulse response between n_rth receive and n_tth transmit antenna. Thus the received signal matrix is given as $\mathbf{Y} = \mathcal{H}_C *_2 \mathcal{H}_T *_3 \boldsymbol{\mathcal{D}} + \mathbf{V}$ where $\mathbf{V} \in \mathbb{C}^{N_R\times N}$ is the received noise matrix. With the receive filter represented using $\mathcal{H}_R \in \mathbb{C}^{S\times K\times M\times N_R\times N}$, the received signal matrix $\mathbf{Y}$ is converted to the output tensor $\tilde{\boldsymbol{\mathcal{D}}} \in \mathbb{C}^{S\times K\times M}$ containing the received data elements on each stream, sub-carrier and sub-symbol, i.e.

$$\tilde{\boldsymbol{\mathcal{D}}} = \mathcal{H}_R *_2 \mathbf{Y} = \underbrace{\mathcal{H}_R *_2 \mathcal{H}_C *_2 \mathcal{H}_T}_{\mathcal{H}} *_3 \boldsymbol{\mathcal{D}} + \boldsymbol{\mathcal{N}} \tag{2.57}$$

where $\mathcal{H} \in \mathbb{C}^{S\times K\times M\times S\times K\times M}$ represents the sixth order equivalent channel between the input and the output tensors and the noise tensor is given by $\boldsymbol{\mathcal{N}} = \mathcal{H}_R *_2 \mathbf{V}$. The system model developed in (2.57) expresses the output data elements as a linear combination of all the elements of the input tensor with the help of Einstein product. The coefficients of this linear combination are encapsulated in the equivalent channel $\mathcal{H}$ which thereby contains all the multi-domain interference terms. A model for such a MIMO GFDM system has been considered in [26] where the complex symbols corresponding to each sub-carrier, sub-symbol and antenna for both transmitter and receiver are arranged in single transmit and receive vectors of size $N \cdot N_T$ and $N \cdot N_R$ respectively, and channel as a matrix of size $N \cdot N_R \times N \cdot N_T$. Such a model involves a large matrix where several domain symbols have been merged and hence the distinction between them has been obscured. However, the tensor model maintains the identifiability of all the domains.

The overall system model is represented as

$$\tilde{\boldsymbol{\mathcal{D}}} = \mathcal{H} *_3 \boldsymbol{\mathcal{D}} + \boldsymbol{\mathcal{N}} \tag{2.58}$$

and it is illustrated in Figure 2.7 which represents the system model for a MIMO GFDM system with 2 transmit and 2 receive antennas. In Figure 2.7 a matrix is shown as a square and a higher order tensor as a double-line square with its order written on top right corner. A third order tensor is represented as a cube with staggered edges. The data corresponding to each antenna for all the K sub-carriers and M sub-symbols is represented as a $K \times M$ matrix, where the matrices corresponding to each antenna form slices of the third order input tensor.

Multi-user MIMO GFDM

The MU MIMO GFDM system model has been presented in [146, 147] where a matrix channel based approach has been used. Consider an uplink scenario with U users each with N_T transmit antennas, and a base station equipped with N_R receive antennas. Let $\underline{\mathbf{d}}^{(n_t,u)} \in \mathbb{C}^{KM\times 1}$ denote the data vector to be transmitted from n_tth transmit antenna of the uth user. The transmitted GFDM symbol is generated as [147]:

$$\underline{\mathbf{x}}^{(n_t,u)} = \mathrm{A}\underline{\mathbf{d}}^{(n_t,u)} \tag{2.59}$$

where $\underline{\mathbf{x}}^{(n_t,u)} \in \mathbb{C}^{KM\times 1}$ and A is the $KM \times KM$ transmitter matrix as in (2.53). The received symbol at the base station is given as :

$$\underline{\mathbf{y}}^{(n_r)} = \sum_{u=1}^{U}\sum_{n_t=1}^{N_T} \mathrm{C}^{(n_r,n_t,u)}\underline{\mathbf{x}}^{(n_t,u)} + \underline{\mathbf{n}}^{(n_r)} \tag{2.60}$$

where $\underline{\mathbf{y}}^{(n_r)} \in \mathbb{C}^{KM\times 1}$ and $\mathrm{C}^{(n_r,n_t,u)} \in \mathbb{C}^{KM\times KM}$ is the circular convolution matrix generated from the channel impulse response [147]. The received vector for different antennas can be concatenated to form :

$$\underbrace{\begin{bmatrix} \underline{\mathbf{y}}^{(1)} \\ \vdots \\ \underline{\mathbf{y}}^{(N_R)} \end{bmatrix}}_{\tilde{\underline{\mathbf{y}}}} = \sum_{u=1}^{U} \underbrace{\begin{bmatrix} \mathrm{C}^{(1,1,u)}\mathrm{A} & \dots & \mathrm{C}^{(1,N_T,u)}\mathrm{A} \\ \vdots & \ddots & \vdots \\ \mathrm{C}^{(N_R,1,u)}\mathrm{A} & \dots & \mathrm{H}^{(N_R,N_T,u)}\mathrm{A} \end{bmatrix}}_{\tilde{\mathrm{C}}^{(u)}} \underbrace{\begin{bmatrix} \underline{\mathbf{d}}^{(1,u)} \\ \vdots \\ \underline{\mathbf{d}}^{(N_T,u)} \end{bmatrix}}_{\underline{\tilde{\mathbf{d}}}^{(u)}} + \underbrace{\begin{bmatrix} \underline{\mathbf{n}}^{(1)} \\ \vdots \\ \underline{\mathbf{n}}^{(N_R)} \end{bmatrix}}_{\tilde{\underline{\mathbf{n}}}} \tag{2.61}$$

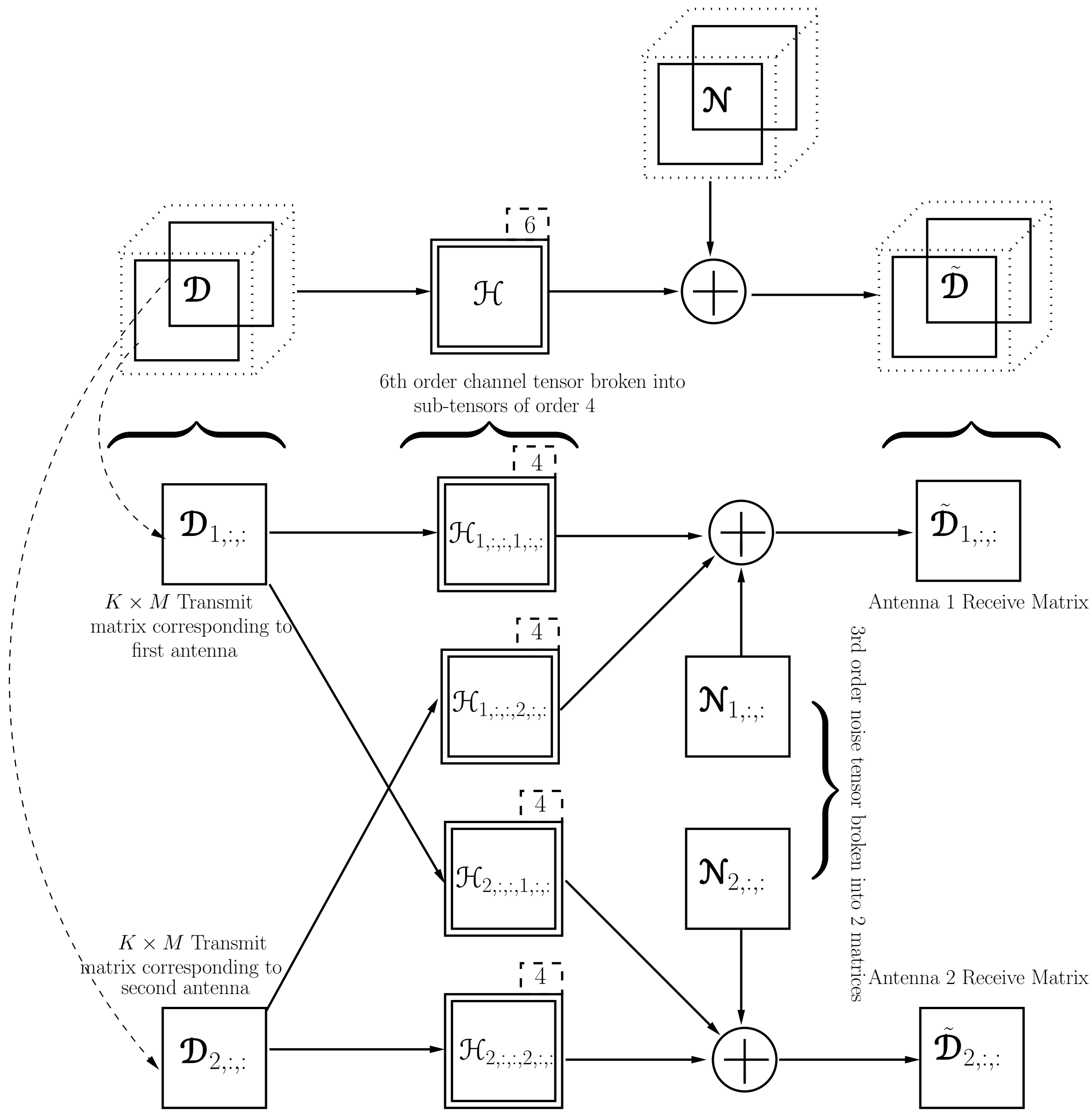


Fig. 2.7: Tensor system model for MIMO GFDM with 2 antennas ($N_T = N_R = 2$).

where $\underline{\tilde{\mathbf{y}}}$ and $\underline{\tilde{\mathbf{n}}}$ are vectors of size $K \cdot M \cdot N_R$. Demodulation is performed at the receiver such that the resulting estimated vector is given by :

$$\underline{\hat{\mathbf{d}}} = \mathrm{W}\underline{\tilde{\mathbf{y}}} \tag{2.62}$$

where $\underline{\hat{\mathbf{d}}} \in \mathbb{C}^{KMN_TU\times 1}$ contains the estimated transmitted data from all users. A specific structure of the matrix $\mathrm{W} \in \mathbb{C}^{KMN_TU\times KMN_R}$ for joint detection and demodulation based on MMSE filtering is presented in [147]. In essence, the model explained so far is concatenating the data corresponding to sub-symbols, sub-carrier, antennas and users into a single vector. A more intuitive representation of such a system keeping the structure of the input and output intact can be provided by using tensors. The transmitted and estimated data represented as $\underline{\mathbf{d}}^{(n_t,u)} \in \mathbb{C}^{KM\times 1}$ and $\underline{\hat{\mathbf{d}}} \in \mathbb{C}^{KMN_TU\times 1}$ respectively can be set as fourth order tensors $\boldsymbol{\mathcal{D}}, \hat{\boldsymbol{\mathcal{D}}} \in \mathbb{C}^{K\times M\times N_T\times U}$. The received signal represented earlier by $\underline{\tilde{\mathbf{y}}} \in \mathbb{C}^{KMN_R\times 1}$ is now a third order tensor $\boldsymbol{\mathcal{Y}} \in \mathbb{C}^{K\times M\times N_R}$, and the effective channel is a seventh order tensor $\mathcal{H} \in \mathbb{C}^{K\times M\times N_R\times K\times M\times N_T\times U}$. The input/output system relation is :

$$\boldsymbol{\mathcal{Y}} = \mathcal{H} *_4 \boldsymbol{\mathcal{D}} + \boldsymbol{\mathcal{N}} \tag{2.63}$$

where $\boldsymbol{\mathcal{N}}$ is the noise tensor of same size as $\boldsymbol{\mathcal{Y}}$. The joint estimation operation can be achieved through tensor $\mathcal{W} \in \mathbb{C}^{K\times M\times N_T\times U\times K\times M\times N_R}$ as :

$$\hat{\boldsymbol{\mathcal{D}}} = \mathcal{W} *_3 \boldsymbol{\mathcal{Y}} = \underbrace{\mathcal{W} *_3 \mathcal{H}}_{\text{channel}} *_4 \boldsymbol{\mathcal{D}} + \underbrace{\mathcal{W} *_3 \boldsymbol{\mathcal{N}}}_{\text{noise}}. \tag{2.64}$$

Hence we can see that the effective channel between $\boldsymbol{\mathcal{D}}$ and $\hat{\boldsymbol{\mathcal{D}}}$, represented by $\mathcal{W} *_3 \mathcal{H}$ is an eighth order tensor of size $K\times M\times N_T\times U\times K\times M\times N_T\times U$ where all the domains being used namely sub-carriers, sub-symbols, antennas and users have been distinctly incorporated.

Through all these examples, we can clearly see how the tensor framework facilitates incorporating multiple domains in the system model in a systematic and intuitive manner while maintaining the distinction between them. Similarly several other multi-domain systems such as MIMO Digital Subscriber Line (DSL) [148], MIMO CDMA [149], MIMO FBMC [27] etc., can be represented using the tensor based system model.

Chapter 3

Shannon Capacity of the Tensor Channel

In this chapter, we find the Shannon capacity of higher order tensor channels associated with any multi-domain communication system. We assume that the channel is known at the transmitter and receiver, and is deterministic.

3.1 Information Theoretic Notions for Tensors

Throughout this chapter, we assume the noise tensor from (2.35) to be modelled using circularly symmetric complex Gaussian distribution. For the system model in (2.35), we will now show that in the presence of circularly symmetric complex Gaussian noise, the input distribution that achieves the channel capacity is also circularly symmetric complex Gaussian. Thus, the following section presents some results concerning the circularly symmetric complex Gaussian distributed tensors.

3.1.1 Differential Entropy of circularly symmetric complex Gaussian tensor

A complex random tensor $\mathcal{X}$ is defined as *circular* if it is rotationally invariant, i.e. $\mathcal{X}$ and $\mathcal{Y} = e^{j\alpha}\mathcal{X}$ have the same probability distribution for any given real α. A complex Gaussian random vector is circularly symmetric if and only if it is zero mean and proper [121]. This statement can be extended to tensor case also :

Lemma 3. *A complex Gaussian random tensor is circularly symmetric if and only if it is zero mean and proper.*

The proof of Lemma 3 directly follows from the definition of proper and circularly symmetric tensors. The distribution of a circularly symmetric complex Gaussian tensor $\boldsymbol{\mathcal{X}} \in \mathbb{C}^{I_1 \times \ldots \times I_N}$ with covariance $\mathcal{Q}$ is given by (2.25), with $\bar{\mathcal{M}} = 0_{\mathcal{T}}$. Subsequently, the differential entropy of such a tensor is given by :

$$
\begin{aligned}
\mathcal{H}(\boldsymbol{\mathcal{X}}) &= \mathbb{E}[-\log p_{\boldsymbol{\mathcal{X}}}(\boldsymbol{\mathcal{X}})] \\
&= \mathbb{E}\Big[-\log\Big\{\frac{\exp\left(-\boldsymbol{\mathcal{X}}^* *_N \mathcal{Q}^{-1} *_N \boldsymbol{\mathcal{X}}\right)}{(\pi)^{I_1 I_2 \ldots I_N}\det(\mathcal{Q})}\Big\}\Big] \\
&= \log\Big((\pi)^{I_1 \ldots I_N}\det(\mathcal{Q})\Big) + (\log e)\mathbb{E}\Big[((\boldsymbol{\mathcal{X}}^* *_N \mathcal{Q}^{-1}) *_N \boldsymbol{\mathcal{X}})\Big] \\
&= \log\Big((\pi)^{I_1 \ldots I_N}\det(\mathcal{Q})\Big) + (\log e)\mathbb{E}\Big[\operatorname{tr}\left(\boldsymbol{\mathcal{X}} \circ (\boldsymbol{\mathcal{X}}^* *_N \mathcal{Q}^{-1})\right)\Big] \quad \text{(from (2.18))} \\
&= \log\Big((\pi)^{I_1 \ldots I_N}\det(\mathcal{Q})\Big) + (\log e)\operatorname{tr}\big(\underbrace{\mathbb{E}[\boldsymbol{\mathcal{X}} \circ \boldsymbol{\mathcal{X}}^*] *_N \mathcal{Q}^{-1}}_{\text{identity tensor}}\big) \quad \text{(from associativity rule, (2.9))} \\
&= \log\Big((e\pi)^{I_1 \ldots I_N}\det(\mathcal{Q})\Big). \qquad (3.1)
\end{aligned}
$$

Lemma 4. *Let $\boldsymbol{\mathcal{X}} \in \mathbb{C}^{I_1 \times \ldots \times I_N}$ be a zero mean circularly symmetric complex Gaussian random tensor with covariance tensor $\mathcal{Q}$. Let $\boldsymbol{\mathcal{Y}} \in \mathbb{C}^{I_1 \times \ldots \times I_N}$ be another zero-mean random tensor with same covariance tensor. Then, $\mathcal{H}(\boldsymbol{\mathcal{X}}) \geq \mathcal{H}(\boldsymbol{\mathcal{Y}})$.*

Proof. Let $p_{\boldsymbol{\mathcal{X}}}(\mathcal{X})$ and $p_{\boldsymbol{\mathcal{Y}}}(\mathcal{Y})$ be the density functions of $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{Y}}$ respectively where $p_{\boldsymbol{\mathcal{X}}}(\mathcal{X})$ is given by (2.25) with $\bar{\mathcal{M}} = 0_{\mathcal{T}}$. Since $\mathbb{E}_{\boldsymbol{\mathcal{X}}}[\boldsymbol{\mathcal{X}} \circ \boldsymbol{\mathcal{X}}^*] = \mathbb{E}_{\boldsymbol{\mathcal{Y}}}[\boldsymbol{\mathcal{Y}} \circ \boldsymbol{\mathcal{Y}}^*] = \mathcal{Q}$, we have $\mathcal{H}(\boldsymbol{\mathcal{X}}) = -\mathbb{E}_{\boldsymbol{\mathcal{X}}}[\log p_{\boldsymbol{\mathcal{X}}}(\boldsymbol{\mathcal{X}})] = -\mathbb{E}_{\boldsymbol{\mathcal{Y}}}[\log p_{\boldsymbol{\mathcal{X}}}(\boldsymbol{\mathcal{Y}})]$. Therefore,

$$
\begin{aligned}
\mathcal{H}(\boldsymbol{\mathcal{X}}) - \mathcal{H}(\boldsymbol{\mathcal{Y}}) &= -\mathbb{E}_{\boldsymbol{\mathcal{Y}}}[\log p_{\boldsymbol{\mathcal{X}}}(\boldsymbol{\mathcal{Y}})] + \mathbb{E}_{\boldsymbol{\mathcal{Y}}}[\log p_{\boldsymbol{\mathcal{Y}}}(\boldsymbol{\mathcal{Y}})] \\
&= \mathbb{E}_{\boldsymbol{\mathcal{Y}}}\Big[-\log\frac{p_{\boldsymbol{\mathcal{X}}}(\boldsymbol{\mathcal{Y}})}{p_{\boldsymbol{\mathcal{Y}}}(\boldsymbol{\mathcal{Y}})}\Big] \geq \mathbb{E}_{\boldsymbol{\mathcal{Y}}}\Big[1 - \frac{p_{\boldsymbol{\mathcal{X}}}(\boldsymbol{\mathcal{Y}})}{p_{\boldsymbol{\mathcal{Y}}}(\boldsymbol{\mathcal{Y}})}\Big]\frac{1}{\ln 2} = 0 \\
\Rightarrow \mathcal{H}(\boldsymbol{\mathcal{X}}) &\geq \mathcal{H}(\boldsymbol{\mathcal{Y}}).
\end{aligned}
$$

□

Lemma 4 essentially implies that for a given covariance tensor, a circularly symmetric complex Gaussian distribution is the entropy maximizer. Any non-Gaussian or non-circular input distribution will lead to a lower entropy.

3.1.2 Mutual Information

We now derive the output covariance and pseudo-covariance for the system model defined in (2.35) in terms of the input and noise covariances and pseudo-covariances. For (2.35), the output covariance tensor is given by

$$\mathcal{Q}_{\boldsymbol{\mathcal{Y}}} = \text{cov}(\boldsymbol{\mathcal{Y}}) = \mathbb{E}[\boldsymbol{\mathcal{Y}} \circ \boldsymbol{\mathcal{Y}}^*] = \mathbb{E}[(\mathcal{H} *_N \boldsymbol{\mathcal{X}} + \boldsymbol{\mathcal{N}}) \circ (\mathcal{H}^* *_N \boldsymbol{\mathcal{X}}^* + \boldsymbol{\mathcal{N}}^*)] \tag{3.2}$$

$$\mathcal{Q}_{\boldsymbol{\mathcal{Y}}} = \underbrace{\mathbb{E}[(\mathcal{H} *_N \boldsymbol{\mathcal{X}}) \circ (\mathcal{H}^* *_N \boldsymbol{\mathcal{X}}^*)]}_{\text{Main term}} + \underbrace{\mathbb{E}[\boldsymbol{\mathcal{N}} \circ \boldsymbol{\mathcal{N}}^*]}_{\text{Noise Covariance}} + \underbrace{\mathbb{E}[(\mathcal{H} *_N \boldsymbol{\mathcal{X}}) \circ \boldsymbol{\mathcal{N}}^*] + \mathbb{E}[\boldsymbol{\mathcal{N}} \circ (\mathcal{H}^* *_N \boldsymbol{\mathcal{X}}^*)]}_{\text{cross terms}} \tag{3.3}$$

Assuming $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{N}}$ are zero mean and independent, the cross terms will be zero. Based on the commutativity rule (2.10), we get $\mathcal{H}^* *_N \boldsymbol{\mathcal{X}}^* = \boldsymbol{\mathcal{X}}^* *_N (\mathcal{H}^*)^T = \boldsymbol{\mathcal{X}}^* *_N \mathcal{H}^H$. Using the associativity rule (2.9), we get :

$$\begin{aligned}\mathcal{Q}_{\boldsymbol{\mathcal{Y}}} &= \mathbb{E}[(\mathcal{H} *_N \boldsymbol{\mathcal{X}}) \circ (\boldsymbol{\mathcal{X}}^* *_N \mathcal{H}^H)] + \mathcal{Q}_{\boldsymbol{\mathcal{N}}} \\ &= (\mathcal{H} *_N \mathbb{E}[\boldsymbol{\mathcal{X}} \circ \boldsymbol{\mathcal{X}}^*] *_N \mathcal{H}^H) + \mathcal{Q}_{\boldsymbol{\mathcal{N}}} \\ &= \mathcal{H} *_N \mathcal{Q}_{\boldsymbol{\mathcal{X}}} *_N \mathcal{H}^H + \mathcal{Q}_{\boldsymbol{\mathcal{N}}} \end{aligned} \tag{3.4}$$

where $\mathcal{Q}_{\boldsymbol{\mathcal{X}}}$ and $\mathcal{Q}_{\boldsymbol{\mathcal{N}}}$ are the input and noise covariance tensors respectively. Similarly, the output pseudo-covariance tensor can be derived as:

$$\tilde{\mathcal{Q}}_{\boldsymbol{\mathcal{Y}}} = \mathbb{E}[\boldsymbol{\mathcal{Y}} \circ \boldsymbol{\mathcal{Y}}] = \mathbb{E}[(\mathcal{H} *_N \boldsymbol{\mathcal{X}} + \boldsymbol{\mathcal{N}}) \circ (\mathcal{H} *_N \boldsymbol{\mathcal{X}} + \boldsymbol{\mathcal{N}})] = \mathcal{H} *_N \tilde{\mathcal{Q}}_{\boldsymbol{\mathcal{X}}} *_N \mathcal{H}^T + \tilde{\mathcal{Q}}_{\boldsymbol{\mathcal{N}}} \tag{3.5}$$

where $\tilde{\mathcal{Q}}_{\boldsymbol{\mathcal{X}}}$ and $\tilde{\mathcal{Q}}_{\boldsymbol{\mathcal{N}}}$ are the input and noise pseudo-covariance tensors respectively. Since we consider noise to be circularly symmetric, its pseudo-covariance is $0_{\mathcal{T}}$. Further, if input pseudo-covariance is also $0_{\mathcal{T}}$, then the output pseudo-covariance will also be $0_{\mathcal{T}}$. Subsequently, the following lemma can be established:

Lemma 5. *If $\boldsymbol{\mathcal{X}} \in \mathbb{C}^{I_1 \times I_2 \times \ldots \times I_N}$ and $\boldsymbol{\mathcal{N}} \in \mathbb{C}^{J_1 \times \ldots \times J_M}$ are independent circularly symmetric complex Gaussian tensors, then $\boldsymbol{\mathcal{Y}} = \mathcal{H} *_N \boldsymbol{\mathcal{X}} + \boldsymbol{\mathcal{N}}$ for any deterministic tensor $\mathcal{H} \in \mathbb{C}^{J_1 \times J_2 \times \ldots \times J_M \times I_1 \times I_2 \times \ldots \times I_N}$, is also circularly symmetric complex Gaussian.*

Proof. Since $\mathcal{X}$, and $\mathcal{N}$ are circularly symmetric Gaussian tensors, from Lemma 3 we know they are zero mean and proper, i.e.
$\mathbb{E}[\mathcal{X}] = 0_{\mathcal{T}}$, $\mathbb{E}[\mathcal{N}] = 0_{\mathcal{T}}$, $\tilde{\mathcal{Q}}_{\mathcal{X}} = 0_{\mathcal{T}}$, and $\tilde{\mathcal{Q}}_{\mathcal{N}} = 0_{\mathcal{T}}$. The tensor $\mathcal{Y}$ will also be Gaussian, and in order to show that it would be circularly symmetric, we need to show that it has zero mean and zero pseudo-covariance. Hence, we have

$$\mathbb{E}[\mathcal{Y}] = \mathcal{H} *_N \mathbb{E}[\mathcal{X}] + \mathbb{E}[\mathcal{N}] = 0_{\mathcal{T}}. \tag{3.6}$$

Also, the pseudo-covariance of $\mathcal{Y}$ from (3.5) is given as:

$$\tilde{\mathcal{Q}}_{\mathcal{Y}} = \mathcal{H} *_N 0_{\mathcal{T}} *_N \mathcal{H}^T + 0_{\mathcal{T}} = 0_{\mathcal{T}} \tag{3.7}$$

which concludes the proof. □

Since the input tensor $\mathcal{X}$ and the noise tensor $\mathcal{N}$ are assumed independent, we can write the mutual information between the input and output tensors as :

$$\mathcal{I}(\mathcal{X};\mathcal{Y}) = \mathcal{H}(\mathcal{Y}) - \mathcal{H}(\mathcal{Y}|\mathcal{X}) \tag{3.8}$$

$$= \mathcal{H}(\mathcal{Y}) - \mathcal{H}(\mathcal{N}) \tag{3.9}$$

$$= \mathcal{H}(\mathcal{Y}) - \log\left((e\pi)^{J_1 J_2 \cdots J_M} \det(\mathcal{Q}_{\mathcal{N}})\right). \tag{3.10}$$

Based on Lemma 4 and the received covariance derived in (3.4), we can write :

$$\mathcal{H}(\mathcal{Y}) \leq \log\left((e\pi)^{J_1 J_2 \cdots J_M} \det(\mathcal{H} *_N \mathcal{Q}_{\mathcal{X}} *_N \mathcal{H}^H + \mathcal{Q}_{\mathcal{N}})\right) \tag{3.11}$$

$$\Rightarrow \mathcal{I}(\mathcal{X};\mathcal{Y}) \leq \log\left[\frac{\det\left(\mathcal{H} *_N \mathcal{Q}_{\mathcal{X}} *_N \mathcal{H}^H + \mathcal{Q}_{\mathcal{N}}\right)}{\det(\mathcal{Q}_{\mathcal{N}})}\right] \tag{3.12}$$

where equality is achieved only if $\mathcal{Y}$ is circularly symmetric Gaussian distributed.

3.2 Capacity of a Fixed Tensor Channel

In order to find the Shannon capacity of the tensor channel, it is required to maximize the mutual information between the input and the output tensors over input distributions under possible constraints. We assume that the tensor channel is known and the noise tensor is zero-mean circularly symmetric complex Gaussian distributed having independent components with variance σ^2, and hence the noise covariance tensor is given by $\mathcal{Q}_{\mathcal{N}} = \sigma^2 \mathcal{I}_M$.

For simplicity, we assume $\sigma^2 = 1$. Now let us consider the mutual information inequality of (3.12) where equality is achieved only if $\boldsymbol{\mathcal{Y}}$ is circularly symmetric Gaussian distributed. In the proposed tensor system model (2.35), if $\boldsymbol{\mathcal{N}}$ is zero-mean circularly symmetric complex Gaussian, then using Lemma 5 implies that $\boldsymbol{\mathcal{Y}}$ will also be zero-mean circularly symmetric complex Gaussian if $\boldsymbol{\mathcal{X}}$ is so. Hence for maximizing the mutual information, we take $\boldsymbol{\mathcal{X}}$ as zero-mean circularly symmetric complex Gaussian with covariance $\mathcal{Q}_{\boldsymbol{\mathcal{X}}} = \mathcal{Q}$. Thus, with the noise covariance tensor as identity tensor $\mathcal{I}_M$, we get

$$\mathcal{I}(\boldsymbol{\mathcal{X}};\boldsymbol{\mathcal{Y}}) = \log\left[\det\left(\mathcal{H} *_N \mathcal{Q} *_N \mathcal{H}^H + \mathcal{I}_M\right)\right] \tag{3.13}$$

and the capacity is given by,

$$\begin{aligned} &\max_{\mathcal{Q}} \ \Big(\log\left[\det\left(\mathcal{H} *_N \mathcal{Q} *_N \mathcal{H}^H + \mathcal{I}_M\right)\right]\Big) \\ &s.t. \ \ f(\mathcal{Q}) \leq 0, \quad \mathcal{Q} \succeq 0. \end{aligned} \tag{3.14}$$

where the inequality constraint $f(\mathcal{Q}) \leq 0$ can represent a family of power constraints. At this point, it should be noted that for any Hermitian positive semi-definite tensor $\mathcal{A} \in \mathbb{C}^{I_1\times\ldots\times I_N\times I_1\times\ldots\times I_N}$, $\log[\det(\mathcal{A})]$ is a concave function. The proof has been provided in Appendix B.2. Thus the above optimization is a maximization of a concave function subject to a family of power constraints.

3.2.1 Family of Power Constraints

In a practical system, the power constraints on the transmit symbol may span multiple domains. For instance, in a transmission scheme employing the space, time, and frequency domains, instead of imposing power across the tensor symbol, individual power constraint might apply on each antenna, or each antenna and time slot, or each antenna, time slot and frequency bin. Using the tensor framework, we have the flexibility of mathematically representing a family of such power constraints.

First we introduce some notations for ease of understanding. Let the sequence of indices $(i_1, i_2, \ldots, i_N)$ be denoted as $\underline{i}$. Let $\underline{i}_c$ denotes the sequence of indices indicating tensor symbol elements under power constraint and let $\underline{i}_r$ represents the the indices in $\underline{i}$ which are not in $\underline{i}_c$. For example, in an order-5 tensor of size $I_1 \times I_2 \times I_3 \times I_4 \times I_5$, the sequence of all indices is denoted as $\underline{i} = (i_1, i_2, i_3, i_4, i_5)$. If the domains which are under individual power constraints are 1 and 3, then $\underline{i}_c = (i_1, i_3)$ and $\underline{i}_r = (i_2, i_4, i_5)$. With this choice of notations, we will express $\sum_{i_1=1}^{I_1} \sum_{i_3=1}^{I_3}$ as $\sum_{\underline{i}_c}$. Notations corresponding to the cases where either $\underline{i}_c$ or $\underline{i}_r$ is empty are explained in Table 3.1.

Table 3.1: Simplified notation for indices.

Cases	$\sum_{\underline{i}_r} \mathcal{Q}_{\underline{i},\underline{i}}$ denotes	$P_{\underline{i}_c}$ denotes	Interpretation
$\underline{i}_c$ is empty and $\underline{i}_r = \underline{i}$	$\sum_{\underline{i}} \mathcal{Q}_{\underline{i},\underline{i}} = \mathrm{tr}(\mathcal{Q})$	P	sum power constraint
$\underline{i}_r$ is empty and $\underline{i}_c = \underline{i}$	$\mathcal{Q}_{i_1,\ldots,i_N,i_1,\ldots,i_N}$	$P_{i_1,\ldots,i_N}$	per element constraints

Using these simplified notations, we will now formulate a family of optimization problems to find the tensor channel capacity which can cover different types of power constraints, as follows :

$$\max_{\mathcal{Q}} \quad \Big(\log \big[\det \left(\mathcal{H} *_N \mathcal{Q} *_N \mathcal{H}^H + \mathcal{I}_M \right) \big] \Big) \tag{3.15}$$

$$s.t. \quad \sum_{\underline{i}_r} \mathcal{Q}_{\underline{i},\underline{i}} \leq P_{\underline{i}_c} \quad \forall \underline{i}_c, \tag{3.16}$$

$$\mathcal{Q} \succeq 0. \tag{3.17}$$

To illustrate how the above framework can represent a large variety of constraints, let us consider a few specific cases. The case $\underline{i}_c = \underline{i}$, hence $\underline{i}_r$ is empty, will represent the situation where we have per element power constraints with $P_{\underline{i}_c} = P_{i_1,\ldots,i_N}$ and (3.16) becomes :

$$\mathcal{Q}_{i_1,\ldots,i_N,i_1,\ldots,i_N} \leq P_{i_1,\ldots,i_N}, \quad \forall i_1, i_2, \ldots, i_N. \tag{3.18}$$

When $\underline{i}_c = i_K$, where $K \leq N$, we have the case with per domain element constraint for the Kth domain where each element i_K has a different budget of P_{i_K}, i.e. $P_{\underline{i}_c} = P_{i_K}$ and (3.16) becomes :

$$\sum_{i_1=1}^{I_1} \cdots \sum_{i_{K-1}=1}^{I_{K-1}} \sum_{i_{K+1}=1}^{I_{K+1}} \cdots \sum_{i_N=1}^{I_N} \mathcal{Q}_{i_1,\ldots,i_K,\ldots,i_N,i_1,\ldots,i_K,\ldots,i_N} \leq P_{i_K}, \quad \forall i_K. \tag{3.19}$$

Now let us assume we have power constraints on two domains K and L such that $K < L \leq N$. Then $\underline{i}_c = (i_K, i_L)$ and $\underline{i}_r = (i_1, \ldots, i_{K-1}, i_{K+1}, \ldots i_{L-1}, i_{L+1}, \ldots, i_N)$. In this case

$P_{\underline{i}_c} = P_{i_K, i_L}$ and (3.16) becomes :

$$\sum_{i_1=1}^{I_1} \cdots \sum_{i_{K-1}=1}^{I_{K-1}} \sum_{i_{K+1}=1}^{I_{K+1}} \cdots \sum_{i_{L-1}=1}^{I_{L-1}} \sum_{i_{L+1}=1}^{I_{L+1}} \cdots \sum_{i_N=1}^{I_N} \mathcal{Q}_{i_1,\ldots,i_K,\ldots,i_L,\ldots,i_N,i_1,\ldots,i_K,\ldots,i_L,\ldots,i_N} \leq P_{i_K,i_L}, \ \forall (i_K, i_L). \tag{3.20}$$

Similarly, we can represent constraints on any number of domains. Lastly, let us assume that $\underline{i}_c$ is empty, hence $\underline{i}_r = \underline{i}$ and the power constraint translates to the sum power constraint, i.e. $P_{\underline{i}_c} = P$ and we get :

$$\sum_{i_1=1}^{I_1} \cdots \sum_{i_N=1}^{I_N} \mathcal{Q}_{i_1,\ldots,i_N,i_1,\ldots,i_N} = \operatorname{tr}(\mathcal{Q}) \leq P. \tag{3.21}$$

All such power constraints are linear, and the objective function in (3.15) is concave. Note that the feasible set for this optimization problem is the set of positive semi-definite tensors satisfying the given power constraints which are linear. Hence the feasible set is convex. Thereby (3.15), (3.16) and (3.17) represent a family of convex optimization problems which can be solved using the KKT conditions [150]. Furthermore, since $P_{\underline{i}_c}$ are finite and non-negative, an obvious choice of the covariance tensor belonging to the feasible set could be a pseudo-diagonal tensor with non-negative entries such that they satisfy the power constraints. So the feasible set is a non-empty convex set. Hence by Slater's condition [150], strong duality holds and the optimal solution always exist. Next we will find the optimal solution using the KKT conditions. A description of the KKT conditions for tensors can be found in Appendix A.1.

3.2.2 Solution using KKT conditions

Let $\mathcal{M} \succeq 0$ be the Lagrange multiplier tensor for the positive semi-definite constraint from (3.17) of size $I_1 \times \ldots I_N \times I_1 \times \ldots I_N$. Let $\mu_{\underline{i}_c} \geq 0$ for all $\underline{i}_c$ denote the Lagrange multipliers corresponding to all the linear constraints from (3.16). Then the Lagrangian functional can be defined as :

$$\mathcal{L}(\mathcal{Q}, \{\mu_{\underline{i}_c}\}, \mathcal{M}) = -\log[\det(\mathcal{H} *_N \mathcal{Q} *_N \mathcal{H}^H + \mathcal{I}_M)] + \sum_{\underline{i}_c} \mu_{\underline{i}_c} \Big(\sum_{\underline{i}_r} \mathcal{Q}_{\underline{i},\underline{i}} - P_{\underline{i}_c}\Big) - \operatorname{tr}(\mathcal{M} *_N \mathcal{Q}). \tag{3.22}$$

For any optimization problem, the Lagrangian functional is a function of the objective function, and the constraint functions along with their respective Lagrange multipliers. For a detailed explanation regarding the Lagrangian, refer to Appendix A.1. We arrange the values $\{\mu_{\underline{i}_c}\}$ in a pseudo-diagonal tensor $\mathcal{B}$ of same size as the input covariance such that its non-zero entries are $\mathcal{B}_{\underline{i},\underline{i}} = \mu_{\underline{i}_c}, \forall \underline{i}_r$. For instance, if $\underline{i}_c = (i_1, i_2)$, then $\mathcal{B}_{i_1,\ldots,i_N,i_1,\ldots,i_N} = \mu_{i_1,i_2}$ for all $(i_3, \ldots, i_N)$. Then we get

$$\sum_{\underline{i}_c} \mu_{\underline{i}_c} \cdot \sum_{\underline{i}_r} \mathcal{Q}_{\underline{i},\underline{i}} = \sum_{\underline{i}_c} \sum_{\underline{i}_r} \mu_{\underline{i}_c} \cdot \mathcal{Q}_{\underline{i},\underline{i}} \tag{3.23}$$

$$= \sum_{\underline{i}} \mathcal{B}_{\underline{i},\underline{i}} \cdot \mathcal{Q}_{\underline{i},\underline{i}} \tag{3.24}$$

$$= \mathrm{tr}(\mathcal{B} *_N \mathcal{Q}). \tag{3.25}$$

Based on (3.25), we can re-write the Lagrangian from (3.22) as :

$$\mathcal{L}(\mathcal{Q}, \{\mu_{\underline{i}_c}\}, \mathcal{M}) = -\log[\det(\mathcal{H} *_N \mathcal{Q} *_N \mathcal{H}^H + \mathcal{I}_M)] - \sum_{\underline{i}_c} \mu_{\underline{i}_c} P_{\underline{i}_c} + \mathrm{tr}(\mathcal{B} *_N \mathcal{Q}) - \mathrm{tr}(\mathcal{M} *_N \mathcal{Q}). \tag{3.26}$$

The first KKT condition is obtained by setting the gradient of the Lagrangian with respect to $\mathcal{Q}$ to $0_{\mathcal{T}}$. In (3.26), the gradient of the $\log[\det(\cdot)]$ term can be found using (2.27), and the gradient of the trace terms can be found using (2.28). Thus, the gradient of the Lagrangian with respect to $\mathcal{Q}$ can be written as :

$$\nabla_{\mathcal{Q}} \mathcal{L} = -\mathcal{H}^H *_M (\mathcal{H} *_N \mathcal{Q} *_N \mathcal{H}^H + \mathcal{I}_M)^{-1} *_M \mathcal{H} + \mathcal{B} - \mathcal{M}. \tag{3.27}$$

Equating $\nabla_{\mathcal{Q}} \mathcal{L}$ from (3.27) to $0_{\mathcal{T}}$, we get the first KKT condition as

$$\mathcal{H}^H *_M (\mathcal{H} *_N \mathcal{Q} *_N \mathcal{H}^H + \mathcal{I}_M)^{-1} *_M \mathcal{H} = \mathcal{B} - \mathcal{M}. \tag{3.28}$$

The KKT equations also include complementary slackness condition corresponding to each constraint and its associated Lagrange multiplier [150]. For the linear constraints in (3.16), the definition of complementary slackness leads to :

$$\mu_{\underline{i}_c}(\sum_{\underline{i}_r} \mathcal{Q}_{\underline{i},\underline{i}} - P_{\underline{i}_c}) = 0, \quad \forall \underline{i}_c. \tag{3.29}$$

For the constraint in (3.17), based on the approach taken for semi-definite programming [150], the complementary slackness can be written as $\mathrm{tr}(\mathcal{M} *_N \mathcal{Q}) = 0$. Since $\mathcal{M}, \mathcal{Q} \succeq 0$, we have $\mathrm{tr}(\mathcal{M} *_N \mathcal{Q}) = 0 \Rightarrow \mathcal{M} *_N \mathcal{X} = 0_{\mathcal{T}}$ (See Lemma 9 from Appendix A.1). Also, since $\mathrm{tr}(\mathcal{M} *_N \mathcal{Q}) = \mathrm{tr}(\mathcal{M} *_N \mathcal{Q}^{1/2} *_N \mathcal{Q}^{1/2}) = \mathrm{tr}(\mathcal{Q}^{1/2} *_N \mathcal{M} *_N \mathcal{Q}^{1/2})$, the complementary slackness for the positive semi-definite constraint is written as :

$$\mathcal{Q}^{1/2} *_N \mathcal{M} *_N \mathcal{Q}^{1/2} = 0_{\mathcal{T}}. \tag{3.30}$$

The tensor KKT conditions for the problem in (3.15)-(3.17) are given by (3.28), (3.29) and (3.30).

Notice that all the entries of $\mathcal{B}$, i.e $\mu_{\underline{i}_c}$, will be strictly greater than 0 at optimum because the inequality constraint must be met with equality at optimum. So $\mathcal{B}$ is a positive definite tensor, i.e. $\mathcal{B} \succ 0$ and hence invertible. Also since $\mu_{\underline{i}_c} > 0$, (3.29) can be written as :

$$\sum_{\underline{i}_r} \mathcal{Q}_{\underline{i},\underline{i}} - P_{\underline{i}_c} = 0, \quad \forall \underline{i}_c. \tag{3.31}$$

Let us define a tensor $\bar{\mathcal{H}} \in \mathbb{C}^{I_1 \times \ldots \times I_N \times I_1 \times \ldots \times I_N}$ and its tensor EVD as :

$$\bar{\mathcal{H}} \triangleq \mathcal{B}^{-1/2} *_N (\mathcal{H}^H *_M \mathcal{H}) *_N \mathcal{B}^{-1/2} = \mathcal{V} *_N \bar{\mathcal{D}} *_N \mathcal{V}^H. \tag{3.32}$$

Theorem 1 from [151] presents a general framework in a matrix setting to solve equations originating from KKT conditions for MIMO input covariance optimization. By extending it to tensor case, we can solve (3.28) and (3.30) subject to $\mathcal{Q} \succeq 0, \mathcal{M} \succeq 0$ and $\mathcal{B} \succ 0$ and obtain the optimal $\mathcal{Q}$. A detailed solution of the KKT equations in the tensor framework has been included in Appendix A.2. Based on the results from Appendix A.2, the optimal $\mathcal{Q}$ is given by

$$\mathcal{Q}_{opt} = \mathcal{B}^{-1/2} *_N \mathcal{V} *_N \left(\mathcal{I}_N - \bar{\mathcal{D}}^{-1}\right)^+ *_N \mathcal{V}^H *_N \mathcal{B}^{-1/2} \tag{3.33}$$

where $\bar{\mathcal{D}}$ and $\mathcal{V}$ are obtained through the tensor EVD of $\bar{\mathcal{H}}$ from (3.32) and $(\mathcal{Z})^+$ denotes a pseudo-diagonal tensor whose pseudo-diagonal entries are all non-negative, i.e.

$(\mathcal{Z}_{i_1,\dots,i_N})^+ = \max\{0, \mathcal{Z}_{i_1,\dots,i_N}\}$. Hence we can calculate the capacity as :

$$\begin{aligned}
C &= \log\left[\det\left(\mathcal{H} *_N \mathcal{Q}_{opt} *_N \mathcal{H}^H + \mathcal{I}_M\right)\right] \\
&= \log\left[\det\left(\mathcal{Q}_{opt} *_N \mathcal{H}^H *_M \mathcal{H} + \mathcal{I}_N\right)\right] \\
&= \log\left[\det(\mathcal{B}^{-1/2} *_N \mathcal{V} *_N (\mathcal{I}_N - \bar{\mathcal{D}}^{-1})^+ *_N \mathcal{V}^H *_N \mathcal{B}^{-1/2} *_N \mathcal{H}^H *_M \mathcal{H} + \mathcal{I}_N)\right] \\
&= \log\left[\det(\mathcal{V} *_N (\mathcal{I}_N - \bar{\mathcal{D}}^{-1})^+ *_N \mathcal{V}^H *_N \right. \\
&\quad \left. \underbrace{\mathcal{B}^{-1/2} *_N \mathcal{H}^H *_M \mathcal{H} *_N \mathcal{B}^{-1/2}}_{\bar{\mathcal{H}}} + \mathcal{I}_N)\right] \qquad \text{(using (2.20), (3.32))} \\
&= \log\left[\det(\mathcal{V} *_N ((\mathcal{I}_N - \bar{\mathcal{D}}^{-1})^+ *_N \bar{\mathcal{D}} + \mathcal{I}_N) *_N \mathcal{V}^H)\right].
\end{aligned} \tag{3.34}$$

Since the determinant is the product of eigenvalues, we get :

$$\begin{aligned}
C &= \sum_{i_1,\dots,i_N} \log((1 - \bar{d}^{-1}_{i_1,\dots,i_N})^+ \cdot \bar{d}_{i_1,\dots,i_N} + 1) \\
&= \sum_{i_1,\dots,i_N} (\log(\bar{d}_{i_1,\dots,i_N}))^+
\end{aligned} \tag{3.35}$$

where $\bar{d}_{i_1,\dots,i_N}$ are the non-zero eigenvalues of $\bar{\mathcal{H}}$. Note that the optimum covariance tensor from (3.33) depends not only on the eigenvalues, but also the eigentensors of $\bar{\mathcal{H}}$. Hence ensuring an optimum input covariance leads to not only an optimum power allocation scheme, but also a joint multi-domain precoding at the transmitter which is required for achieving capacity.

In order to find the optimal input covariance, we need to find the tensor $\mathcal{B}$ containing the Lagrange multipliers. We will now simplify the expression for covariance under high SNR assumption, and find the elements of $\mathcal{B}$. Further, we will develop an algorithm to approximate the optimum covariance and capacity using (3.31), (3.33) and (3.35) for any SNR value.

Assuming high SNR, we can ignore $()^+$ and write (3.33) as:

$$\begin{aligned}
\mathcal{Q} &= \mathcal{B}^{-1/2} *_N \left(\mathcal{I}_N - \underbrace{\mathcal{V} *_N \bar{\mathcal{D}}^{-1} *_N \mathcal{V}^H}_{\bar{\mathcal{H}}^{-1}}\right) *_N \mathcal{B}^{-1/2} \\
&= \mathcal{B}^{-1/2} *_N (\mathcal{I}_N - \mathcal{B}^{1/2} *_N (\mathcal{H}^H *_M \mathcal{H})^{-1} *_N \mathcal{B}^{1/2}) *_N \mathcal{B}^{-1/2} \quad \text{(using (3.32))} \\
&= \mathcal{B}^{-1} - (\mathcal{H}^H *_M \mathcal{H})^{-1}
\end{aligned} \tag{3.36}$$

and (3.34) as :

$$
\begin{aligned}
C &= \log\left[\det(\mathcal{V} *_N ((\mathcal{I}_N - \bar{\mathcal{D}}^{-1}) *_N \bar{\mathcal{D}} + \mathcal{I}_N) *_N \mathcal{V}^H)\right] \\
&= \log\left[\det(\mathcal{V} *_N \bar{\mathcal{D}} *_N \mathcal{V}^H)\right] = \log\left[\det(\bar{\mathcal{H}})\right] \\
&= \log\left[\det(\mathcal{B}^{-1/2} *_N (\mathcal{H}^H *_M \mathcal{H}) *_N \mathcal{B}^{-1/2})\right] \\
&= \log\left[\det(\mathcal{B}^{-1} *_N (\mathcal{H}^H *_M \mathcal{H}))\right] \quad \text{(using (2.21))} \\
&= \log\left[\det(\mathcal{B}^{-1}) \cdot \det(\mathcal{H}^H *_M \mathcal{H})\right].
\end{aligned} \tag{3.37}
$$

It is important to note that the channel capacity and optimum input covariance can be exactly calculated using (3.37) and (3.36) only at high SNR. Under high SNR approximation, we can find the elements of $\mathcal{B}$ by substituting $\mathcal{Q}$ from (3.36) into (3.31) to get :

$$
\sum_{\underline{i}_r} (\mathcal{B}^{-1} - (\mathcal{H}^H *_M \mathcal{H})^{-1})_{\underline{i},\underline{i}} = P_{\underline{i}_c} \quad \forall \underline{i}_c \tag{3.38}
$$

$$
\sum_{\underline{i}_r} (\mathcal{B}^{-1})_{\underline{i},\underline{i}} - \sum_{\underline{i}_r} ((\mathcal{H}^H *_M \mathcal{H})^{-1})_{\underline{i},\underline{i}} = P_{\underline{i}_c} \quad \forall \underline{i}_c. \tag{3.39}
$$

Let $N_{\underline{i}_r}$ denote the number of values that $\underline{i}_r$ can take. For example, if $\underline{i}_r = (i_1, \ldots, i_N)$, then $N_{\underline{i}_r} = I_1 \cdots I_N$, if $\underline{i}_r = (i_K, i_L)$, then $N_{\underline{i}_r} = I_K \cdot I_L$. Since $\mathcal{B}$ contains $\mu_{\underline{i}_c}$ on its pseudo-diagonal with each $\mu_{\underline{i}_c}$ appearing exactly $N_{\underline{i}_r}$ times, from (3.39) we can write :

$$
N_{\underline{i}_r} \cdot \mu_{\underline{i}_c}^{-1} - \sum_{\underline{i}_r} ((\mathcal{H}^H *_M \mathcal{H})^{-1})_{\underline{i},\underline{i}} = P_{\underline{i}_c} \quad \forall \underline{i}_c \tag{3.40}
$$

$$
\mu_{\underline{i}_c} = \frac{N_{\underline{i}_r}}{P_{\underline{i}_c} + \sum_{\underline{i}_r} ((\mathcal{H}^H *_M \mathcal{H})^{-1})_{\underline{i},\underline{i}}} \qquad \forall \underline{i}_c \tag{3.41}
$$

which gives us all the elements of $\mathcal{B}$.

The proposed solution in (3.41), (3.37) and (3.36) assumes high SNR as we have ignored the $()^+$ operation. To extend the solution for any SNR, we now present a scaling approach to approximate the input covariance tensor at any SNR setting, and verify that the resulting covariance satisfies the constraints. If the covariance $\mathcal{Q}$ obtained from (3.36) has negative eigenvalues, then we force the negative eigenvalues of $\mathcal{Q}$ to be zero. If the tensor EVD of $\mathcal{Q}$ is given as $\mathcal{U} *_N \mathcal{D} *_N \mathcal{U}^H$, the pseudo-diagonal elements of $\mathcal{Q}$ can be written as, $\mathcal{Q}_{\underline{i},\underline{i}} = \sum_{\underline{i}'} \mathcal{U}_{\underline{i},\underline{i}'} \mathcal{D}_{\underline{i}',\underline{i}'} \mathcal{U}^H_{\underline{i}',\underline{i}}$. Thus the brute force approach of setting the negative values in $\mathcal{D}$ to zero can result into larger values at the pseudo-diagonal of $\mathcal{Q}$. This in turn can make the solution infeasible, i.e. the power constraint $P_{\underline{i}_c}$ from (3.16) might not be met and we may get a different power allotted say $P'_{\underline{i}_c}$ where $P'_{\underline{i}_c} \geq P_{\underline{i}_c}$. Note that

$$P'_{\underline{i}_c} = \sum_{\underline{i}_r} \mathcal{Q}_{\underline{i},\underline{i}}, \quad \forall \underline{i}_c. \tag{3.42}$$

So we scale the resulting $\mathcal{Q}$ using another pseudo-diagonal tensor $\mathcal{S}$ such that the power constraints remain satisfied, i.e. $\mathcal{Q}_{scaled} = \mathcal{S} *_N \mathcal{Q} *_N \mathcal{S}^H$ where the pseudo-diagonal entries of scaling tensor $\mathcal{S}$ are given as :

$$\mathcal{S}_{\underline{i},\underline{i}} = \sqrt{\frac{P_{\underline{i}_c}}{P'_{\underline{i}_c}}}, \quad \forall \underline{i}_r \tag{3.43}$$

such that the pseudo-diagonal entries of $\mathcal{Q}_{scaled}$ become :

$$(\mathcal{Q}_{scaled})_{\underline{i},\underline{i}} = \mathcal{Q}_{\underline{i},\underline{i}} \cdot \frac{P_{\underline{i}_c}}{P'_{\underline{i}_c}}. \tag{3.44}$$

Hence, based on (3.42) we have

$$\sum_{\underline{i}_r} (\mathcal{Q}_{scaled})_{\underline{i},\underline{i}} = \sum_{\underline{i}_r} \mathcal{Q}_{\underline{i},\underline{i}} \cdot \frac{P_{\underline{i}_c}}{P'_{\underline{i}_c}} = P_{\underline{i}_c}. \tag{3.45}$$

Thus the choice of scaling operation ensures that $\mathcal{Q}_{scaled}$ satisfies the power constraints. A similar technique has been used for matrix-field water-filling in [102] where the diagonal elements of the covariance matrix are scaled to ensure that the resulting matrix remains positive semi-definite while satisfying the power constraints. Such a scaling approach simplifies the computation of the input covariance but also makes it suboptimal and hence leads to an approximation of the capacity. However, this approximation gets better as SNR grows and is exact at sufficiently high SNR. This is because a sufficiently high SNR ensures that all the eigenvalues of the input covariance are non-negative, and thus the covariance tensor is positive semi-definite without any requirement of scaling. The procedure is systematically presented in Algorithm 1. Finding the capacity is a convex optimization problem, hence can be solved using software tools such as CVX [152], which can be compared with the capacity obtained via the approximation in Algorithm 1 to assess the validity of the proposed approach. We present such a comparison in the numerical examples later on in this chapter (Figure 3.7) to illustrate the accuracy of the proposed approach.

Algorithm 1 Finding the Input Covariance tensor

Input $\mathcal{H}, P_{\underline{i}_c}, N_{\underline{i}_r}$
Initialize $flag \leftarrow 0$
Find pseudo-diagonal elements of $\mathcal{B}$ by calculating $\mu_{\underline{i}_c}$ using (3.41)
Calculate $\mathcal{Q}$ using (3.36).
Perform tensor EVD of $\mathcal{Q} = \mathcal{U} *_N \mathcal{D} *_N \mathcal{U}^H$
for all $i_1, i_2, \ldots, i_N$
    if $\mathcal{D}_{i_1,\ldots,i_N} < 0$
        $\mathcal{D}_{i_1,\ldots,i_N} \leftarrow 0$
        $flag \leftarrow 1$
    end if
end for
if $flag == 1$
    Update $\mathcal{Q} \leftarrow \mathcal{U} *_N \mathcal{D} *_N \mathcal{U}^H$
    Calculate $P'_{\underline{i}_c}$ using (3.42)
    Find pseudo-diagonal tensor $\mathcal{S}$ using (3.43)
    Update $\mathcal{Q} \leftarrow \mathcal{S} *_N \mathcal{Q} *_N \mathcal{S}^H$
end if
return $\mathcal{Q}$

3.2.3 Complexity Analysis of Algorithm 1

Algorithm 1 is used to approximate the optimal input covariance tensor using (3.36) and a scaling process to ensure a feasible solution. For a given channel, the algorithm requires fixed computational resources and can be deployed off line. In this section, we analyze the computational resources required to execute this algorithm. Since Algorithm 1 requires extensive tensor operations which scale with the tensor size, using cloud services to implement it can be a suitable option. Several cloud services provide parallel and distributed computing infrastructures for faster and efficient computations [153]. Thus depending on the platform and the number of multi-core processors being employed, the time of execution of the algorithm can significantly differ. However, irrespective of the computing infrastructure available, a suitable measure of the computational complexity is the required number of mathematical operations to be performed in a given algorithm, as used in [154, 155]. Hence, here we analyze the computational complexity of Algorithm 1 in terms of the required number of flops for a given step as discussed in section 2.1.7.

Assume channel $\mathcal{H}$ is of size $J_1 \times \ldots \times J_M \times I_1 \times \ldots \times I_N$ and $N_{\underline{i}_c}$ denotes the number of values that $\underline{i}_c$ can take. In Algorithm 1, the first two steps are input initialization. Step 3 requires computing the Einstein product over the common M modes of $\mathcal{H}^H$ and $\mathcal{H}$, which based on the discussion in section 2.1.7 has a complexity of $\mathcal{O}((I_1 \cdots I_N)^2(J_1 \cdots J_M))$. Further within step 3, it is required to find the inverse of $\mathcal{H}^H *_M \mathcal{H}$ which is an order $2N$ tensor. The inverse of an order $2N$ tensor can be calculated using the HOBG method described in [48] or NM where each iteration has a computational cost of $\mathcal{O}((I_1 \cdots I_N)^3)$ as discussed in section 2.1.7. It is important to note that the complexity of such iterative methods depends on the number of iterations, which in turn depends on the desired accuracy level set to achieve convergence. Furthermore, parallel processing can be employed to reduce the time complexity of such operations as discussed in Appendix B.7. Also, non-iterative methods such as Gauss elimination based on triangular decomposition of tensors [70] can be used for tensor inversion which requires a computational complexity of $\mathcal{O}((I_1 \cdots I_N)^3)$. Hence, the worst case complexity of tensor inversion without any use of parallel processors is $\mathcal{O}((I_1 \cdots I_N)^3)$. Eventually step 3 calculates each $\mu_{\underline{i}_c}$ using (3.41) which requires $N_{\underline{i}_r} + 1$ additions and 1 division, and this needs to be done for all the $N_{\underline{i}_c}$ values that $\underline{i}_c$ can take. Thus this step requires $(N_{\underline{i}_r} + 2) \cdot N_{\underline{i}_c}$ flops and its complexity is $\mathcal{O}(N_{\underline{i}_r} \cdot N_{\underline{i}_c})$. Note that since $N_{\underline{i}_r} \cdot N_{\underline{i}_c} = I_1 \cdots I_N$, the complexity of finding $\mu_{\underline{i}_c}$ can be written as $\mathcal{O}(I_1 \cdots I_N)$. Step 4 which finds $\mathcal{Q}$ using (3.36) subtracts an order $2N$ tensor from a pseudo-diagonal tensor. Since the number of pseudo-diagonal elements are $I_1 \cdots I_N$, step 4 performs $I_1 \cdots I_N$ subtractions and thus has a complexity of $\mathcal{O}(I_1 \cdots I_N)$. Further step 5 finds the tensor EVD of an order $2N$ tensor $\mathcal{Q}$. The complexity of finding the EVD of a tensor of size $I_1 \times \ldots \times I_N \times I_1 \times \ldots \times I_N$ using the Einstein product properties is $\mathcal{O}((I_1 \cdots I_N)^3)$ [46]. Algorithms which generalize the matrix eigen decomposition approaches to tensor EVD using the Einstein product properties can be found in [70, 69], [46, Algorithm C.2] . Steps 6 to 11 essentially perform the operation $\max(0, \mathcal{D}_{i_1,\ldots,i_N})$ on each of the $I_1 \cdots I_N$ pseudo-diagonal elements of the tensor $\mathcal{D}$. Hence it has a complexity of $\mathcal{O}(I_1 \cdots I_N)$. Step 12 is just a single scalar comparison, and step 13 updates $\mathcal{Q}$ using the Einstein product between tensors of order $2N$ with size $I_1 \times \ldots \times I_N \times I_1 \times \ldots \times I_N$ for which the complexity is $\mathcal{O}((I_1 \cdots I_N)^3)$.

Step 14 finds $P'_{\underline{i}_c}$ for all the $N_{\underline{i}_c}$ values of $\underline{i}_c$. Thus it performs $N_{\underline{i}_r}$ additions for each of the $N_{\underline{i}_c}$ values that $\underline{i}_c$ can take. Hence step 14 requires $N_{\underline{i}_r} \cdot N_{\underline{i}_r}$ flops and thus has a complexity of $\mathcal{O}(N_{\underline{i}_r} \cdot N_{\underline{i}_r})$ which is same as $\mathcal{O}(I_1 \cdots I_N)$. Step 15 calculates the scaling factor for all $\underline{i}_c$, thus performs $N_{\underline{i}_c}$ divisions and square roots. Hence it has a complexity of $\mathcal{O}(N_{\underline{i}_c})$. Finally, step 16 updates $\mathcal{Q}$ using the Einstein product between tensors of order $2N$ and thus has a complexity of $\mathcal{O}((I_1 \cdots I_N)^3)$.

Table 3.2: Computational complexity of Algorithm 1.

Step	Operation	Complexity
3	Find $\mathcal{H}^H *_M \mathcal{H}$	$\mathcal{O}((I_1 \cdots I_N)^2 \cdot J_1 \cdots J_M)$
3	Find $(\mathcal{H}^H *_M \mathcal{H})^{-1}$	$\mathcal{O}((I_1 \cdots I_N)^3)$
3	Find $\mu_{\underline{i}_c}$ using (3.41)	$\mathcal{O}(I_1 \cdots I_N)$
4	Find $\mathcal{Q}$ using (3.36)	$\mathcal{O}(I_1 \cdots I_N)$
5	EVD of $\mathcal{Q}$	$\mathcal{O}((I_1 \cdots I_N)^3)$
6-11	Check $\mathcal{D}$	$\mathcal{O}(I_1 \cdots I_N)$
13	Update $\mathcal{Q}$	$\mathcal{O}((I_1 \cdots I_N)^3)$
14	Find $P'_{\underline{i}_c}$ using (3.42)	$\mathcal{O}(I_1 \cdots I_N)$
15	Find $\mathcal{S}$ using (3.43)	$\mathcal{O}(N_{\underline{i}_c})$
16	Update $\mathcal{Q}$	$\mathcal{O}((I_1 \cdots I_N)^3)$

Table 3.2 summarizes the step by step computational complexity cost of Algorithm 1. The first column in Table 3.2 indicates the step number from Algorithm 1, second column describes the operation and third column states the complexity. We can observe that all the entries in complexity column of Table 3.2, have a complexity order of 3 (cubic) or less in $I_1 \cdots I_N$ except the first operation which has a complexity of $\mathcal{O}((I_1 \cdots I_N)^2 \cdot J_1 \cdots J_M)$. Hence on summing all the entries of the third column in Table 3.2, we see that the overall complexity of Algorithm 1 is given as $\mathcal{O}((I_1 \cdots I_N)^2 \cdot J_1 \cdots J_M) + \mathcal{O}((I_1 \cdots I_N)^3)$. Further in the case when $J_1 \cdots J_M \leq I_1 \cdots I_N$, the complexity of Algorithm 1 can be written as $\mathcal{O}((I_1 \cdots I_N)^3)$.

Note that the steps in Algorithm 1 with complexity of $\mathcal{O}((I_1 \cdots I_N)^3)$ primarily rely on the Einstein product operation. However, since this algorithm can be executed on multi-core computer systems, the time complexity of performing all the operations in Einstein product can be significantly reduced by making use of parallel processing. This can help reduce the time complexity of operations such as tensor inversion and EVD. Appendix B.7 includes details on parallel implementation of the Einstein product and the NM. The steps in Algorithm 1 which have a complexity of $\mathcal{O}(I_1 \cdots I_N)$ or less (step 15), can also be performed faster using parallel processors. For example in step 3, all the $\mu_{\underline{i}_c}$ of (3.41) for different $\underline{i}_c$ can be calculated simultaneously on parallel processors. Similarly all the $I_1 \cdots I_N$ operations in steps 4 and 6-11, can be performed simultaneously. In steps 14 and 15, the $P'_{\underline{i}_c}$ from (3.42) and the scaling factors from (3.43) for all the $\underline{i}_c$ can also be computed simultaneously. Hence Algorithm 1 can be suitably adapted to run on parallel processing multi-core computer systems depending on the number of processors available. Several computing platforms such as MATLAB provide support for parallel implementation of such algorithms. A more detailed study into the parallelization of the proposed algorithm for faster time complexity has been left for future investigation.

3.2.4 Comparing different constraints

Let the set of all possible positive semi-definite tensors of size $I_1 \times \ldots \times I_N \times I_1 \times \ldots \times I_N$ be represented by $\mathbb{Q}$. Let the feasible set for the optimization problem in (3.15)-(3.17) for two different settings $\underline{i}_{c1}$ and $\underline{i}_{c2}$ be $\mathbb{Q}_1$ and $\mathbb{Q}_2$ respectively. Assume $\underline{i}_{c1}$ is a subsequence of $\underline{i}_{c2}$. For instance, let $\underline{i}_{c2} = (i_1, i_2, i_3)$, and $\underline{i}_{c1} = (i_1, i_2)$. So $\mathbb{Q}_1$ represents a set of all positive semi-definite tensors $\mathcal{Q}$ which satisfies

$$\sum_{i_3, i_4, \ldots, i_N} \mathcal{Q}_{i_1,\ldots,i_N,i_1,\ldots,i_N} \leq P_{i_1,i_2} \quad \forall (i_1, i_2) \tag{3.46}$$

and $\mathbb{Q}_2$ represents a set of all positive semi-definite tensors $\mathcal{Q}$ which satisfies

$$\sum_{i_4, \ldots, i_N} \mathcal{Q}_{i_1,\ldots,i_N,i_1,\ldots,i_N} \leq P_{i_1,i_2,i_3} \quad \forall (i_1, i_2, i_3). \tag{3.47}$$

In (3.46), the first two domains are under power constraints, whereas in (3.47), the power constraints span the third domain as well with $\sum_{i_3} P_{i_1,i_2,i_3} = P_{i_1,i_2}$. Summing over i_3 in (3.47) gives (3.46). Hence every $\mathcal{Q}$ that satisfies (3.47) will also satisfy (3.46), showing that the set of $\mathcal{Q}$ satisfying (3.47) is a subset of the set of $\mathcal{Q}$ satisfying (3.46), i.e. $\mathbb{Q}_2 \subseteq \mathbb{Q}_1$.

Let the optimal value of the objective function in the optimization problem in (3.15)-(3.17) for set $\mathbb{Q}_1$ be C_1 and for $\mathbb{Q}_2$ be C_2. From the basic principles of optimization [150], it is known that a globally optimal point is also locally optimal. Hence if C_1 is the maximum of the objective function over the set of constraints $\mathbb{Q}_1$, then C_1 is also the maximum of the objective function over $\mathbb{Q}_2$ since $\mathbb{Q}_2 \subseteq \mathbb{Q}_1$. Hence $C_2 \leq C_1$, where equality is possible if the optimal Q lies in the set $\mathbb{Q}_2$. This holds for any configuration of $\underline{i}_{c1}$ and $\underline{i}_{c2}$ so far as $\underline{i}_{c1}$ is a subsequence of $\underline{i}_{c2}$. Essentially, as more domains are put under constraints, the feasible set for the optimization problem shrinks, possibly lowering the capacity.

For instance, consider a 2×2 input where the two domains are antenna and time slots. Let the capacity achieved under total power constraint P, be C_1. Let the capacity achieved under per antenna power constraints of P_1 for antenna 1 and P_2 for antenna 2 such that $P_1 + P_2 = P$, be C_2. Since the set of feasible solution with per antenna power constraints is a subset of the set of feasible solution with total power constraint we have $C_2 \leq C_1$. Similarly, capacity achieved under power constraints per element, i.e. $P_{11}, P_{12}, P_{21}, P_{22}$ where $P_{i,j}$ represents power budget on the ith antenna and the jth time slot, such that $P_{11} + P_{12} = P_1, P_{21} + P_{22} = P_2$, be C_3, then $C_3 \leq C_2$. This has also been shown in [101] for a MIMO channel where capacity under per antenna power constraint is always smaller than the capacity under sum power constraint.

3.2.5 Capacity under sum power constraint

Under the sum power constraint of (3.21), $\underline{i}_c$ is empty and hence there is a single Lagrange multiplier μ associated with the constraint (3.21). Hence the tensor $\mathcal{B}$, which contains the Lagrange multipliers on its pseudo-diagonal, will be a scaled identity tensor. Substituting $\mathcal{B} = \mu \mathcal{I}_N$ in (3.32) gives $\bar{\mathcal{H}} = \mu^{-1} \cdot (\mathcal{H}^H *_M \mathcal{H}) = \mathcal{V} *_N (\mu^{-1} \cdot \mathcal{D}) *_N \mathcal{V}^H$, and subsequently (3.33) becomes

$$\mathcal{Q} = \mathcal{V} *_N \left(\mu^{-1} \mathcal{I}_N - \mathcal{D}^{-1}\right)^+ *_N \mathcal{V}^H. \tag{3.48}$$

Substituting $\bar{d}_{i_1,\ldots,i_N} = d_{i_1,\ldots,i_N}/\mu$ in (3.35) gives :

$$C = \sum_{i_1 \ldots, i_N} \log \left[\left(\frac{1}{\mu} - \frac{1}{d_{i_1,\ldots,i_N}} \right)^+ \cdot d_{i_1,\ldots,i_N} + 1 \right] \tag{3.49}$$

$$= \sum_{i_1,\ldots,i_N} \left(\log \left(\frac{d_{i_1,\ldots,i_N}}{\mu} \right) \right)^+ \tag{3.50}$$

where a^+ denotes $\max\{0, a\}$, $d_{i_1,\ldots,i_N}$ are the non-zero eigenvalues of $\mathcal{H}^H *_M \mathcal{H}$ and $1/\mu$ is chosen to satisfy (3.31):

$$\mathrm{tr}(\mathcal{Q}) = \sum_{i_1,\ldots,i_N} \left(\frac{1}{\mu} - \frac{1}{d_{i_1,\ldots,i_N}}\right)^+ = P. \tag{3.51}$$

The optimum covariance derived in (3.48) is a generalization of the water-filling solution for the MIMO matrix channel to multiple domains. Hence under sum power constraint, we compute the tensor EVD of $\mathcal{H}^H *_M \mathcal{H}$ to obtain $\mathcal{V}$ and $\mathcal{D}$. Further, we use (3.51) to find μ and subsequently use (3.49) to find the capacity.

3.2.6 Multiplexing Gain

We can characterize the capacity contribution by each constrained domain separately and the multiplexing gain under various power constraints. For a fixed tensor channel, the tensor $\mathcal{B}^{-1}$ is pseudo-diagonal with entries $\mu_{\underline{i}_c}^{-1}$, hence $\det(\mathcal{B}^{-1}) = \prod_{\underline{i}} \mu_{\underline{i}_c}^{-1} = \prod_{\underline{i}_r}(\prod_{\underline{i}_c} \mu_{\underline{i}_c}^{-1}) = \left(\prod_{\underline{i}_c} \mu_{\underline{i}_c}^{-1}\right)^{N_{\underline{i}_r}}$. For instance, assume that out of N input domains, elements of the first domain are under individual power constraints such that $\underline{i}_c = (i_1)$ and $\underline{i}_r = (i_2, \ldots, i_N)$, then

$$\det(\mathcal{B}^{-1}) = \prod_{i_1,i_2,\ldots,i_N} \mu_{i_1}^{-1} = \left(\prod_{i_1} \mu_{i_1}^{-1}\right)^{I_2 \cdot I_3 \cdots I_N}. \tag{3.52}$$

Also $\det(\mathcal{H}^H *_M \mathcal{H}) = \prod_{\underline{i}} d_{\underline{i}}$ where $d_{\underline{i}}$ are eigenvalues of $\mathcal{H}^H *_M \mathcal{H}$. Hence using (3.37) we have

$$\begin{aligned} C &= \log\left[\prod_{\underline{i}} \frac{d_{\underline{i}}}{\mu_{i_1}}\right] \\ &= \sum_{\underline{i}} \log \frac{d_{\underline{i}}}{\mu_{i_1}} \\ &= \sum_{i_1}\Bigg(\sum_{i_2,\ldots,i_N} \log \frac{d_{\underline{i}}}{\mu_{i_1}}\Bigg) = \sum_{i_1} C_{i_1} \end{aligned} \tag{3.53}$$

where $C_{i_1} = \sum_{i_2,\ldots,i_N} \log \dfrac{d_{\underline{i}}}{\mu_{i_1}}$ can be seen as the contribution of the i_1th element of the constrained domain to the overall capacity. For instance if the first domain refers to space domain, then C_{i_1} is the capacity contribution of the i_1th antenna. For any general case where $\underline{i}_c$ contains the indices of domains under constraint, we can write :

$$C = \log \Big[\prod_{\underline{i}_c, \underline{i}_r} \frac{d_{\underline{i}}}{\mu_{\underline{i}_c}} \Big] = \sum_{\underline{i}_c} \Big(\sum_{\underline{i}_r} \log \frac{d_{\underline{i}}}{\mu_{\underline{i}_c}} \Big) = \sum_{\underline{i}_c} C_{\underline{i}_c} \tag{3.54}$$

where $C_{\underline{i}_c} = \sum_{\underline{i}_r} \log \dfrac{d_{\underline{i}}}{\mu_{\underline{i}_c}}$ can be seen as the contribution of the $\underline{i}_c$th element of the constrained domains to the overall capacity. Substituting $\mu_{\underline{i}_c}$ from (3.41) into $C_{\underline{i}_c}$, we can further write :

$$C_{\underline{i}_c}(P_{\underline{i}_c}) = \sum_{\underline{i}_r} \log \frac{d_{\underline{i}}}{\mu_{\underline{i}_c}} \tag{3.55}$$

$$= \sum_{\underline{i}_r} \log \Big(d_{\underline{i}} \cdot \frac{P_{\underline{i}_c} + \sum_{\underline{i}_r} ((\mathcal{H}^H *_M \mathcal{H})^{-1})_{\underline{i},\underline{i}}}{N_{\underline{i}_r}} \Big) \tag{3.56}$$

$$= \sum_{\underline{i}_r} \Big[\log \Big(P_{\underline{i}_c} + \sum_{\underline{i}_r} ((\mathcal{H}^H *_M \mathcal{H})^{-1})_{\underline{i},\underline{i}} \Big) + \log \Big(\frac{d_{\underline{i}}}{N_{\underline{i}_r}} \Big) \Big]. \tag{3.57}$$

The multiplexing gain provided by a channel is defined as $\lim\limits_{\mathtt{SNR} \to \infty} \dfrac{C(\mathtt{SNR})}{\log \mathtt{SNR}}$ [156]. We can write the multiplexing gain provided by $\underline{i}_c$th constrained domain as :

$$\chi_{\underline{i}_c} = \lim_{P_{\underline{i}_c} \to \infty} \frac{C_{\underline{i}_c}(P_{\underline{i}_c})}{\log(P_{\underline{i}_c})} \tag{3.58}$$

Since for large $P_{\underline{i}_c}$, we have $\log \Big(P_{\underline{i}_c} + \sum_{\underline{i}_r} ((\mathcal{H}^H *_M \mathcal{H})^{-1})_{\underline{i},\underline{i}} \Big) \approx \log(P_{\underline{i}_c})$, and using (3.58) and (3.57) we get

$$\chi_{\underline{i}_c} = \lim_{P_{\underline{i}_c} \to \infty} \frac{\sum_{\underline{i}_r} \Big[\log \Big(P_{\underline{i}_c} \Big) + \log \Big(\frac{d_{\underline{i}}}{N_{\underline{i}_r}} \Big) \Big]}{\log(P_{\underline{i}_c})} \tag{3.59}$$

$$= \lim_{P_{\underline{i}_c} \to \infty} \frac{N_{\underline{i}_r} \log(P_{\underline{i}_c}) + \sum_{\underline{i}_r} \log \Big(\frac{d_{\underline{i}}}{N_{\underline{i}_r}} \Big)}{\log(P_{\underline{i}_c})} \tag{3.60}$$

$$= N_{\underline{i}_r}. \tag{3.61}$$

In general, since $N_{\underline{i}_r}$ represents the product of dimensions of the domains not under individual constraints, it increases exponentially in the number of unconstrained domains. As a result, for tensor channels the multiplexing gain increases exponentially with the increase in the number of unconstrained domains. Note that in deriving the multiplexing gain, for simplicity we have assumed that all the eigenvalues of $\mathcal{H}^H *_M \mathcal{H}$ are non-zero, which may not always be the case. In general, the capacity is a function of the given channel's specific singular values, some of which may be zero. In that case depending on which and how many singular values are zero, the multiplexing gain will be different and may not increase exponentially with increase in domains.

Based on (3.61), the multiplexing gain achieved under sum power constraint is $I_1 \cdots I_N$. However (3.61) assumes that inverse of $(\mathcal{H}^H *_M \mathcal{H})$ exists which will be the case if all its eigenvalues are non-zero. In case the inverse does not exist, then a minimum norm least square solution can be adopted which aims to find a tensor $\mathcal{T}$ which represents the pseudo-inverse of $(\mathcal{H}^H *_M \mathcal{H})$ such that $||(\mathcal{H}^H *_M \mathcal{H}) *_N \mathcal{T} - \mathcal{I}_N||^2$ is minimized [48]. For this purpose, a higher order bi-conjugate gradient method is described in [48]. In such a case, the multiplexing gain would be lower than $I_1 \cdots I_N$ and would depend on the number of non-zero eigenvalues of $(\mathcal{H}^H *_M \mathcal{H})$. So next we analyze the multiplexing gain, associated with a tensor channel under sum power constraint. Note that the number of non-zero eigenvalues of $(\mathcal{H}^H *_M \mathcal{H})$ will be same as the number of non-zero singular values of $\mathcal{H}$. Let $I_1 \cdot I_2 \cdots I_N = I$ and $J_1 \cdot J_2 \cdots J_M = J$, then the number of non-zero singular values R are less than or equal to I and J, i.e. $R \leq \min\{I, J\}$ where equality is met if all the the singular values of $\mathcal{H}$ are non-zero.

The multiplexing gain, denoted by χ, is calculated at a very high SNR, for which water-filling reduces to uniform power allocation over the non-zero eigen channels, i.e. $\left(\frac{1}{\mu} - \frac{1}{d_{i_1,\ldots,i_N}}\right)^+ \approx \frac{P}{R}$. Hence using (3.49), the capacity becomes

$$C = \sum_{\substack{i_1,\ldots,i_N \\ d_{i_1,\ldots,i_N} \neq 0}} \left(\log\left(1 + \frac{P}{R} d_{i_1,\ldots,i_N}\right)\right). \tag{3.62}$$

As $P \to \infty$, (3.62) simplifies to :

$$C = \sum_{\substack{i_1,\ldots,i_N \\ d_{i_1,\ldots,i_N} \neq 0}} \log\left(\frac{P}{R} d_{i_1,\ldots,i_N}\right) \tag{3.63}$$

$$= \sum_{\substack{i_1,\ldots,i_N \\ d_{i_1,\ldots,i_N} \neq 0}} \left[\log(P) + \log\left(\frac{d_{i_1,\ldots,i_N}}{R}\right)\right] \tag{3.64}$$

$$= R\log(P) + \sum_{\substack{i_1,\ldots,i_N \\ d_{i_1,\ldots,i_N} \neq 0}} \log\left(\frac{d_{i_1,\ldots,i_N}}{R}\right) \tag{3.65}$$

$$\Rightarrow \chi = \lim_{P\to\infty} \frac{C}{\log(P)} = R. \tag{3.66}$$

Assuming all the singular values of the tensor channel are non-zero, we have $R = \min\{I_1 \cdot I_2 \cdots I_N, J_1 \cdot J_2 \cdots J_M\}$. For a conventional MIMO matrix channel, capacity pre-log is known to be less than or equal to the minimum of the number of transmit and receive antennas [90] which is the specific case of the tensor pre-log. For MIMO matrix channel $N = M = 1$, and we have $\chi = \min\{I_1, J_1\}$ where I_1 and J_1 are the number of transmit and receive antennas respectively. For a tensor case, it is interesting to see that assuming equal dimension sizes on each domain i.e. $I_1 = I_2 = \cdots = I_N = J_1 = J_2 \cdots J_M = L$, then the capacity pre-log can be given as :

$$\begin{aligned} \chi &= \min\{I_1 \cdot I_2 \cdots I_N, J_1 \cdot J_2 \cdots J_M\} \\ &= \min\{L^N, L^M\} \\ &= L^{\min\{N,M\}} \end{aligned} \tag{3.67}$$

which is exponential in the number of domains.

3.3 Numerical Examples and Applications

In this section, we present numerical examples to illustrate previous results.

3.3.1 Examples with different input constraints and channel sizes

Our results assume that the channel is deterministic. For the numerical examples, rather than using a specific channel tensor, we generate the channel using the Rayleigh model as in [90, 91, 157]. The channel tensor consists of realizations of i.i.d. circularly symmetric complex Gaussian entries of zero mean and unit variance such that $\mathbb{E}[|\boldsymbol{\mathcal{H}}_{j_1,\ldots,j_M,i_1\ldots,i_N}|^2] = 1$ [91]. These channel realizations are known at the transmitter and receiver.

Let us denote the capacity of a deterministic tensor channel $\mathcal{H}$ as $C(\mathcal{H})$. Assume that we calculate capacities of K such channels denoted by $C(\mathcal{H}^{(k)})$ for $k = 1, \ldots, K$ where the tensor $\mathcal{H}^{(k)}$ consist of realizations of complex Gaussian random variables. The average of K such deterministic channels is given by $\bar{C}_K = \frac{1}{K}\sum_{k=1}^{K} C(\mathcal{H}^{(k)})$. Due to law of large numbers, as $K \to \infty$, we have $\bar{C}_K \to \mathbb{E}[C(\boldsymbol{\mathcal{H}})]$ where $\boldsymbol{\mathcal{H}}$ is a tensor of Gaussian random variables. All the numerical results included in this section present $\bar{C}_K$ for $K = 100$, which can be interpreted as the ergodic capacity of a random tensor channel when its realizations are known at the transmitter and the receiver. The SNR is defined as P/σ^2 as used in [90], where P is the sum power constraint or the total transmit power and the noise tensor contains i.i.d. circularly symmetric complex Gaussian entries with zero mean and variance $\sigma^2 = 1$.

Capacity for different sizes of Channel Tensor

Figure 3.1 presents the channel capacity in bits/channel-use for a fourth order channel tensor under a sum power constraint at SNR of 10 dB. Input and output are order-2 tensors of size $X \times Y$ each, corresponding to a $X \times Y \times X \times Y$ tensor channel. It is seen that increasing X and Y individually leads to an increase in total capacity.

To understand the effect of different types of channel, we also find capacity when the channel is normalized such that the total receive average power is identical to the transmitted power. Such a normalization ensures that the channel gain is unity and has been suggested in [158, 159] for the MIMO matrix case. In tensor channels, such a normalization is achieved when the individual entries of the channel tensor $\mathcal{H} \in \mathbb{C}^{J_1\times\ldots\times J_M\times I_1\times\ldots\times I_N}$ are

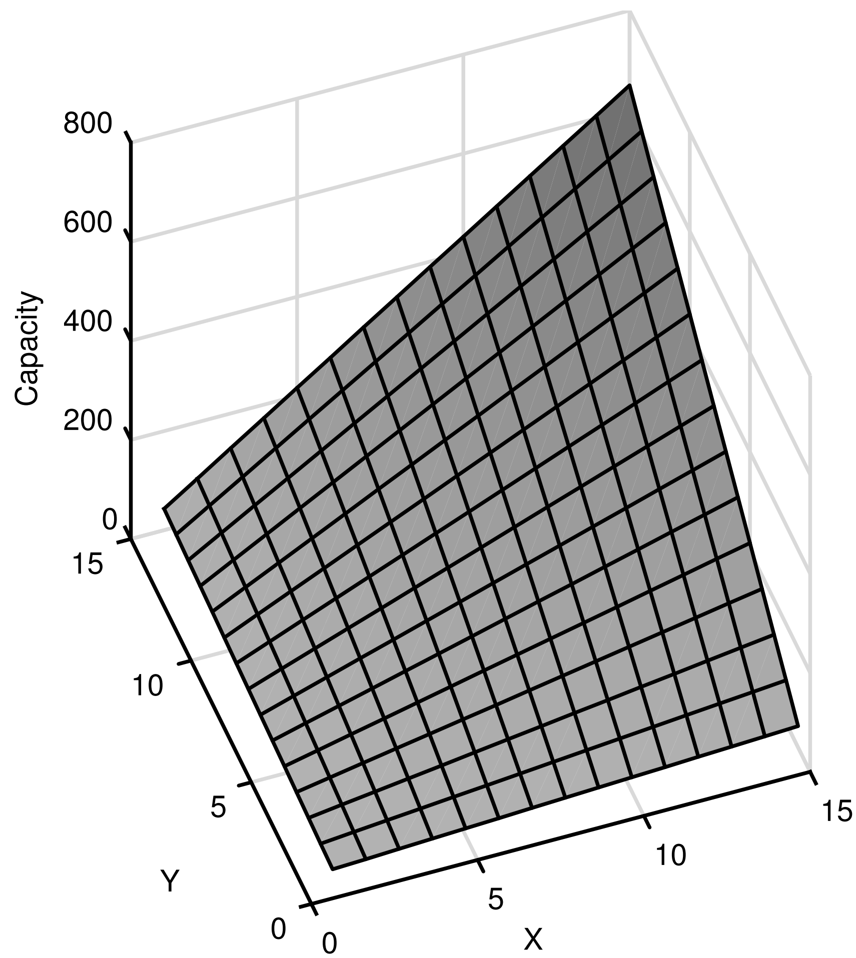


Fig. 3.1: Capacity [bits/channel-use] vs X vs Y for channel with $X \times Y$ size input and output tensor.

generated as circular complex Gaussian with zero mean and variance $1/(J_1 \cdots J_M)$. Figure 3.2 shows the capacity of such a normalized tensor channel of size $X \times Y \times X \times Y$ with both input and output of size $X \times Y$ each, under sum power constraint at a fixed SNR of 10 dB. On increasing the size of input and output tensors, the rate of increase in capacity is lower as compared to Figure 3.1, and the capacity tends to reach a saturation for large values of X and Y.

For the normalized channel case where the channel gain is unity, and hence transmit and receive signal power are same, capacity is plotted against received SNR for a $2 \times 2 \times 2 \times 2$

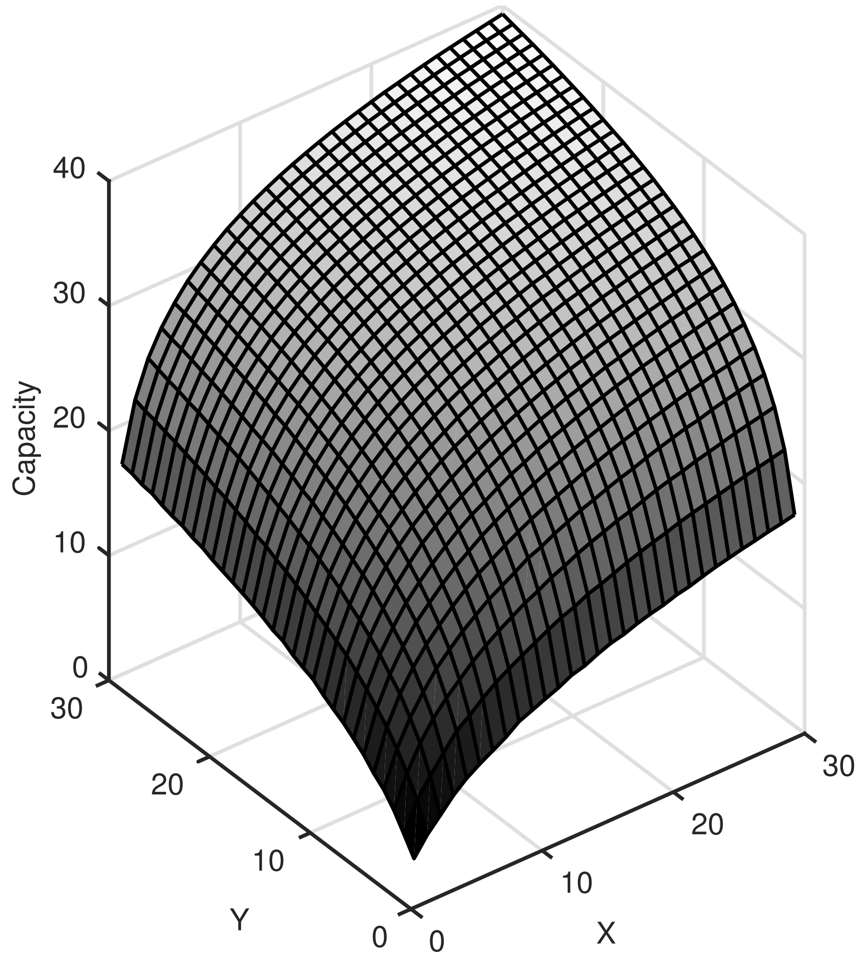


Fig. 3.2: Capacity [bits/channel-use] for normalized channel vs X vs Y where $X \times Y$ is the size of input and output tensor.

tensor channel in Figure 3.3. A comparison with corresponding scalar and matrix channels with the same received signal power is also presented. The gain in the capacity achieved by moving from scalar to a tensor channel can be attributed to the multiplexing gain provided by the tensor channel which increases with the number of domains.

For the rest of the numerical examples, the more widely used model from [90, 91] where the channel entries are circular complex Gaussian with zero mean and unit variance is employed, as for Figure 3.1. Figure 3.4 presents the capacity of a fourth order tensor channel under sum power constraint where the output is fixed as a 2×2 tensor and input

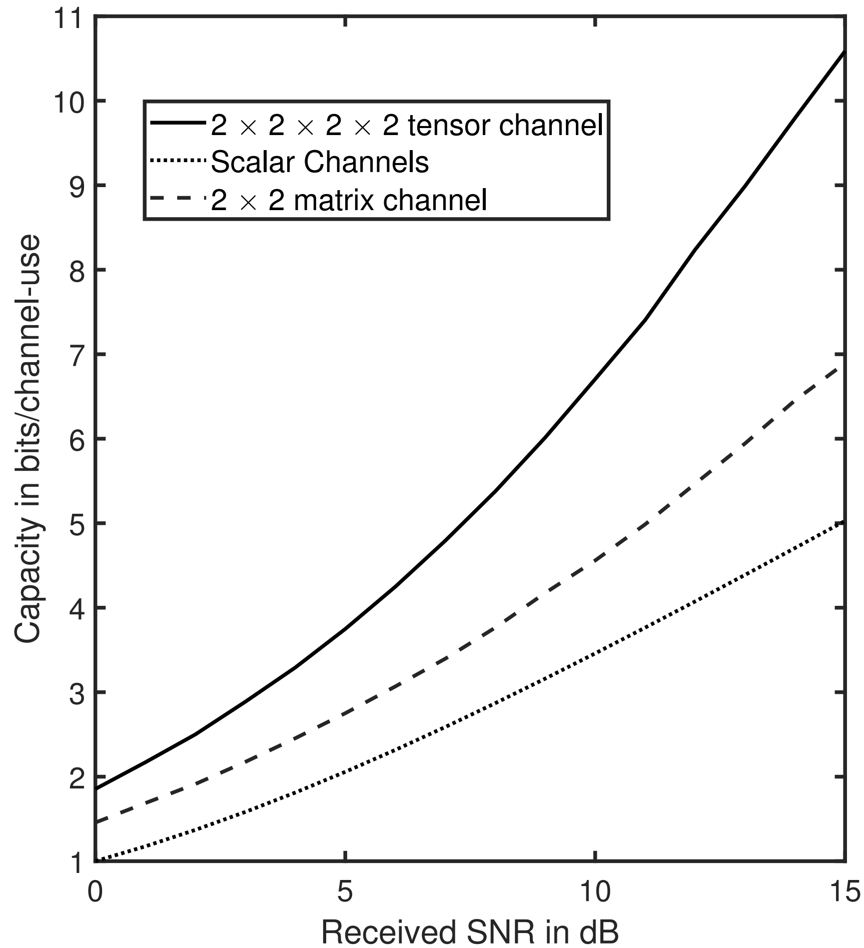


Fig. 3.3: Channel capacity for tensor, matrix and scalar channel with same received signal power.

is $X \times Y$ tensor at 10 dB SNR, under sum power constraint. The rate of increase in capacity for X and Y is lower as compared to Figure 3.1 for higher values of X and Y since the output tensor size is fixed as 2×2. So the capacity pre-log which is bounded by $\min\{X \cdot Y, 2 \cdot 2\} = 4$, does not increase with increasing X and Y. We observe that increasing the size of individual domains of the input tensor does not provide significant gain if the number and size of the corresponding domains of the output tensor are fixed.

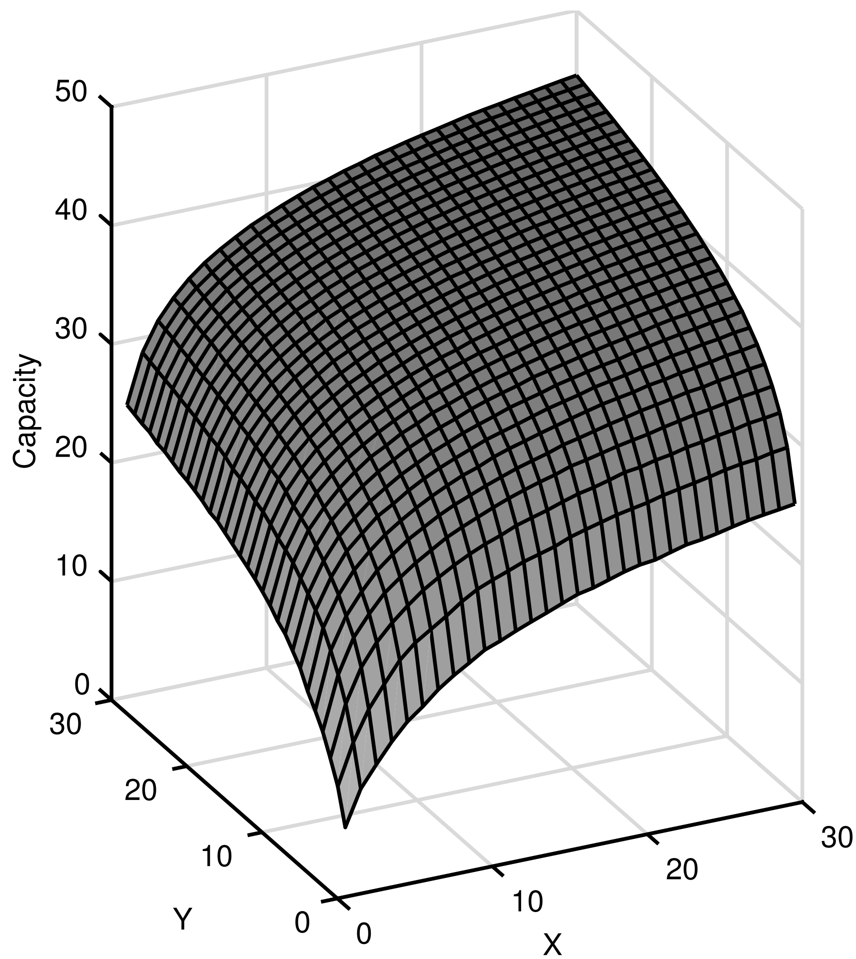


Fig. 3.4: Capacity [bits/channel-use] vs X vs Y for channel with $X \times Y$ size input tensor and 2×2 output tensor.

Capacity under different domain power constraints

In this section we compare different possible power constraints. Algorithm 1 is used to approximate the optimum input covariance and thereby capacity, under per domain element and per element power constraints.

Figure 3.5 illustrates the capacity under sum power constraint and per domain element power constraints for a $2 \times 2 \times 2 \times 2$ channel with power constraint on input tensor of size 2×2. The power budgets on one of the domains of the input tensor are $P_1 = x \cdot P$ and $P_2 = (1 - x) \cdot P$. For instance, assume that the two domains are space and time.

Then such a constraint reflects that the power budget for the first time slot for both the antennas is P_1 and for the second slot for both the antennas is P_2 with total available power $P_1 + P_2 = P$. The plot in Figure 3.5 is presented for capacity against $x = P_1/P$ at 10 dB SNR. The flat line represents the capacity under sum power constraint which shows no variation with x, and the curved line shows the capacity with per domain element power constraints. As can be observed from Figure 3.5, the capacity under per domain element constraints is always lower than the capacity under sum power constraint, and these become very close to each other when $x \approx 0.5$, i.e. uniform power is allotted to the elements of the constrained domain. Note that such a behaviour is observed over an average of 100 channel realizations. For a given specific realization, the two curves may not meet at $x = 0.5$. For the MIMO case, a similar numerical result has been presented in [101] under per antenna power constraints for a fixed channel.

In Figure 3.6 we present the capacity under per element power constraints and compare it with sum power constraint. If the total available power is P, then as before $P_1 = x \cdot P$ and $P_2 = (1-x) \cdot P$. Further, $P_{11} = y \cdot P_1, P_{12} = (1-y) \cdot P_1, P_{21} = y \cdot P_2$ and $P_{22} = (1-y) \cdot P_2$. Thus, with $0 \leq x, y \leq 1$, $P_{11}, P_{12}, P_{21}, P_{22}$ represent the individual power constraints on all the four elements of the input tensor such that $P_{11} + P_{12} + P_{21} + P_{22} = P$. With different choices of x and y, we achieve different per element power constraints such that total power remains P. The capacity with per element power constraints against x and y at SNR of 15 dB is presented in Figure 3.6. The flat surface represents capacity under sum power constraint which shows no variation with x and y, and the curved surface shows capacity under per element power constraints. It can be seen that for different values of x and y, the capacity achieved under per element constraints can be significantly lower than the capacity achieved under sum power constraint.

Note that Algorithm 1 only approximates the optimum input covariance and thus does not provide the exact capacity at low SNRs. However, since the problem at hand is a convex optimization problem, several software tools for numerical optimization can be used to calculate the capacity. To analyze how well the scaling approximation in Algorithm 1 works, we present a comparison between the capacity calculated through the convex op-

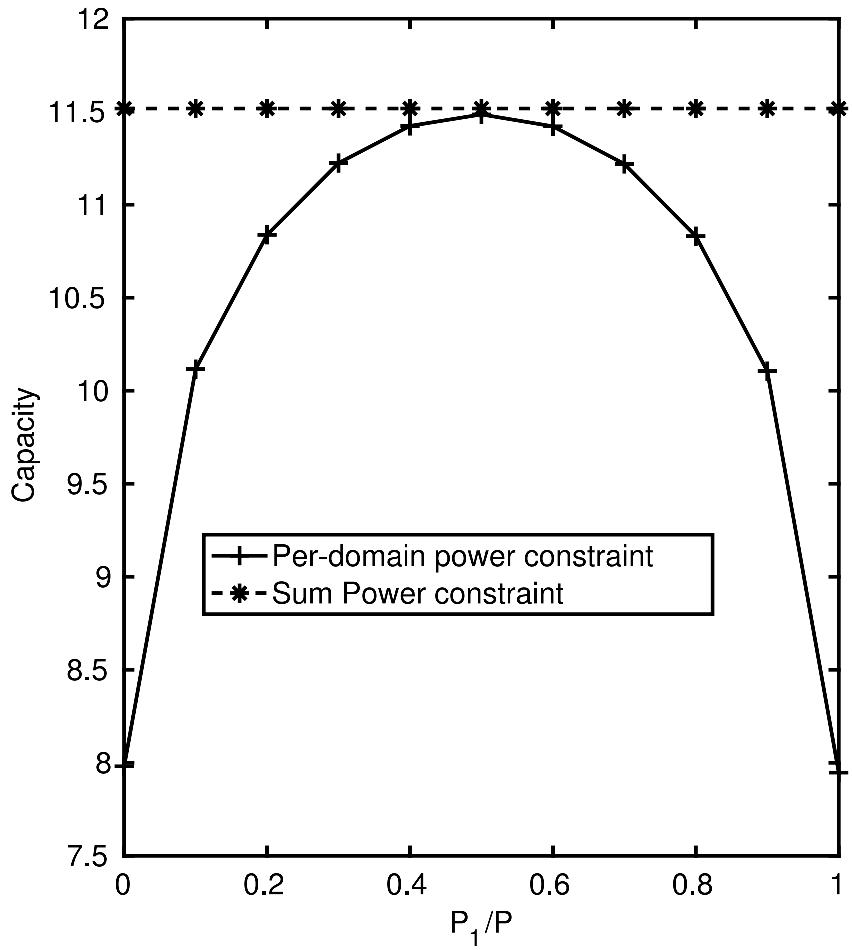


Fig. 3.5: Capacity [bits/channel-use] for a $2 \times 2 \times 2 \times 2$ tensor channel under per domain element power constraints and total power constraint at 10 dB SNR.

timization software tool CVX [152], and the capacity approximated through Algorithm 1. Figure 3.7 presents the capacity for sum power constraint, per domain element power constraints where a single domain of dimension 2 is constrained with power budgets P_1, P_2, and per element power constraints where all the four elements have different power budgets $P_{11}, P_{12}, P_{21}, P_{22}$ against SNR for $x = y = 0.1$. We present such results for capacity calculated by using two methods. First method uses Algorithm 1 for which the graphs are presented using solid curves. Second method uses CVX for which the graphs are presented using dashed curves. As can be observed, the capacity calculated via the approximation

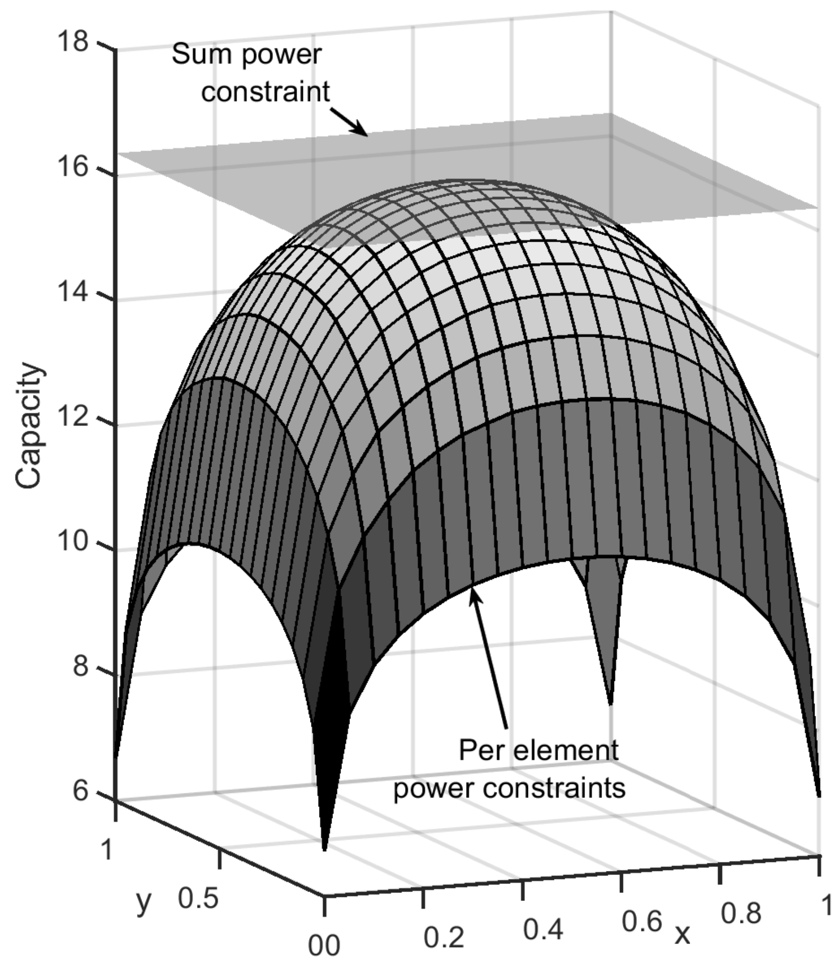


Fig. 3.6: Capacity [bits/channel-use] for per element power constraints vs x vs y at 15 dB SNR.

of Algorithm 1 matches very closely to the one calculated via CVX, and is almost indistinguishable at moderate to high SNR. This shows that Algorithm 1 provides a reasonably good approximation to the optimal solution at low SNR, while providing an exact solution at high SNR. Furthermore, it can be seen that the capacity under per domain element and per element constraints is always upper bounded by the capacity under sum power constraint. The capacity increases with SNR for all three cases, but the performance difference also gradually increases between the three solid curves, with sum power constraint performing the best, followed by per domain element and lastly, the per element constraint.

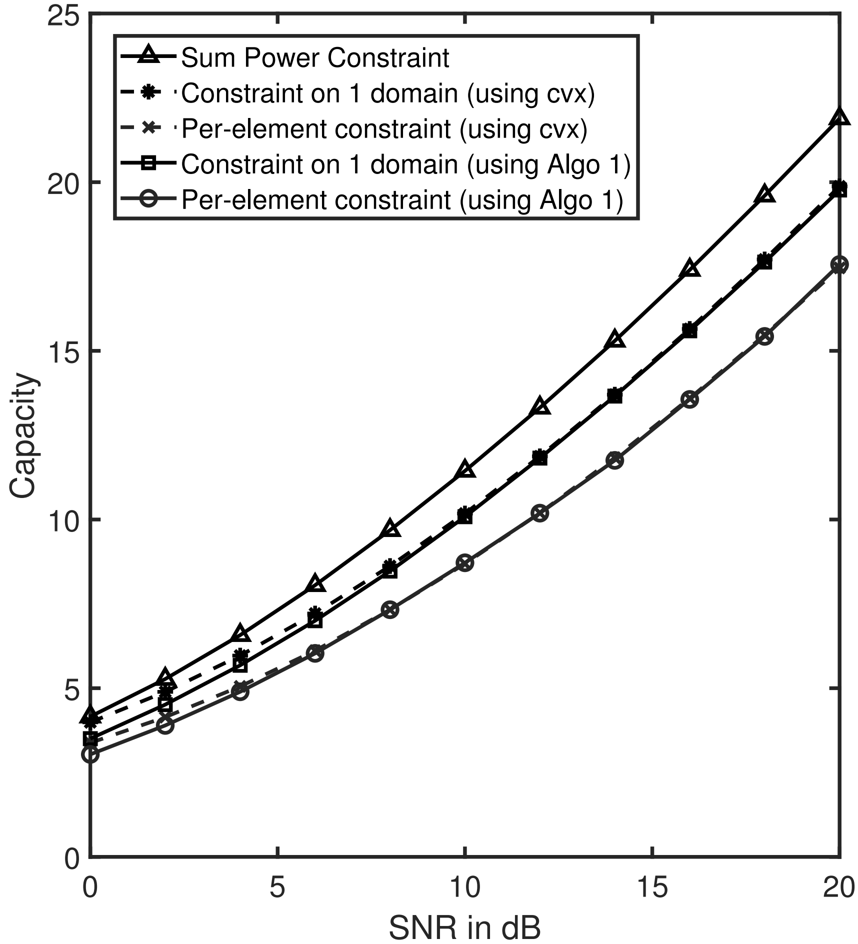


Fig. 3.7: Capacity [bits/channel-use] for different constraints vs SNR using Algorithm 1 and CVX.

Figure 3.8 compares capacity under sum power and per domain element power constraints with $x = 0.1$ for different N where both input and output are order N. The channel is an order $2N$ tensor and the size of each domain is 2. The capacity increases exponentially with N in both the cases. However, the capacity under per domain element constraint is always upper bounded by the capacity under sum power constraint.

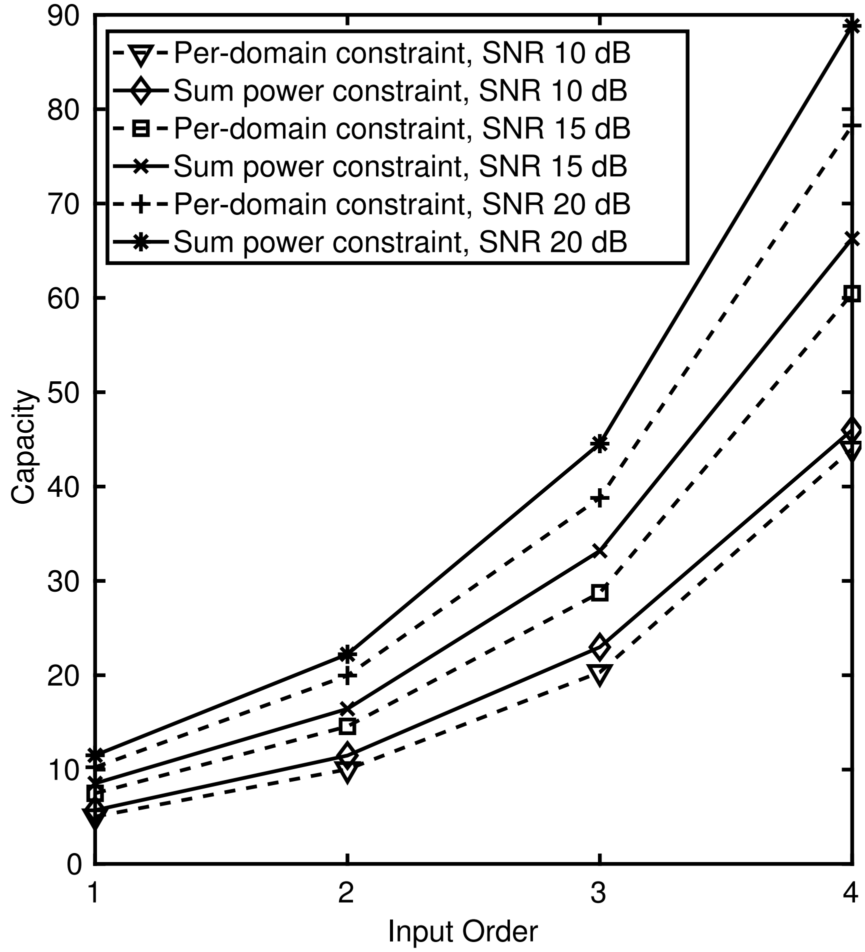


Fig. 3.8: Capacity [bits/channel-use] vs input order comparing sum power and per domain element power constraints with $P_1/P = 0.1$.

Correlated Tensor Channel

Consider an order N random tensor $\boldsymbol{\mathcal{H}} \in \mathbb{C}^{I_1 \times \ldots \times I_N}$ containing i.i.d. zero mean and unit variance elements. Let $\Psi^{(n)} \in \mathbb{C}^{I_n \times I_n}$ for $n = 1, \ldots, N$ be a sequence of Hermitian matrices such that $\Psi^{(n)} = \mathrm{A}^{(n)}\mathrm{A}^{(n)H}$ where $\mathrm{A}^{(n)} \in \mathbb{C}^{I_n \times I_n}$ is the square root matrix of $\Psi^{(n)}$. The mode-n product of tensor $\boldsymbol{\mathcal{H}}$ across all the modes with these matrices can be written as [160]:

$$\boldsymbol{\mathcal{H}}^{corr} = \boldsymbol{\mathcal{H}} \times_1 \mathrm{A}^{(1)} \times_2 \mathrm{A}^{(2)} \times_3 \cdots \times_N \mathrm{A}^{(N)} \tag{3.68}$$

where the elements of $\boldsymbol{\mathcal{H}}^{corr}$ are correlated. Let vec($\boldsymbol{\mathcal{H}}$) be denoted as $\underline{\mathbf{h}}$, then using the property of mode-n product from [160],[161, Lemma 2.1], we can write (3.68) using the Kronecker product denoted by $\otimes$ as :

$$\mathrm{vec}(\boldsymbol{\mathcal{H}}^{corr}) = (\mathrm{A}^{(N)} \otimes \cdots \otimes \mathrm{A}^{(1)})\underline{\mathbf{h}}. \tag{3.69}$$

The correlation matrix of the vectorized channel can be written as :

$$\mathbb{E}[\mathrm{vec}(\boldsymbol{\mathcal{H}}^{corr})\,\mathrm{vec}(\boldsymbol{\mathcal{H}}^{corr})^H]$$
$$= \mathbb{E}[(\mathrm{A}^{(N)} \otimes \cdots \otimes \mathrm{A}^{(1)})\underline{\mathbf{h}} \cdot \underline{\mathbf{h}}^H(\mathrm{A}^{(N)} \otimes \cdots \otimes \mathrm{A}^{(1)})^H] \tag{3.70}$$
$$= (\mathrm{A}^{(N)} \otimes \cdots \otimes \mathrm{A}^{(1)})\mathbb{E}[\underline{\mathbf{h}} \cdot \underline{\mathbf{h}}^H](\mathrm{A}^{(N)} \otimes \cdots \otimes \mathrm{A}^{(1)})^H \tag{3.71}$$
$$= (\mathrm{A}^{(N)} \otimes \cdots \otimes \mathrm{A}^{(1)})(\mathrm{A}^{(N)} \otimes \cdots \otimes \mathrm{A}^{(1)})^H \tag{3.72}$$
$$= (\mathrm{A}^{(N)}\mathrm{A}^{(N)H} \otimes \cdots \otimes \mathrm{A}^{(1)}\mathrm{A}^{(1)H}) \tag{3.73}$$
$$= \Psi^{(N)} \otimes \cdots \otimes \Psi^{(1)} \tag{3.74}$$

where (3.72) to (3.73) follow from matrix Kronecker product properties [162, Corollary 4]. Thus the correlation matrix of the vectorized tensor is given in terms of the Kronecker product of different mode-n factor correlation matrices denoted by $\Psi^{(n)}$. Such a model is called a separable correlation model and it is considered for real random variables in [163]. While (3.74) expresses the correlation as a matrix by vectorizing $\boldsymbol{\mathcal{H}}^{corr}$, it is shown in [163, Proposition 2.1] that the correlation of $\boldsymbol{\mathcal{H}}^{corr}$ from (3.68) can also be expressed as an order $2N$ tensor obtained via the outer product of the factor matrices $\Psi^{(n)}$ defined as $\bar{\mathcal{R}} = \Psi^{(1)} \circ \cdots \circ \Psi^{(N)}$. Note that the correlation tensor when defined as $\mathcal{R} = \mathbb{E}[\boldsymbol{\mathcal{H}}^{corr} \circ \boldsymbol{\mathcal{H}}^{corr*}]$ is just a permuted version of $\bar{\mathcal{R}}$, where $\mathcal{R}_{i_1,\ldots,i_N,i'_1,\ldots,i'_N} = \bar{\mathcal{R}}_{i_1,i'_1,\ldots,i_N,i'_N} = \mathbb{E}[\boldsymbol{\mathcal{H}}^{corr}_{i_1,\ldots,i_N} \cdot \boldsymbol{\mathcal{H}}^{corr*}_{i'_1,\ldots,i'_N}] = \Psi^{(1)}_{i_1,i'_1} \cdots \Psi^{(N)}_{i_N,i'_N}$. Hence the separable model implies that each element in the order $2N$ correlation tensor can be expressed as the product of the elements of the factor matrices. The proof in [163] is for real tensors, but can be easily applied to complex tensors as well.

The well known MIMO matrix Kronecker correlation model forms a specific case of (3.68) where the tensor $\boldsymbol{\mathcal{H}}$ is order-2 and the factor matrices $\mathrm{A}^{(1)}$ and $\mathrm{A}^{(2)}$ denote the square root of row and column correlation matrices respectively [164]. The MIMO Kronecker model may not be very accurate in several scenarios, however it is still widely used because of its tractable analytic form, see for example [165, 95, 166, 167].

Now consider an order-4 tensor channel of size $3 \times 3 \times 3 \times 3$ corresponding to an order-2 input and order-2 output. We generate such a channel $\boldsymbol{\mathcal{H}}^{corr}$ with correlated elements using (3.68), where the elements of $\boldsymbol{\mathcal{H}}$ are i.i.d zero mean complex Gaussian with unit variance. For the numerical examples, we consider the correlation matrices generated using the exponential model with different correlation factor ρ_n where the elements of the correlation matrix $\Psi^{(n)}$ are defined as $\Psi^{(n)}_{i,j} = \rho_n^{|i-j|}$ for $\rho_n \in [0, 1)$ [168]. The four correlation matrices $\Psi^{(1)}, \Psi^{(2)}, \Psi^{(3)}$ and $\Psi^{(4)}$ are generated using the exponential model with correlation coefficients ρ_1, ρ_2, ρ_3 and ρ_4 respectively. Assuming that the channel realization is known at the transmitter and the receiver, we find the capacity of such a channel with correlated elements under sum power constraint. Figure 3.9 presents capacity at 10 dB SNR for different values of correlation coefficients where the receive domains are correlated with $\rho_1 = \rho_2 = \rho_R$ and the transmit domains are correlated with $\rho_3 = \rho_4 = \rho_T$. The plot shows that the capacity decreases with increase in ρ_T and ρ_R, and it is least when ρ_T and ρ_R approach 1. Capacity is largest when both ρ_T and ρ_R approach zero in which case the correlation matrices are identity and the channel has only uncorrelated elements across all the domains.

Next we investigate the impact of correlation on the tensor channel capacity when the correlation spans over a variable number of domains. Figure 3.10 presents the capacity against SNR for different number of domains having correlated entries. Capacity is lowest when ρ_n is non-zero (0.7 in the figure) for all the domains (i.e. for $n = 1, 2, 3, 4$) and is highest when ρ_n is zero for all n, i.e. all entries are uncorrelated. Further it can be observed that the capacity difference between the various cases presented in Figure 3.10 is more significant at higher SNR. It is seen that the capacity decreases with increase in the number of domains having non-zero correlation factor. Such a loss of capacity with increase in the domains having correlation is further illustrated in Figure 3.11 for various tensor channel order.

In Figure 3.11, we present the capacity against SNR for order $2M$ correlated tensor channels with order M input and output having individual dimensions of 3. The factor correlation matrices along each of the $2M$ modes are based on the exponential model with correlation factor ρ. The graph is presented for $M = 2, 3, 4, 5$ and $\rho = 0.4, 0.7$. Note that different values of M lead to different tensor channel sizes with order $4, 6, 8$, and 10. Hence for a meaningful comparison of the impact of correlation, the capacity plotted in Figure 3.11 is normalized with respect to the number of elements in the input tensor symbol which is 3^M. As can be observed in Figure 3.11, for a fixed ρ, as the number of domains of the tensor channel with all correlated elements increases, the capacity per element decreases. Further, this loss is more significant for higher correlation (larger values of ρ) as can be seen by comparing the curves for $\rho = 0.7$ and 0.4.

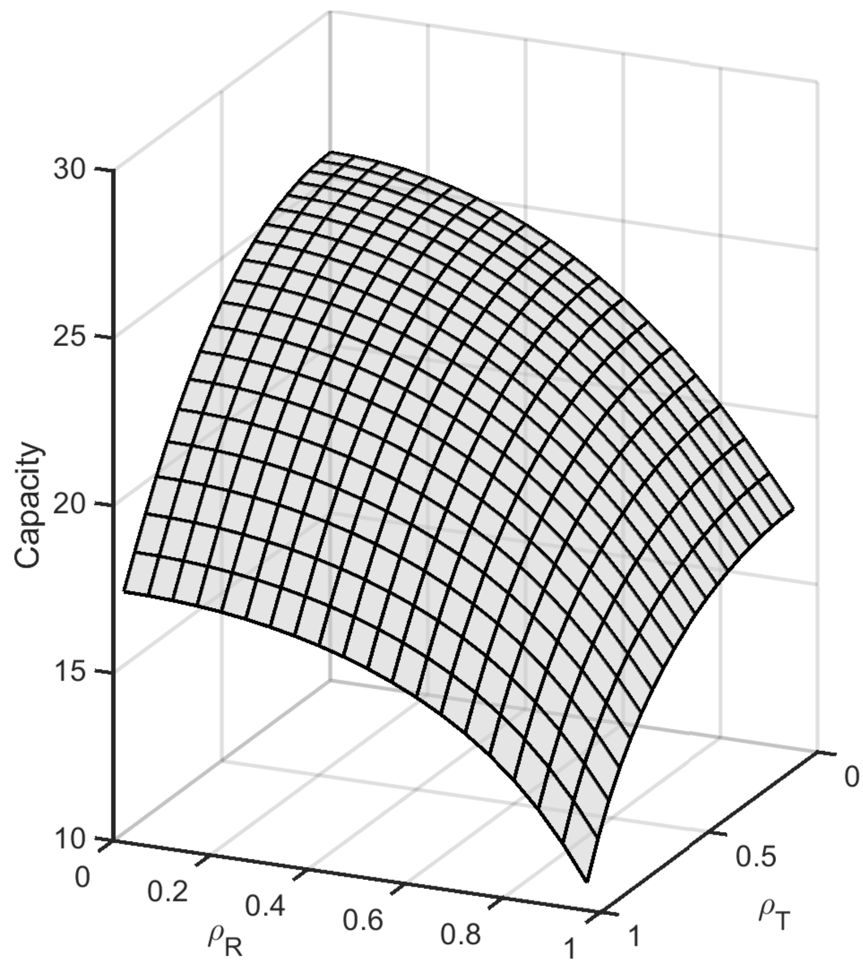


Fig. 3.9: Capacity [bits/channel-use] vs correlation coefficients for tensor channel with correlated entries.

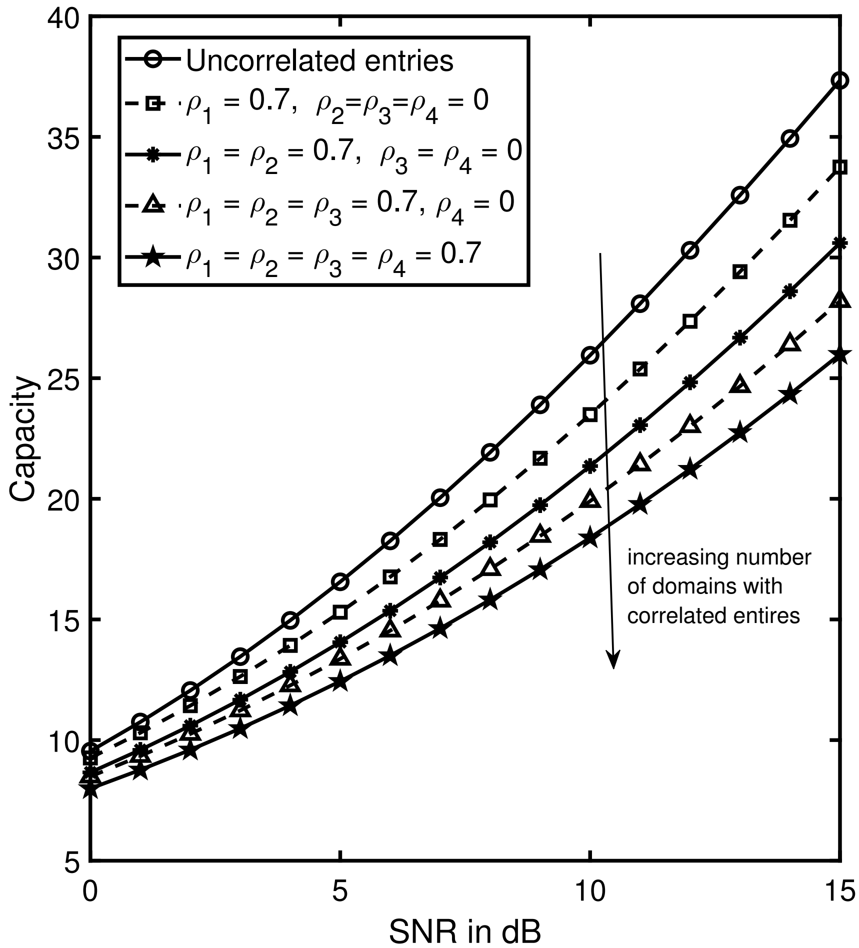


Fig. 3.10: Capacity [bits/channel-use] vs SNR for tensor channel with correlated entries.

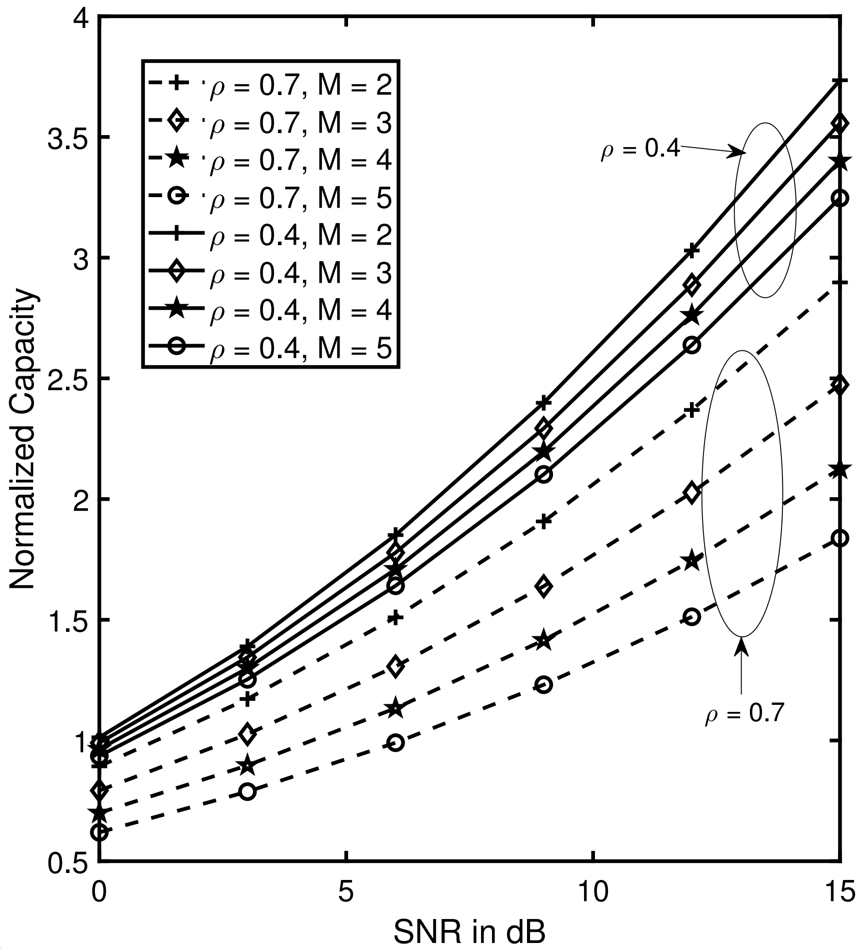


Fig. 3.11: Capacity [bits/channel-use] normalized with respect to the size of transmit tensor vs SNR for different order tensor channels with correlated entries.

Domain Trade-Off

An advantage of looking at the channel as a higher order tensor is that the trade-off which may exist between different domains can be mathematically analyzed and harnessed. The lack of resources in one domain could be compensated by resources in another domain. This paves the path for certain flexibility in resource allocation across domains. To demonstrate such a domain trade-off, we consider Figure 3.12 which illustrates the behaviour of the capacity under sum-power constraint against SNR. The transmitter employs a vector input, i.e. a single domain of dimension 16 and the output is a tensor of various number of domains.

The channel capacity curves show an increase in the capacity achieved when the number of receive domains is increased. It can also be observed that the capacity achieved by a 16×16 channel is same as that of the capacity achieved by an $16 \times 2 \times 2 \times 2 \times 2$ channel. This implies that even with limited resources in one domain, increasing the number of domains can give higher capacity by exploiting the trade-off between multiple domains.

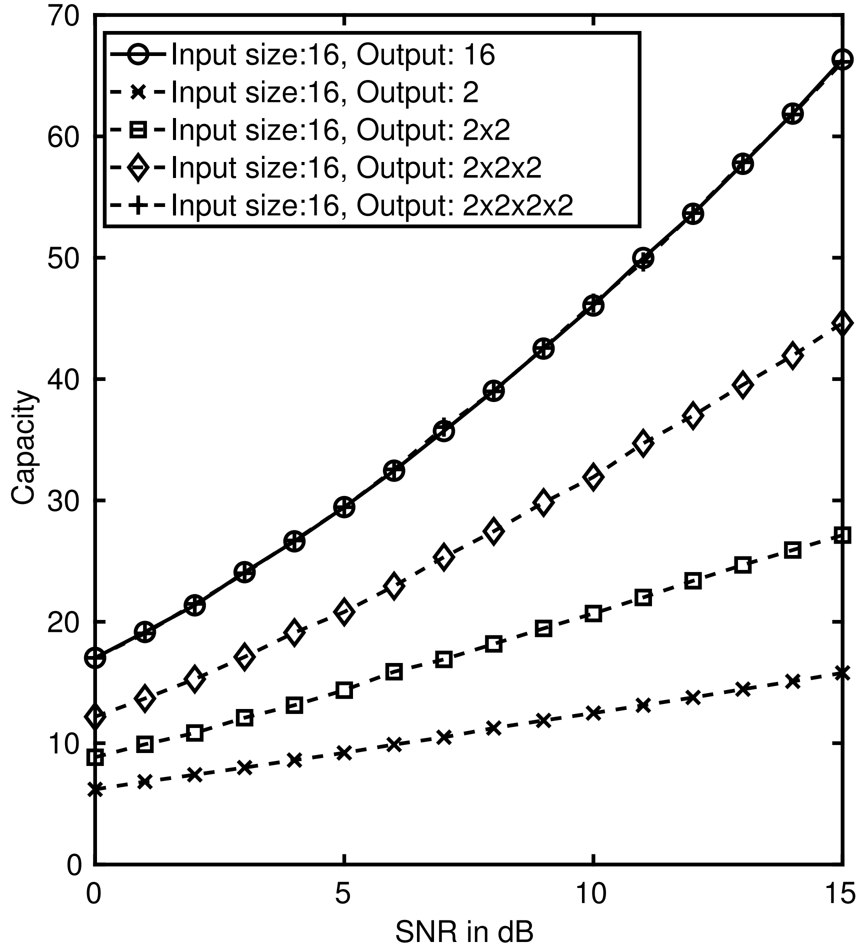


Fig. 3.12: Capacity [bits/channel-use] vs SNR for 16×1 input and different configurations of output structure.

Figure 3.12 suggests that the individual domain's dimensions can be flexibly interchanged if the overall size of the input and output remains constant. However, it is important to note that such a behaviour is observed over an average of 100 channel realizations when all the tensor channel elements are i.i.d. Gaussian with zero mean and unit variance, which has been the case in our numerical examples so far. For any two given tensor channels with same overall size but different dimensions of individual domains, the capacity may not always be exactly same. The exact behavior of such a domain trade-off would depend on the specific tensor channel eigenvalues. The example of a channel with i.i.d. Gaussian entries without assigning specific physical meaning to the domains incorporated is presented only to illustrate the basic idea of trade-off. Further, to understand such a trade-off in a more realistic set-up, we consider the example of a MIMO GFDM channel in section 3.3.3.

3.3.2 Tensor Channels for various Input-Output Configurations

For a fixed order tensor channel, different configurations of input and output can lead to different capacities. Hence, an insightful comparison between the capacities of tensor channels with different orders requires to clearly specify the divide between the number of input and output domains. For instance, when we state the capacity of an order-6 channel, we need to specify that it is corresponding to say order-3 input and order-3 output, or order-4 input and order-2 output, or any other possible configuration. In this section, we analyze how the capacity for a given tensor channel changes with change in such input and output configurations. We will fix the tensor channel order $M + N$ and increase the input order N (consequently decrease output order M) to understand the capacity behavior. All the results presented in this section are averaged over 100 different channel realizations unless otherwise stated. Also, the dimension of individual domains for all the examples in this section is taken as 2.

Capacity Against Input Order

Figure 3.13 presents the capacity under sum-power constraint of an order-10 tensor channel plotted against the input order N. The output order is $10 - N$. The capacity is plotted for different values of transmit power budget P. The channel is generated as described in the previous section where each element of the channel is drawn from a circularly symmetric complex Gaussian distribution with zero mean and unit variance. It can be seen that for such tensor channels, at high transmit powers, the best configuration of the input and output is when both input and output orders are same for almost any power budget. The difference in the capacity between the case with order-5 input order-5 output, and with any other input-output configuration can be significant especially as the transmit power increases. This is because with an equal divide between the transmit and receive side, we get maximum number of parallel decomposed scalar channels which is same as the number of non-zero singular values of $\mathcal{H}$, and is given as $R \leq \min\{I_1 \cdots I_N, J_1 \cdots J_M\}$. Since we consider individual dimensions of size 2, for this case we have $R \leq \{2^N, 2^{10-N}\}$ which is maximized when $N = 5$. Hence, if sufficient power is available such that power is allotted to most of the eigen channels, having larger number of such channels, i.e. $N = 5$ is preferred. However, as the transmit power decreases, we observe the the peak in the curve shifts towards very low and a very large value of N, indicating that as P decreases a channel with fewer number of domains on a given side may provide larger capacity than the case with equal number of domains on both sides. At low P, the tensor water-filling tends to allocate power to selected strong channels only. Thus having large number of parallel scalar channels is not beneficial since the limited power budget would not allow any power allocation on all such channels. Thus in such a case, having fewer but stronger parallel channels is better than having more number of channels. For a very low power budget such as $P = -15$ dB this is achieved when $R \leq \{2^N, 2^{10-N}\}$ is minimized for which $N = 1$ or $N = 9$.

However, it should be noted that such an observation over an average of several channel realizations depends on the specific distribution used to generate the tensor channel, and thus should not be generalized for any arbitrary channel. In this particular example, since the channel elements are drawn from zero mean unit variance distribution, the channel gain increases with increasing channel size. Channel gain is defined as the gain in power at the receiver provided by the channel assuming uniform power allocation at the transmitter [159]. For any given channel, the total received signal power P_R is given in terms of total

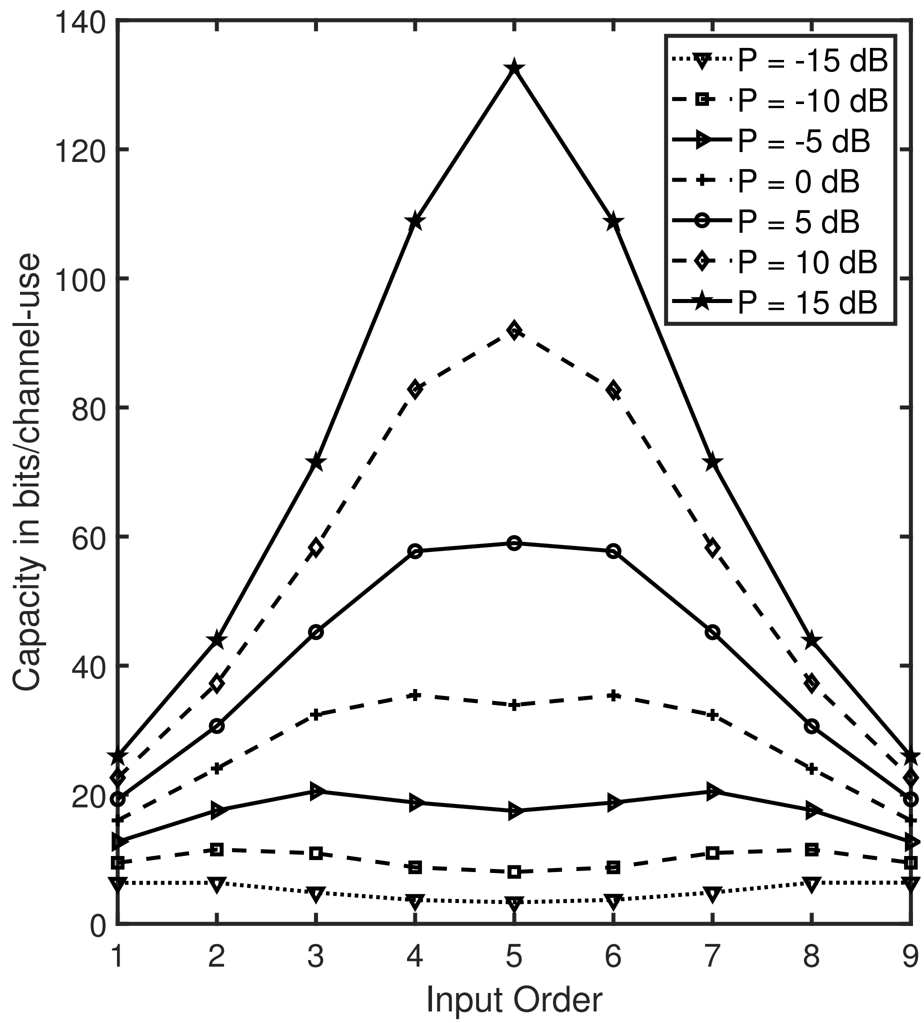


Fig. 3.13: Capacity [bits/channel-use] vs Input Order for Order-10 tensor channel.

transmit signal power P as [159]

$$P_R = ||\mathcal{H}||^2 \frac{P}{I} \tag{3.75}$$

where I denotes the total number of input elements. If the channel elements are generated using zero mean unit variance, then the average received signal power over several channel realizations would be given as :

$$\bar{P}_R = \mathbb{E}[||\boldsymbol{\mathcal{H}}||^2]\frac{P}{I} = \sum_{j_1,\ldots,j_M,i_1,\ldots,i_N} \underbrace{\mathbb{E}[|\boldsymbol{\mathcal{H}}_{j_1,\ldots,j_M,i_1,\ldots,i_N}|^2]}_{=1} \frac{P}{I} = J \cdot I \cdot \frac{P}{I} = J \cdot P. \tag{3.76}$$

which scales with J where $J = J_1 \cdots J_M$ denotes the total number of receive elements. Thus for a given order tensor channel, changing the input and output configurations would change the power gain provided by the channel at the receiver. Hence, for a fair comparison between the performance of different input output orders for a fixed channel, it is important to normalize the channel such that it does not provide any power gain at the receiver. We consider this case next.

Figure 3.14 presents the capacity under sum power constraint of order-10 tensor channel against the input order N, when the channel elements are normalized to provide unit power gain at the receiver. For this purpose, the channel is generated using circularly symmetric complex Gaussian with zero mean and variance $1/(J_1 \cdots J_M)$ as used for the example of Figure 3.2. In such a case, $\mathbb{E}[||\boldsymbol{\mathcal{H}}||^2] = I$ and thus the average received signal as calculated using (3.76) would be same as the transmit signal power. As can be seen that the curves in Figure 3.14 significantly differ from Figure 3.13. In Figure 3.14, the curves are not centred around $N = 5$, but the peaks are shifted towards larger values of N. Also the peak value occurs at different N for different power budgets P. At $P = 15$ dB, input order-6 and output order-4 gives the peak value, whereas at $P = 0$ dB, input order-8 and output order-2 gives the peak value. The peak shifts towards the center as P increases. For such normalized channels, as the input order increases, output order decreases, thus $J_1 \cdots J_M$ decreases, which makes the individual components of the channel stronger. Hence the capacity tends to increase with increasing input order. However at a high transmit power, the increase in capacity with increasing input order takes place only till a specific input order. Note that there are two separate factors that contribute to the channel capacity : the structure of the channel, and the transmit power. The structure of the channel determines the number of non-zero eigenvalues of $(\mathcal{H}^H *_M \mathcal{H})$, which also represents the number of equivalent decomposed parallel scalar channels which carry information. The number of non-zero eigenvalues of $(\mathcal{H}^H *_M \mathcal{H})$ is same as the number of non-zero singular values of $\mathcal{H}$ which is given as $R \leq \min\{I, J\}$ where equality is achieved if all the singular values are non-zero. With increasing input order and decreasing output order, R reduces, thus the number of decomposed parallel scalar channels also reduces. At low P, this is beneficial because the tensor water-filling would anyway favor selected strong channels for transmission and most power would be concentrated on the stronger channels. However, with larger P, power tends to get distributed more evenly across all the decomposed channels, thus having larger R is beneficial which can be achieved when input and output order are closer since $R \leq \min\{I, J\}$. This inherent interplay between the available power level and the available number of input domains causes the peak of the curves in Figure 3.14 to shift towards centre as P increases.

Figure 3.15 illustrates this behavior for various orders of tensor channels from 4 to 8 for various power levels. The x axis in all the sub-plots denote the input order, and the channel is normalized to provide unit power gain. In all the sub-plots it can be seen that the peak is shifted in favor of larger input order and smaller output order for smaller values of P such as 0 dB and 5 dB. For higher values of P also, the peak is towards larger N and smaller M, but it gradually shifts towards the centre. The best input output configurations as observed from Figure 3.15 are summarized in Table 3.3 where it can be seen that for smaller values of P, the value of output order M is almost always 1 and N takes the largest possible value. However as P and channel order increases, we see the value of M is increasing and getting closer to the value of N.

To further analyze the behavior with respect to the number of transmit elements in the input tensor, Figure 3.16 presents the capacity per input tensor element for the same example as Figure 3.14. It can be seen that with increase in the input order there is a decrease in the capacity per input element. As the input order increases, the same power is getting distributed to more individual elements, hence the capacity per input element

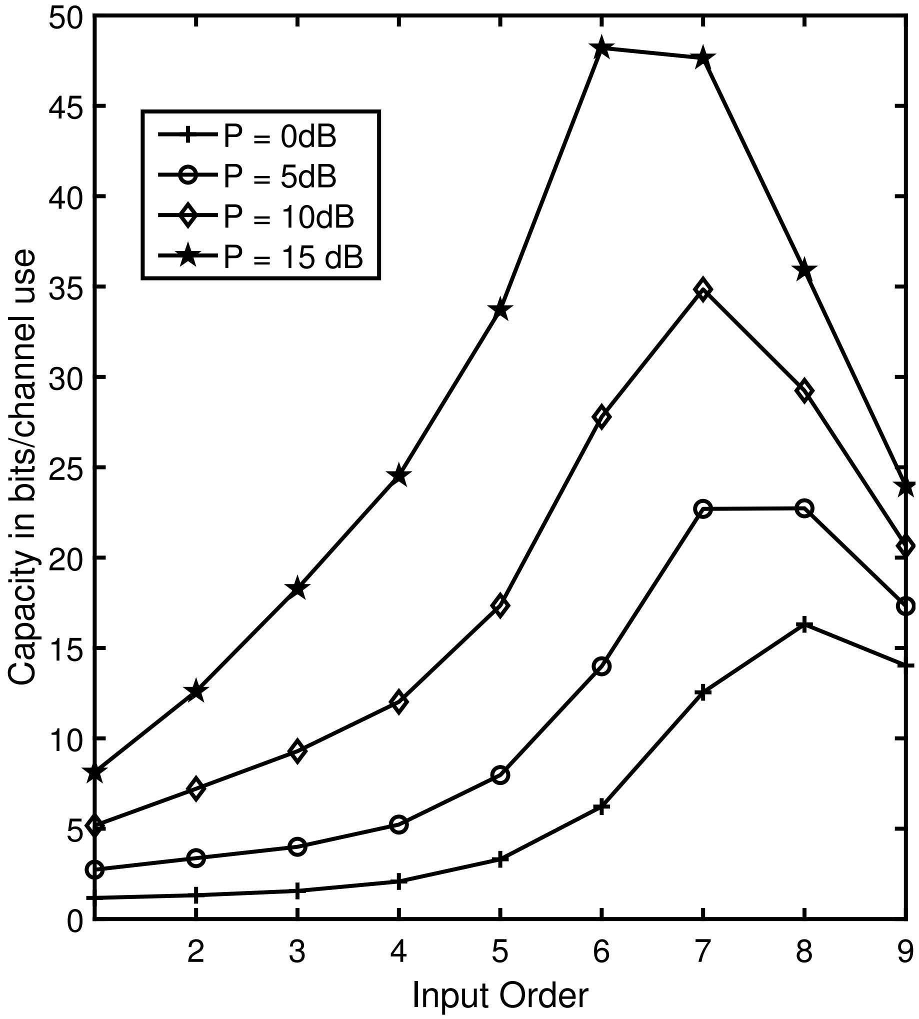


Fig. 3.14: Capacity [bits/channel-use] vs Input Order for Order-10 tensor normalized channel having unit power gain.

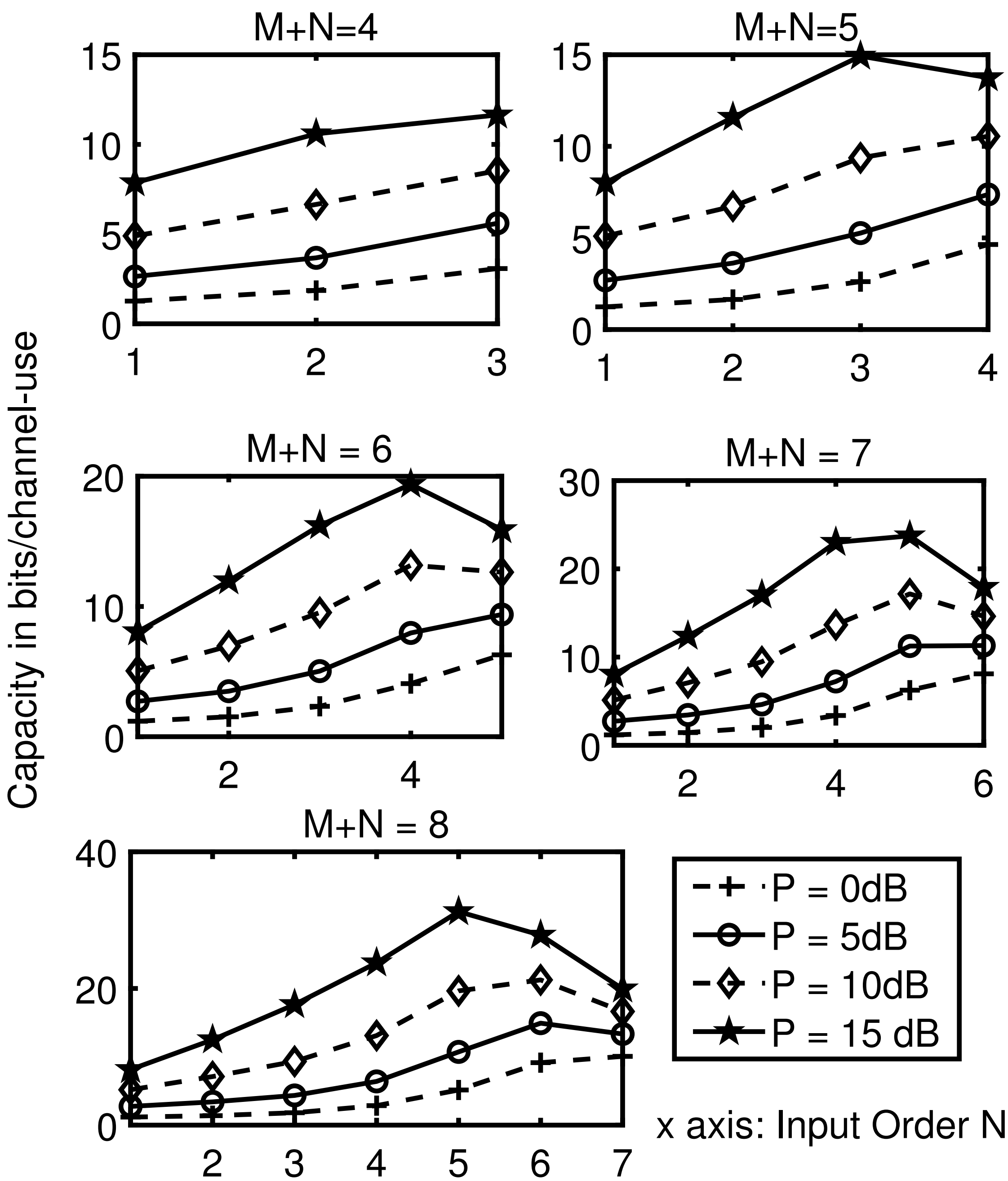


Fig. 3.15: Capacity [bits/channel-use] vs Input Order for various order tensor normalized channel having unit power gain.

Table 3.3: Best Input and Output Orders (N, M) for Normalized Gaussian Tensor Channels with Capacity (C) in bits/channel-use.

Channel Order	P = 15 dB	P = 10 dB	P = 5 dB	P = 0 dB
4	$N = 3, M = 1,$ $C = 11.64$	$N = 3, M = 1,$ $C = 8.53$	$N = 3, M = 1,$ $C = 5.60$	$N = 3, M = 1,$ $C = 3.07$
5	$N = 3, M = 2,$ $C = 14.92$	$N = 4, M = 1,$ $C = 10.54$	$N = 4, M = 1,$ $C = 7.37$	$N = 4, M = 1,$ $C = 4.64$
6	$N = 4, M = 2,$ $C = 19.38$	$N = 4, M = 2,$ $C = 13.16$	$N = 5, M = 1,$ $C = 9.38$	$N = 5, M = 1,$ $C = 6.28$
7	$N = 5, M = 2,$ $C = 23.73$	$N = 5, M = 2,$ $C = 17.16$	$N = 6, M = 1,$ $C = 11.31$	$N = 6, M = 1,$ $C = 8.09$
8	$N = 5, M = 3,$ $C = 31.24$	$N = 6, M = 2,$ $C = 21.25$	$N = 6, M = 2,$ $C = 14.92$	$N = 7, M = 1,$ $C = 10.06$

decreases. However, the total capacity can be significantly larger with large N as shown in Figure 3.14. Moreover, having large N provides more robustness to the system in case of any failed transmissions. With small N, any failure of transmission on an individual element would cause a significant loss of total capacity. However, for large N, loosing transmission on a few transmit elements would not hamper the overall capacity much since the contribution of individual components is small. We explain this notion further through examples.

Robustness with increasing input order

We consider tensor channels of various orders where each domain has a dimension 2 with circularly symmetric complex Gaussian elements normalized to provide unit power gain at the receiver. We find the optimum input covariance and the capacity of such a tensor channel under sum power constraints for various input orders using (3.48) and (3.50). Let us denote this input covariance as $\mathcal{Q}^{(1)}$ and capacity as C_1. Further, we find the capacity when there is one or more failed transmissions at the input. Let the number of input elements which failed transmission be denoted by n. We plot the relative loss in capacity as a function of such failed transmissions n. To simulate n failed or disabled input elements, we randomly select n elements out of the $I_1 \cdot I_2 \cdots I_N$ elements in the input tensor and set them to 0. To this effect, the corresponding values in the covariance tensor $\mathcal{Q}^{(1)}$ are also set

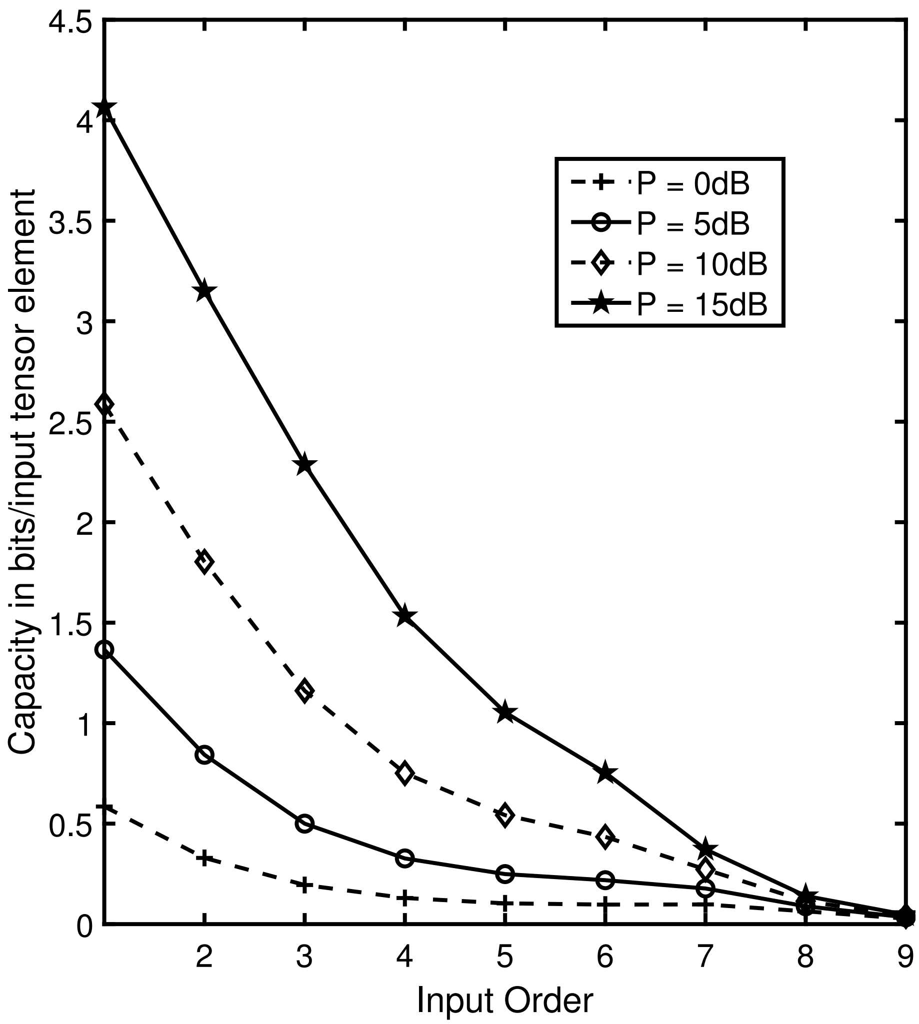


Fig. 3.16: Capacity [bits/input tensor element] vs Input Order for Order-10 tensor normalized channel with unit power gain.

to 0 to obtain a modified covariance $\mathcal{Q}^{(2)}$. For instance, if a randomly selected failed input element is $\mathcal{X}_{i'_1,\ldots,i'_N}$, then we set $\mathcal{Q}^{(1)}_{i_1,\ldots,i_N,j_1,\ldots,j_N} = 0$ wherever either $i_1 = i'_1, \ldots, i_N = i'_N$ or $j_1 = i'_1, \ldots, j_N = i'_N$ to get $\mathcal{Q}^{(2)}$. Let the modified input be denoted by $\tilde{\mathcal{X}}$ which is defined as :

$$\tilde{\mathcal{X}}_{i_1,\ldots,i_N} = \begin{cases} 0, & \text{if } i_1 = i'_1, \ldots, i_N = i'_N \\ \mathcal{X}_{i_1,\ldots,i_N}, & \text{otherwise} \end{cases} \tag{3.77}$$

and the modified covariance denoted by $\mathcal{Q}^{(2)} = \mathbb{E}[\tilde{\mathcal{X}} \circ \tilde{\mathcal{X}}^*]$, be defined as :

$$\mathcal{Q}^{(2)}_{i_1,\ldots,i_N,j_1,\ldots,j_N} = \mathbb{E}[\tilde{\mathcal{X}}_{i_1,\ldots,i_N} \tilde{\mathcal{X}}^*_{j_1,\ldots,j_N}] = \begin{cases} 0, & \text{if } i_1=i'_1,\ldots,i_N=i'_N \text{ or } j_1=i'_1,\ldots,j_N=i'_N \\ \mathcal{Q}^{(1)}_{i_1,\ldots,i_N,j_1,\ldots,j_N}, & \text{otherwise} \end{cases} \tag{3.78}$$

Thus $\mathcal{Q}^{(2)}$ by definition represents the input covariance with failed transmission and the capacity under such failed transmission is calculated using $C_2 = \log[\det(\mathcal{I}_M + \mathcal{H} *_N \mathcal{Q}^{(2)} *_N \mathcal{H}^H)]$, and the relative loss in capacity is calculated using $\dfrac{C_1 - C_2}{C_1}$.

The relative loss in capacity is plotted in Figure 3.17 and 3.18 for order-6 tensor channel at $P = 0$ and 10 dB respectively. It can be seen that for smaller values of N, the loss in capacity with even a single disabled input can be very significant. The loss reaches a value of 1 for $N = 1, n = 2$, for $N = 2, n = 4$ and for $N = 3, n = 8$ because with individual dimensions set as 2, these cases correspond to all the input elements being disabled for transmission making $C_2 = 0$. For larger values of N, the loss in capacity is small since the contribution of individual elements is not too significant in the aggregate scheme due to the large number of input elements carrying information. This demonstrates the robustness of transmission achieved due to large number of transmit elements. This phenomenon is further accentuated in Figures 3.19 and 3.20 which present the relative loss in capacity for order-10 tensor channels at $P = 0$ and 10 dB respectively. In Figures 3.19 and 3.20, as N increases the loss in capacity becomes almost negligible with increased number of disabled elements.

It can be observed that the loss in capacity tends to increase almost linearly with n especially at higher P and lower values of N. When $N \leq M$, with channel containing i.i.d.

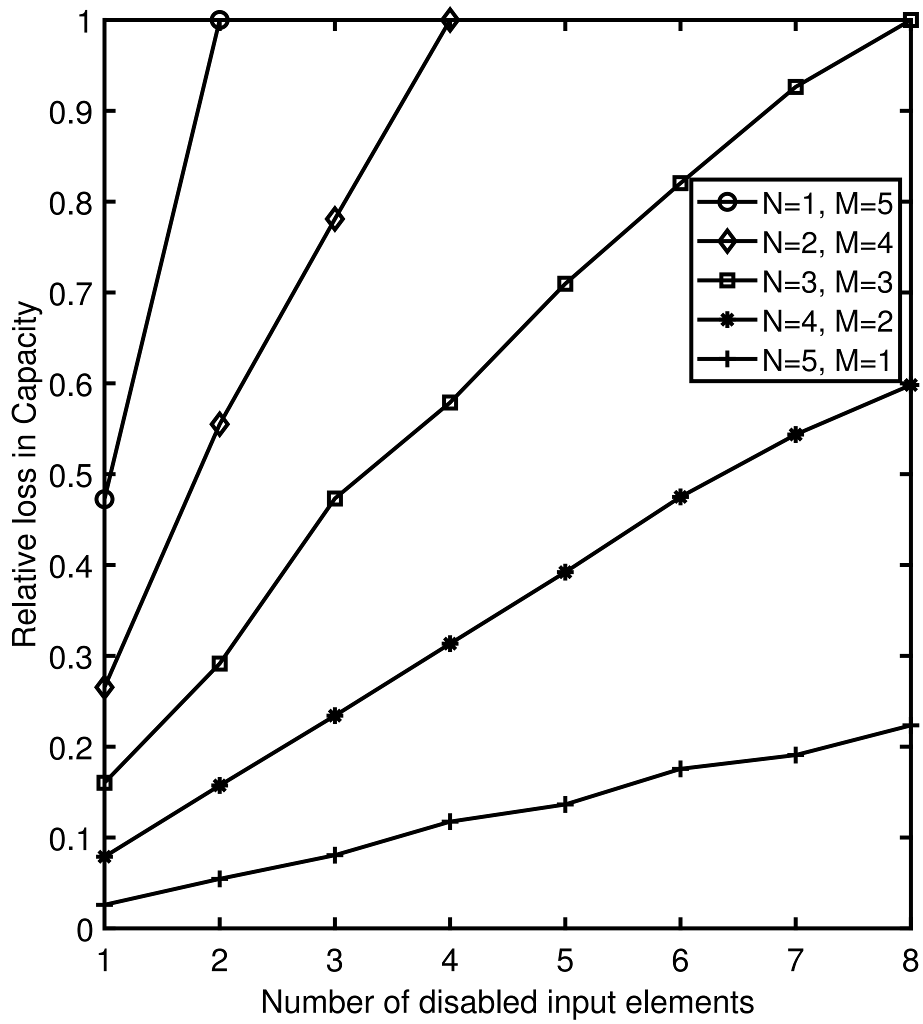


Fig. 3.17: Relative loss in Capacity due to disabled transmit elements for order-6 tensor channel at $P = 0$ dB.

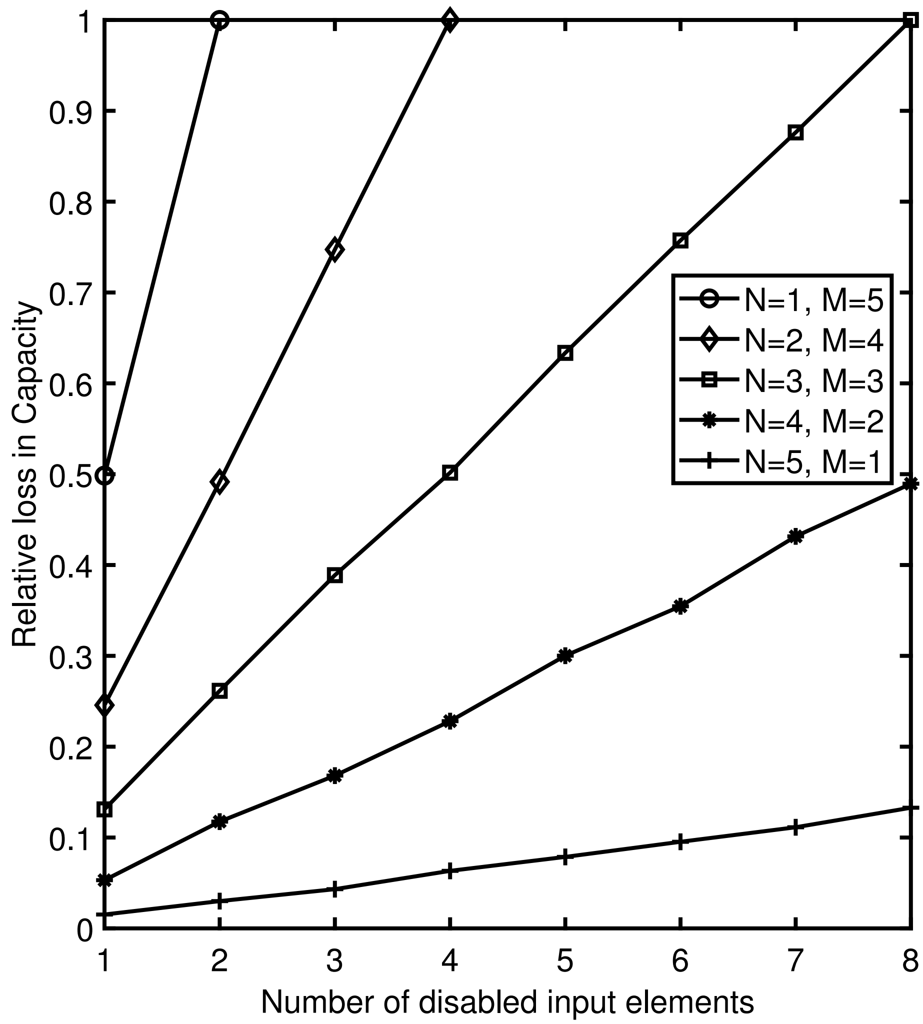


Fig. 3.18: Relative loss in Capacity due to disabled transmit elements for order-6 tensor channel at $P = 10$ dB.

elements the number of non-zero singular values of the channel is $R = I$ where I denotes the number of elements in the input tensor. At high P, since water-filling tends to uniform power allocation across all the $R = I$ non-zero eigenvalues thus $(\mu^{-1}\mathcal{I}_N - \mathcal{D}^{-1})^+$ from (3.48) tends to a scaled identity tensor $\frac{P}{I}\mathcal{I}_N$. This makes the optimal covariance from (3.48) also $\frac{P}{I}\mathcal{I}_N$. Thus as P increases, almost all the input elements receive similar power, and they all carry comparable amount of information which causes the almost linear increase of loss in capacity with n. Such linear behavior gets more pronounced as P increases.

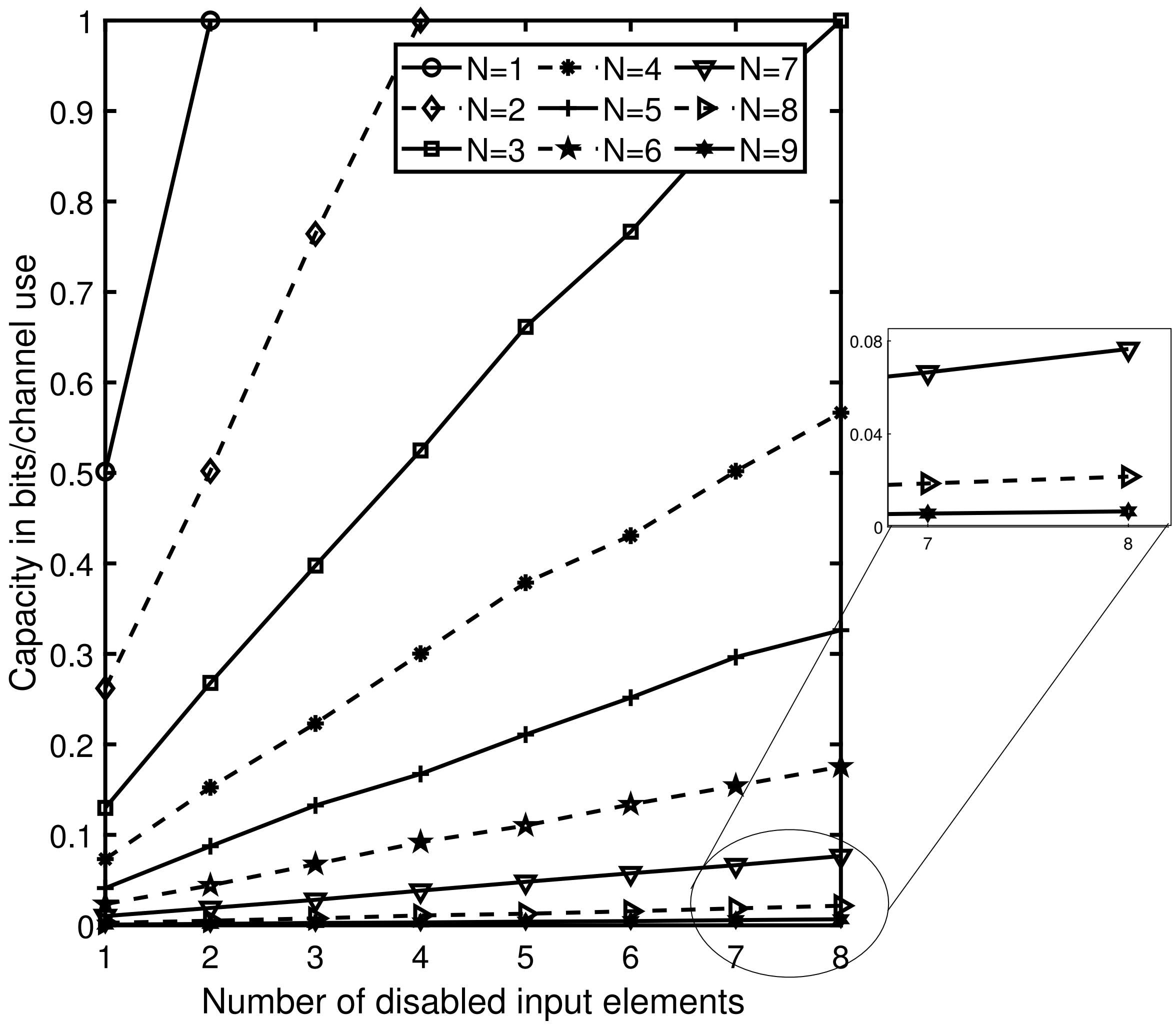


Fig. 3.19: Relative loss in Capacity due to disabled transmit elements for order-10 tensor channel at $P = 0$ dB.

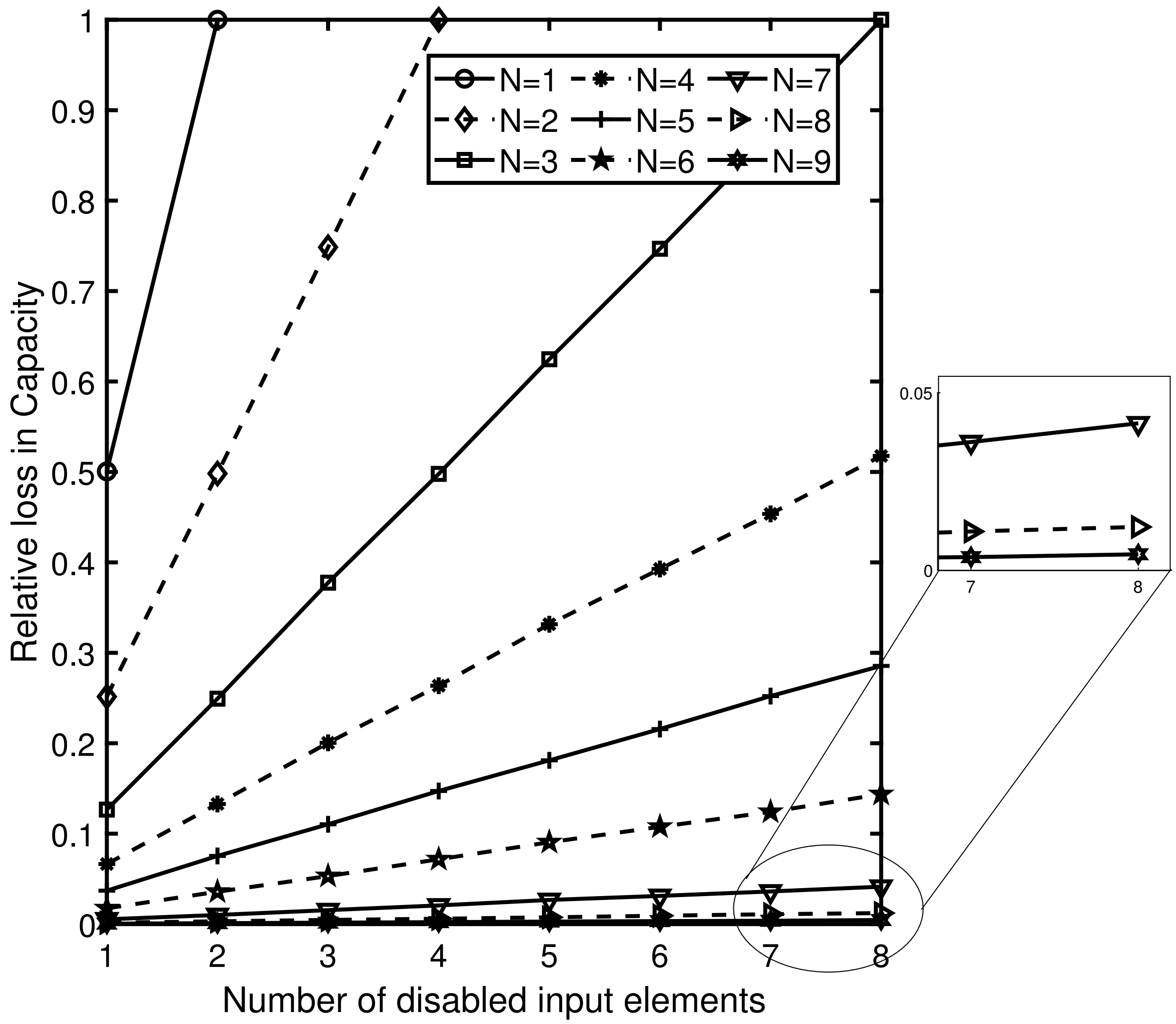


Fig. 3.20: Relative loss in Capacity due to disabled transmit elements for order-10 tensor channel at $P = 10$ dB.

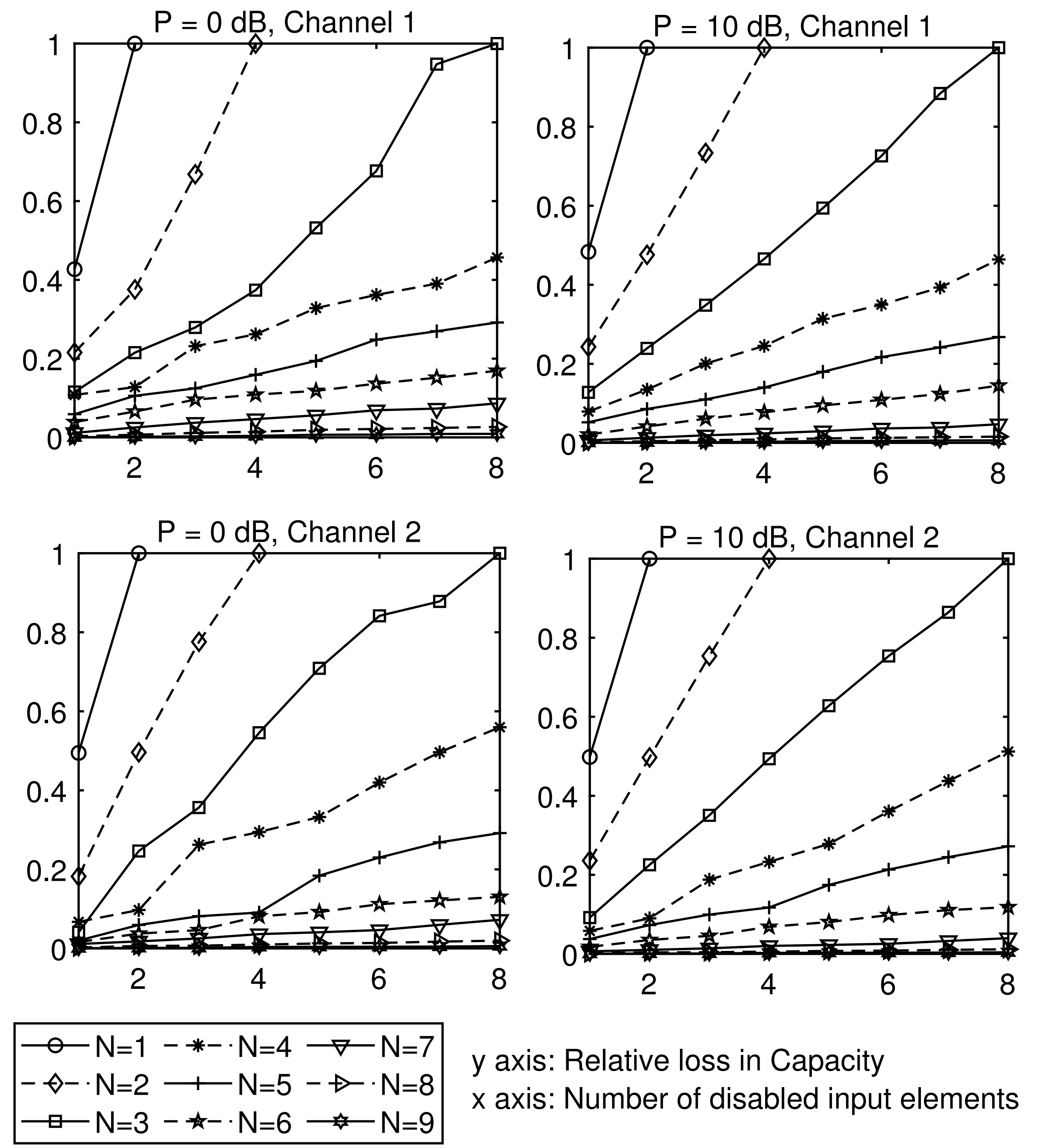


Fig. 3.21: Relative loss in Capacity due to disabled transmit elements for two specific examples of order-10 tensor channel.

Note that for lower P also, averaging over several channel realizations makes the performance closer to uniform power allocation which tends to a linear behavior on average, but this may not be so for each single channel realization. Thus to illustrate the dependence of loss in capacity on P and N, we now take two fixed examples of tensor channel without averaging over several channel realizations. Figure 3.21 presents the loss in capacity with n for two specific realizations of order-10 tensor channel. The specific channels $\mathcal{H}^{(1)}, \mathcal{H}^{(2)}$ are realizations of zero mean unit variance circularly symmetric Gaussian distribution. For normalization of these specific channels, we set $\mathcal{H}^{norm} = \frac{\mathcal{H}}{||\mathcal{H}||}\sqrt{I}$ and I represents the total number of input elements. Please refer to Appendix C for information about the exact channel tensors required to reproduce the results in Figure 3.21. As can be seen in Figure 3.21, the loss in capacity at $P = 0$ dB increases with n but is not linear. However, at $P = 10$ dB, the loss in capacity tends to increase linearly with n because with increasing P, the power tends to get almost uniformly distributed among the input elements. In any case the loss diminishes with increasing N demonstrating the robustness of large input systems.

Capacity Against Channel Order

We now consider the capacity against increasing channel order. For even order ($2N$) channels we take the input and output order as N, and for odd order ($2N + 1$) channels we consider two cases : 1) input N, output $N + 1$ and 2) input $N + 1$, output N. We will also compare these cases with the capacity corresponding to best input output order from Table 3.3.

Capacity for normalized tensor channels at three different power levels is presented against increasing channel order in Figure 3.22. The solid lines represent case 1 and dashed lines represent case 2. For even order channels, the two cases coincide. From Figure 3.22 it can be seen that with increasing channel order the capacity may not necessarily increase monotonically. The specifics of the input and output order play a role. For instance, from order-5 to order-6 channel, the capacity increases if the order-5 corresponds to output order-3 and input order-2, whereas the capacity decreases if the order-5 corresponds to output

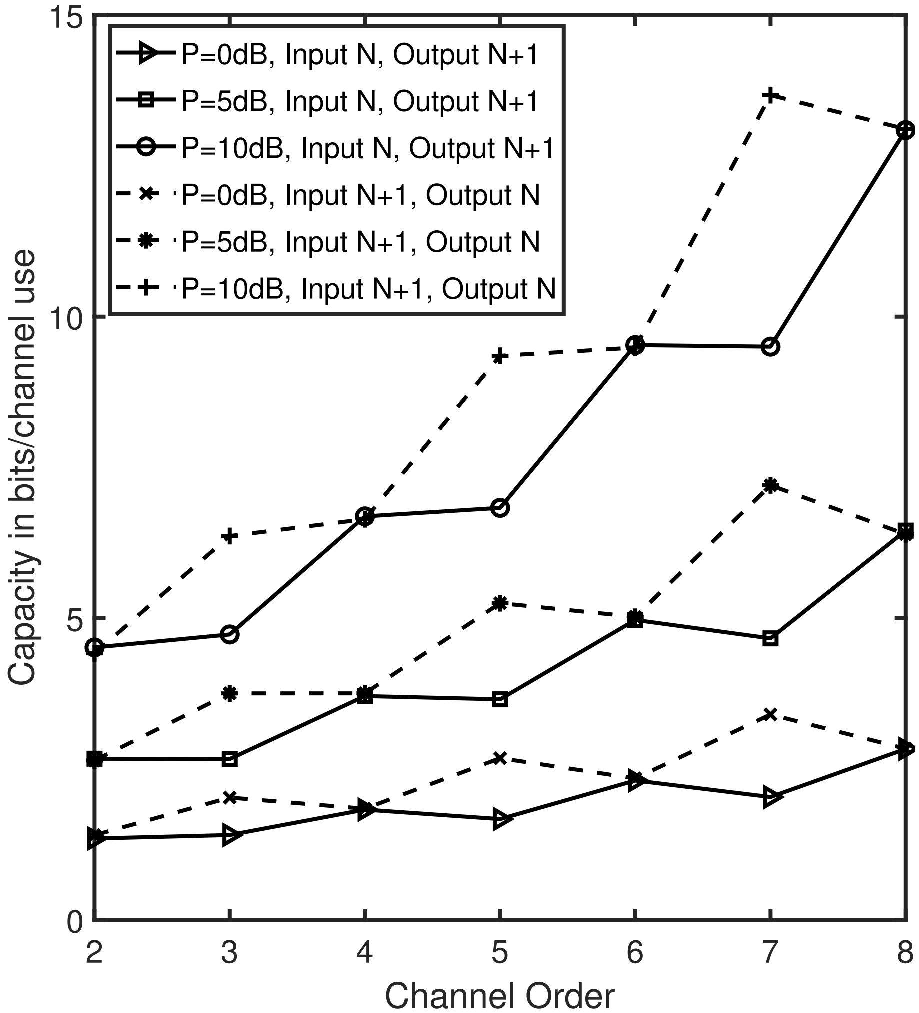


Fig. 3.22: Capacity vs Channel order for normalized Gaussian Tensor Channels at $P = 0, 5, 10$ dB.

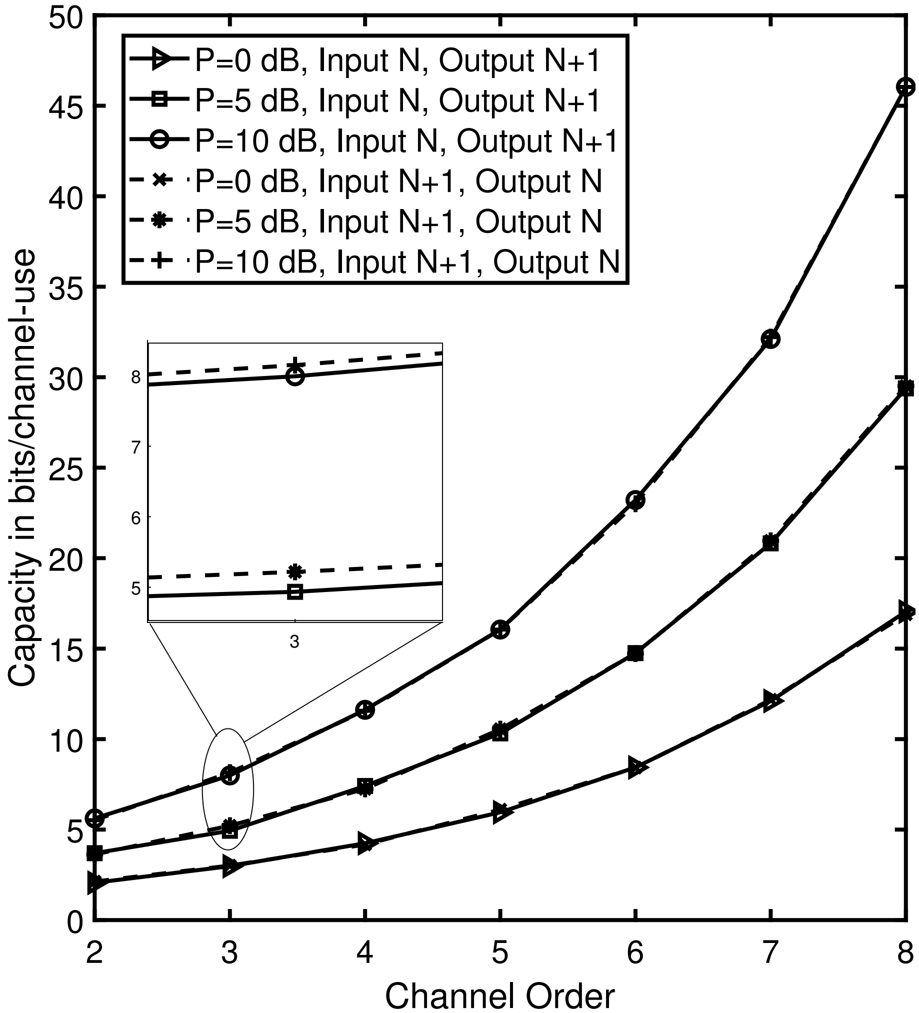


Fig. 3.23: Capacity vs Channel Order for Gaussian Tensor Channels with zero mean and unit variance elements.

order-2 and input order-3 for $P = 5$ and 10 dB. However, note that from Table 3.3, we see that the capacity can increase for any P with increasing channel order if the corresponding input and output order are suitably selected. The difference in the two values of capacity for odd order channels in Figure 3.22 is because of the different ways in which the channel elements are generated. Note that for both the cases (order N input, order $N+1$ output or order $N+1$ input, order N output), the number of decomposed parallel scalar channels will be similar. But if the output order is large, then the channel normalization ensures weaker channel elements thus each equivalent parallel scalar channel is weaker. However, if the input order is large, then the channel normalization ensures stronger channel elements which makes each parallel scalar channel stronger. This behavior of capacity is not observed if the channel is not normalized. This is illustrated in Figure 3.23 where capacity is plotted against increasing input order for channel with zero mean unit variance elements. In such a case, moving from a lower order to higher order channel increases the size of the channel without diminishing the strength of the existing components. Thus the channel provides power gain, and the capacity smoothly increases with channel order. As can be seen in Figure 3.23, case 1 and case 2 do not make a significant difference and the curves almost overlap. With a tensor channel having same dimensions of each domain and containing i.i.d. elements, the number of parallel scalar channels (given by $R = \min\{I, J\}$ where I, J denote the number of transmit and receive elements respectively) remains the same if the input is order $N+1$ and output is order N or if the input is order N and output is order $N+1$. Also, since the channel elements are generated with unit variance, thus changing the input output configuration does not change the strength of the channel components as well, unlike the normalized channel case. Hence case 1 and case 2 overlap in non-normalized channels as opposed to the normalized channel case where the solid curves and dashed curves (Figure 3.22) are distinct for odd order channels. However, it should be noted that even for normalized tensor channels, with increasing power level P, the capacity shows an increasing behavior with channel order irrespective of the input being $N+1$ or output being $N+1$. This is illustrated in Figure 3.24 where the capacity is plotted against channel order for very high values of P. While there is still significant difference in the capacity of the odd order channels for case 1 and case 2, the capacity always increases with the channel order itself. For instance, if we compare Figure 3.24 and 3.22, we can see that in Figure 3.22 capacity of order-8 channel is lower than capacity of order-7 channel when input is order-4 and output is order-3. However, in Figure 3.24, the capacity for order-8 is larger than either of the two input output configurations of order-7 channel. Thus the effect of channel normalization which weakens the individual components with increasing output order can be compensated for by increasing the transmitted signal power so as to ensure sufficient power allocation even to the weaker channels with water-filling.

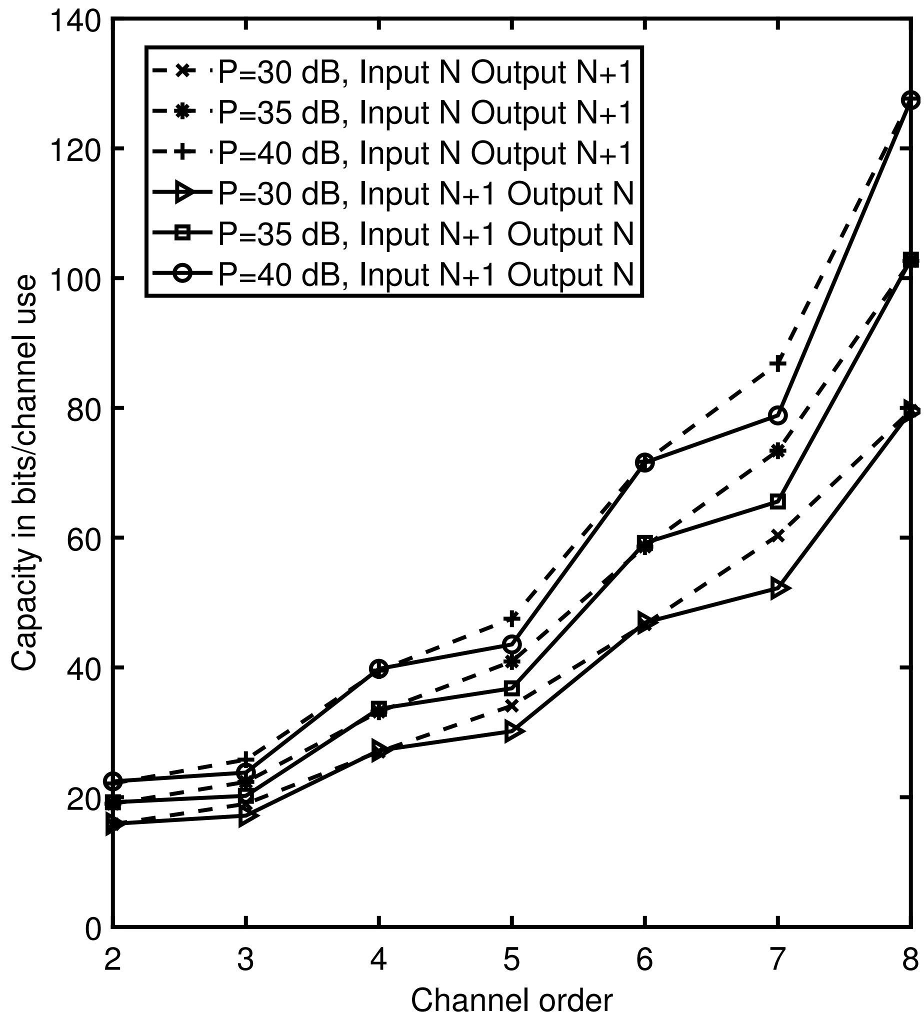


Fig. 3.24: Capacity vs Channel order for normalized Gaussian Tensor Channels at $P = 30, 35, 40$ dB.

3.3.3 MIMO GFDM

For MIMO GFDM, consider the system model from (2.58), where the channel is represented as an order 6 tensor of size $S \times K \times M \times S \times K \times M$. The model is also illustrated in Figure 2.7. Let the number of transmit and receive antennas be N_T and N_R respectively, with $N_T = N_R = S$. We analyze the capacity for different number of data streams, sub-carriers and sub-symbols, denoted by S, K, and M respectively, to explore the trade-off between these domains. The channel is generated as a cascade of transmit filter, physical channel and receive filter. In this example, we use a Raised Cosine (RC) transmit pulse shaping filter with roll off factor 1 at the transmitter. The receive filter is matched to the transmit filter and the elements of the physical channel are generated using i.i.d. complex Gaussian with zero mean and unit variance. Furthermore, the entries of the equivalent channel $\mathcal{H}$ are normalized to ensure that the average received power is same as the transmit power P. The noise tensor $\boldsymbol{\mathcal{N}}$ contains zero mean and unit variance circularly symmetric Gaussian entries such that noise covariance is $N_0\mathcal{I}$ with $N_0 = 1$. With channel gain normalized to one, the signal to noise ratio is defined as $SNR = (N_T E_s)/N_0$ [25] where E_s is the transmit energy per element defined as $P/(N_T KM)$. The tensor framework gives us the capacity in bits/channel-use, where in each transmission a tensor symbol contains elements across all the sub-carriers, sub-symbols and antennas. Hence we normalize the capacity of the MIMO GFDM channel by the number of sub-carriers and sub-symbols as in [25]. Figure 3.25 shows the normalized capacity against SNR for different values of S, K, and M. It can be seen that for a fixed value of KM, as S increases we get higher capacity. Also, for a fixed S, choosing different configurations of K and M, such that the product KM is constant, leads to almost similar capacity results. The capacity when $K = 4, M = 10$ is almost the same as the capacity with $K = 8, M = 5$, and slightly lower than with $K = 2, M = 20$. This exhibits the latent trade-off between the sub-carrier and sub-symbol domains, which can be harnessed using the tensor framework.

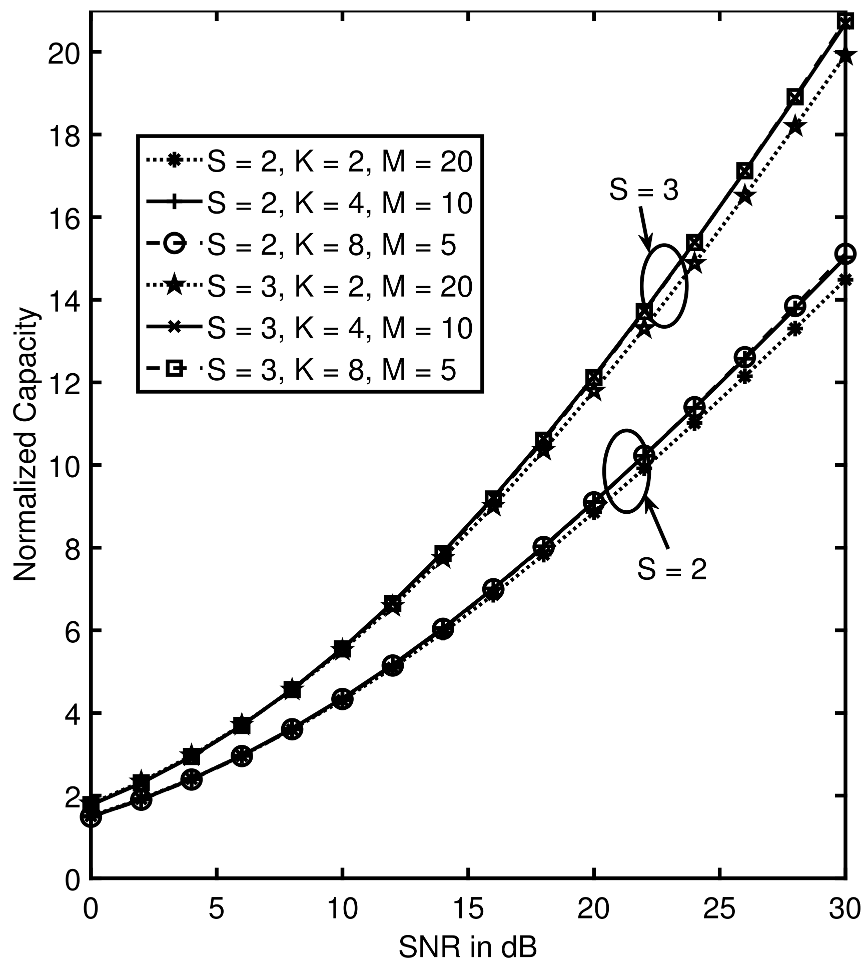


Fig. 3.25: Capacity vs SNR for MIMO GFDM with different S, K, M.

Effect of Pulse Shaping Parameters on the Channel Capacity

Notice that the equivalent channel for MIMO GFDM is generated using a cascade of the transmit filter, physical channel and receive filter as explained in section 2.3.3. Using the tensor framework, we can analyze the capacity behaviour of the channel corresponding to different pulse shaping parameters as well. We simulate a system with $S = N_R = N_T = 2, K = 8$ and $M = 5$. The transmit filter tensor $\mathcal{H}_T$ is generated using different pulse shapes, namely RC, Root Raised Cosine (RRC) and Dirichlet. The roll-off factor α associated with RC and RRC pulse describes the overlap of the sub-carriers in the frequency domain. The Dirichlet pulse is defined by a rectangular function in the frequency domain with width of M frequency bins that are located around the DC bin [169] or in time domain as $g_M[n] = (\sin(Mn/2))/(M\sin(n/2))$. Figure 3.26 shows the normalized capacity of the tensor channel under sum power constraint for all the three transmit filters with different α for RC and RRC.

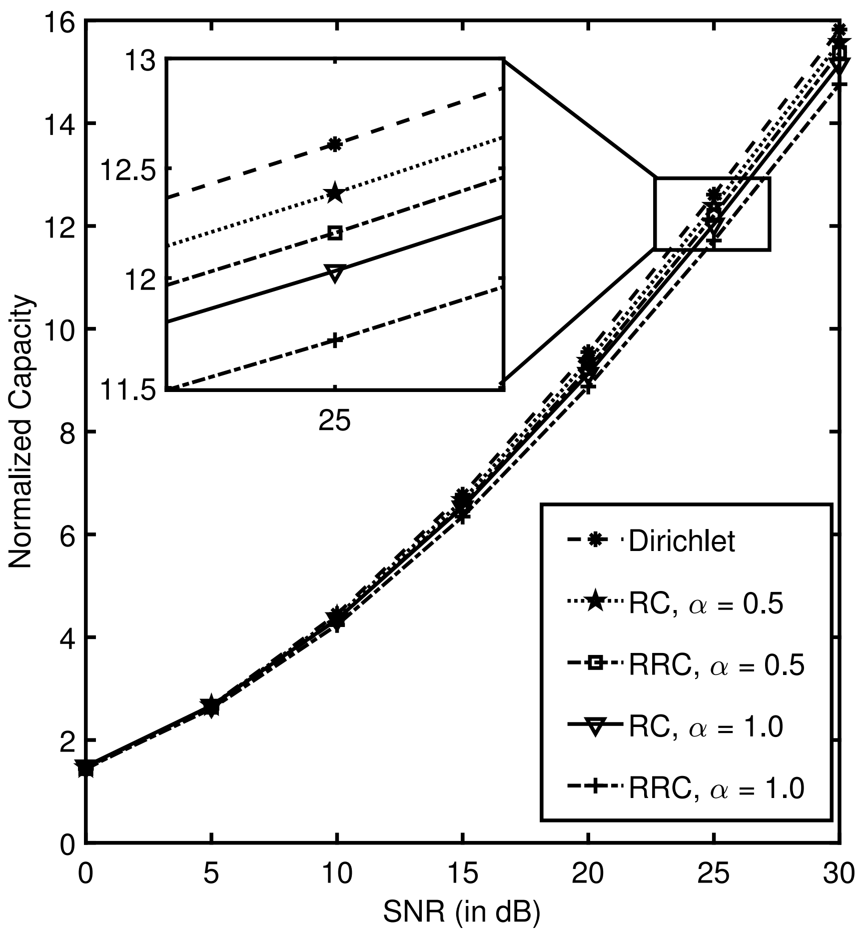


Fig. 3.26: Capacity vs SNR for MIMO GFDM with different pulse shaping filters.

In Figure 3.26, the difference in capacity for various filters at low SNRs is minimal. At high SNR, it can be observed that Dirichlet achieves highest capacity. With RC and RRC, higher roll-off factor of the pulse shape leads to a lower capacity. Since higher values of α denote higher overlap between sub-carriers within a tensor symbol, increasing α increases the intra-tensor interference. When $\alpha = 0$, both RC and RRC reduce to a sinc pulse, and hence are identical. With increasing α, RC gives higher capacity than RRC as for a given α, RC has sharper edges in frequency domain, hence creates less self-interference. Dirichlet pulse shaping filter shows highest capacity as it does not create any self-interference. In fact employing Dirichlet pulse makes the GFDM system orthogonal [169].

3.4 Tensor Multi-user Channel Capacity

In this section, we consider the application of the tensor framework to Multi-User (MU) systems where each user is equipped with multiple antennas. We present numerical examples comparing the tensor approach with other results known in literature for K-user Gaussian multiple access channels and interference channels.

3.4.1 Multiple Access Channels

Consider a MU MIMO network where a Base Station (BS) equipped with N_R antennas is receiving information from K users with N_T antennas each. Let the uplink channel matrix between the kth user and the base station be denoted by $\mathrm{H}^{(k)} \in \mathbb{C}^{N_R \times N_T}$. The discrete time received signal $\underline{\mathbf{y}} \in \mathbb{C}^{N_R \times 1}$ at the BS is given as [170] :

$$\underline{\mathbf{y}} = \sum_{k=1}^{K} \mathrm{H}^{(k)} \underline{\mathbf{x}}^{(k)} + \underline{\mathbf{n}} \tag{3.79}$$

where $\underline{\mathbf{x}}^{(k)} \in \mathbb{C}^{N_T \times 1}$ is the signal transmitted by the kth user and $\underline{\mathbf{n}} \in \mathbb{C}^{N_R \times 1}$ is the received noise vector which is assumed circularly symmetric complex Gaussian with identity covariance matrix. In such a system, each user k is subject to an individual power constraint P_k. If the transmit covariance matrix of user k is denoted as $\mathrm{Q}^{(k)}$, then the power constraint can be expressed as $\mathrm{tr}(\mathrm{Q}^{(k)}) \leq P_k$ for $k = 1, \ldots, K$.

A multi-user channel with K users is characterized by a K-dimensional achievable rate region $\mathcal{C}_{\mathcal{R}}$, known as the capacity region [170], where each point in the region $(R_1, R_2, \ldots, R_K)$ represents the achievable rates R_k at which user k can send information with arbitrarily low error probability. We assume that all the channel matrices are known to the receiver and all the transmitters. We denote the convex hull of the union of sets using the symbol

$\bar{\bigcup}$. With power constraints $(P_1, P_2, \ldots, P_K)$, the capacity region of MIMO Multiple Access Channels (MAC) is expressed as [170] :

$$\mathcal{C}_{\mathcal{R}} = \bar{\bigcup}_{\mathrm{tr}(\mathrm{Q}^{(k)}) \leq P_k, \forall k} \Big\{ (R_1, \ldots, R_K) : 0 \leq \sum_{k \in \mathcal{S}} R_k \leq \log\det\Big(\mathrm{I} + \sum_{k \in \mathcal{S}} \mathrm{H}^{(k)}\mathrm{Q}^{(k)}\mathrm{H}^{(k)H}\Big) \\ \forall \mathcal{S} \subseteq \{1, \ldots, K\} \Big\} \quad (3.80)$$

where $\mathcal{S}$ denotes a subset of the set of users. Each set of covariance matrices $(\mathrm{Q}^{(1)}, \ldots, \mathrm{Q}^{(K)})$ satisfying the power constraints corresponds to a K-dimensional polyhedron [97]. The capacity region is the convex hull of the union of all such polyhedrons. For Gaussian MIMO MAC, the capacity region can be defined using only the union of rate regions and the convex hull is not needed [170]. It is shown in [171] that for Gaussian MIMO MAC, the boundary points of the capacity region can be characterized by maximizing a weighted sum rate $\sum_k \nu_k R_k$ for all non-negative ν_k such that $\sum_k \nu_k = 1$, and thus finding boundary points can be cast into a convex optimization problem.

Essentially, (3.80) represents a set of bounds on individual rates $R_1, R_2, \ldots, R_K$, and combination of rates such as $R_1 + R_2, R_1 + R_3, R_2 + R_3, R_1 + R_2 + R_3$ and so on, including the sum rate $R_1 + R_2 + \cdots + R_K$. For a two users case, this can be represented as :

$$\mathcal{C}_{\mathcal{R}} = \bar{\bigcup}_{\substack{\mathrm{tr}(\mathrm{Q}^{(1)}) \leq P_1, \\ \mathrm{tr}(\mathrm{Q}^{(2)}) \leq P_2}} \left\{ \begin{array}{c} 0 \leq R_1 \leq \log\det\Big(\mathrm{I} + \mathrm{H}^{(1)}\mathrm{Q}^{(1)}\mathrm{H}^{(1)H}\Big) \\ 0 \leq R_2 \leq \log\det\Big(\mathrm{I} + \mathrm{H}^{(2)}\mathrm{Q}^{(2)}\mathrm{H}^{(2)H}\Big) \\ 0 \leq R_1 + R_2 \leq \log\det\Big(\mathrm{I} + \mathrm{H}^{(1)}\mathrm{Q}^{(1)}\mathrm{H}^{(1)H} + \mathrm{H}^{(2)}\mathrm{Q}^{(2)}\mathrm{H}^{(2)H}\Big) \end{array} \right\} \quad (3.81)$$

For a given choice of covariance matrices $\mathrm{Q}^{(1)}$ and $\mathrm{Q}^{(2)}$ which satisfy the power constraints, (3.81) represents an upper bound on R_1, R_2 and $R_1 + R_2$. The maximum of $\log\det(\mathrm{I} + \mathrm{H}^{(1)}\mathrm{Q}^{(1)}\mathrm{H}^{(1)H})$ and $\log\det(\mathrm{I} + \mathrm{H}^{(2)}\mathrm{Q}^{(2)}\mathrm{H}^{(2)H})$ from (3.81) are the individual achievable capacities by user 1 and user 2 respectively assuming the other user is silent. The maximum of $\log\det(\mathrm{I} + \mathrm{H}^{(1)}\mathrm{Q}^{(1)}\mathrm{H}^{(1)H} + \mathrm{H}^{(2)}\mathrm{Q}^{(2)}\mathrm{H}^{(2)H})$ is the sum capacity achievable when both users are transmitting. Note that the choice of covariance matrices $\mathrm{Q}^{(1)}$ and $\mathrm{Q}^{(2)}$ satisfying the power constraints which achieves the sum capacity may not achieve the individual capacities. Similarly the choice of $\mathrm{Q}^{(1)}$ and $\mathrm{Q}^{(2)}$ which achieves the individual capacities may not achieve the sum capacity. In general, different transmit choices lead to different pairs of covariance matrices $(\mathrm{Q}^{(1)}, \mathrm{Q}^{(2)})$ such that the capacity region is given by the convex hull of the union of an infinite number of rate regions each corresponding to a different set $(\mathrm{Q}^{(1)}, \mathrm{Q}^{(2)})$. The optimal choice of $\mathrm{Q}^{(1)}$ and $\mathrm{Q}^{(2)}$ which achieves the sum capacity is found by an iterative water-filling approach [171] which sequentially finds covariance for each user with the single-user classical water-filling method assuming interference from other users as noise. A detailed step by step algorithm can be found in [171].

However, such an approach assumes that different users transmit independently. The iterative water-filling algorithm treats the information available about other users' interfering channels as noise. In the presence of complete channel state information, a better transmit strategy which can provide higher achievable rates would be to allow users to coordinate for transmission. Hence a joint signal transmit strategy can expand the capacity region. Using the tensor framework, we can find the capacity region assuming user coordination. First, let us express (3.79) using the tensor system model. The multi-user MIMO tensor channel can be defined as a third order tensor $\mathcal{H} \in \mathbb{C}^{N_R \times N_T \times K}$ where $\mathcal{H}_{:,:,k} = \mathrm{H}^{(k)}$. The input signal can be denoted using a matrix $\mathbf{X} \in \mathbb{C}^{N_T \times K}$, where each $\underline{\mathbf{x}}^{(k)}$ of (3.79) forms a column of the matrix $\mathbf{X}$. Hence the system model in (3.79) can be represented as :

$$\underline{\mathbf{y}} = \mathcal{H} *_2 \mathbf{X} + \underline{\mathbf{n}}. \tag{3.82}$$

The input covariance is represented as an order-4 tensor $\mathcal{Q} \in \mathbb{C}^{N_T \times K \times N_T \times K}$. Assuming the noise to be circularly symmetric complex Gaussian with identity covariance matrix, denoted by I, the output covariance can be written as $(\mathcal{H} *_2 \mathcal{Q} *_2 \mathcal{H}^H + \mathrm{I})$. Subsequently, the sum capacity of such a system with user coordination can be calculated from the following optimization problem :

$$\max_{\mathcal{Q}} \quad \log\left[\det\left(\mathcal{H} *_2 \mathcal{Q} *_2 \mathcal{H}^H + \mathrm{I}\right)\right] \tag{3.83}$$

$$s.t. \quad \sum_{n=1}^{N_T} \mathcal{Q}_{n,k,n,k} \leq P_k \quad \forall k, \tag{3.84}$$

$$\mathcal{Q} \succeq 0. \tag{3.85}$$

where $\sum_{n=1}^{N_T} \mathcal{Q}_{n,k,n,k} \leq P_k$ represents the individual power constraints for different users. The optimal $\mathcal{Q}$ that achieves capacity can be approximated using Algorithm 1. Note the difference in this tensor formulation and the iterative water-filling approach used in vector formulation is that the latter assumes different users transmit independently despite having perfect channel state information. With independent transmissions, the iterative water-filling maximizes the function $\log\det(\mathrm{I}+\sum_{k=1}^{K}\mathrm{H}^{(k)}\mathrm{Q}^{(k)}\mathrm{H}^{(k)H})$ subject to $\mathrm{tr}(\mathrm{Q}^{(k)})\leq P_k$ and $\mathrm{Q}^{(k)}\succeq 0$ for $k=1,\dots,K$ [171]. Note that this objective function is same as the upper bound on the sum rate from (3.80) for $S=\{1,\dots,K\}$. The vector based iterative water-filling treats inter-user interference as noise since it attempts to optimize the sum rate over all choices of separate covariance matrix for each user as shown in [171]. On the other hand, the tensor approach solves the problem in (3.83)-(3.85) and aims to find a joint covariance across all the users. Thereby, the tensor approach suggests a joint transmit scheme for all the users wherein the inter-user interference is not treated as noise since the interference term also carries signal information. The maximum sum rate achieved by all the K users given in (3.83) is the sum capacity of the K users MIMO MAC under user coordination. Similarly the sum capacity achieved by a subset $\mathcal{S}$ of the all the users $\mathcal{U}=\{1,\dots,K\}$ is given as the maximum of $\log\det(\mathrm{I}+\mathcal{H}^{(\mathcal{S})}*_2\mathcal{Q}^{(\mathcal{S})}*_2\mathcal{H}^{(\mathcal{S})H})$ over the choice of positive semi-definite $\mathcal{Q}^{(\mathcal{S})}$ which satisfies the power constraints. The tensor $\mathcal{H}^{(\mathcal{S})}\in\mathbb{C}^{N_R\times N_T\times|\mathcal{S}|}$ with $|\mathcal{S}|$ denoting the cardinality of $\mathcal{S}$, contains matrices of size $N_R\times N_T$ as slices corresponding to only those users which are included in $\mathcal{S}$. Similarly $\mathcal{Q}^{(\mathcal{S})}\in\mathbb{C}^{N_T\times|\mathcal{S}|\times N_T\times|\mathcal{S}|}$ is the covariance tensor of $\mathbf{X}^{(\mathcal{S})}\in\mathbb{C}^{N_T\times|\mathcal{S}|}$ which contains columns of only those users which are included in $\mathcal{S}$.

Hence the tensor framework allows to define a capacity region with user coordination as :

$$\mathcal{C}_{\mathcal{R}}=\bar{\bigcup_{\sum_{n=1}^{N_T}\mathcal{Q}^{(\mathcal{S})}_{n,i,n,i}\leq\underline{\mathrm{p}}_i^{(\mathcal{S})},\forall i}}\Big\{(R_1,\dots,R_K):0\leq\sum_{k\in\mathcal{S}}R_k\leq\log\det\Big(\mathrm{I}+\mathcal{H}^{(\mathcal{S})}*_2\mathcal{Q}^{(\mathcal{S})}*_2\mathcal{H}^{(\mathcal{S})H}\Big)\quad\forall\mathcal{S}\subseteq\{1,\dots,K\}\Big\}\tag{3.86}$$

where $\mathcal{S}$ contains the list of users being considered. The vector $\underline{\mathrm{p}}^{(\mathcal{S})}$ contains the power budgets of the users included in $\mathcal{S}$, with its components $\underline{\mathrm{p}}_i^{(\mathcal{S})}$ denoting the power budget of the ith user in set $\mathcal{S}$ for $i=1,\dots,|\mathcal{S}|$. Note that the expression $\log\det(\mathrm{I}+\mathcal{H}^{(\mathcal{S})}*_2\mathcal{Q}^{(\mathcal{S})}*_2\mathcal{H}^{(\mathcal{S})H})$ in (3.86) is same as the objective function in (3.83) when $\mathcal{S}$ contains all the users, i.e. $\mathcal{S} = \{1, \dots, K\}$.

As an example, let us consider a three users scenario for which $\mathcal{S}$ can assume the following sets :$\{1\}, \{2\}, \{3\}, \{1,2\}, \{2,3\}, \{1,3\}, \{1,2,3\}$. Subsequently, we can express the capacity region from (3.86) as :

$$\mathcal{C}_{\mathcal{R}} = \bar{\bigcup_{\substack{\mathrm{tr}(\mathrm{Q}^{(i)}) \le P_i,\ i=1,2,3 \\ \sum_n \mathcal{Q}^{(1,2)}_{n,1,n,1} \le P_1,\ \sum_n \mathcal{Q}^{(1,2)}_{n,2,n,2} \le P_2 \\ \sum_n \mathcal{Q}^{(2,3)}_{n,1,n,1} \le P_2,\ \sum_n \mathcal{Q}^{(2,3)}_{n,2,n,2} \le P_3 \\ \sum_n \mathcal{Q}^{(1,3)}_{n,1,n,1} \le P_1,\ \sum_n \mathcal{Q}^{(1,3)}_{n,2,n,2} \le P_3 \\ \sum_n \mathcal{Q}_{n,k,n,k} \le P_k,\ k=1,2,3}}} \left\{ \begin{array}{c} 0 \le R_1 \le \log\det\left(\mathrm{I} + \mathrm{H}^{(1)}\mathrm{Q}^{(1)}\mathrm{H}^{(1)H}\right) \\ 0 \le R_2 \le \log\det\left(\mathrm{I} + \mathrm{H}^{(2)}\mathrm{Q}^{(2)}\mathrm{H}^{(2)H}\right) \\ 0 \le R_3 \le \log\det\left(\mathrm{I} + \mathrm{H}^{(3)}\mathrm{Q}^{(3)}\mathrm{H}^{(3)H}\right) \\ 0 \le R_1 + R_2 \le \log\det\left(\mathrm{I} + \mathcal{H}^{(1,2)} *_2 \mathcal{Q}^{(1,2)} *_2 \mathcal{H}^{(1,2)H}\right) \\ 0 \le R_2 + R_3 \le \log\det\left(\mathrm{I} + \mathcal{H}^{(2,3)} *_2 \mathcal{Q}^{(2,3)} *_2 \mathcal{H}^{(2,3)H}\right) \\ 0 \le R_1 + R_3 \le \log\det\left(\mathrm{I} + \mathcal{H}^{(1,3)} *_2 \mathcal{Q}^{(1,3)} *_2 \mathcal{H}^{(1,3)H}\right) \\ 0 \le R_1 + R_2 + R_3 \le \log\det\left(\mathrm{I} + \mathcal{H} *_2 \mathcal{Q} *_2 \mathcal{H}^{H}\right) \end{array} \right\} \tag{3.87}$$

For the first bound on R_1, we have $\mathcal{S} = \{1\}$, i.e. we consider only first user and find what is the maximum rate that user 1 can transmit given the power constraint on user 1 and that all other users are silent. In this case, $\mathcal{H}^{(1)} \in \mathbb{C}^{N_R \times N_T \times 1}$ is essentially the matrix $\mathrm{H}^{(1)}$ between the user 1 and base station, and $\mathcal{Q}^{(1)} \in \mathbb{C}^{N_T \times 1 \times N_T \times 1}$ is the covariance matrix $\mathrm{Q}^{(1)}$ of user 1. Hence this reduces to single user MIMO channel where the optimal $\mathrm{Q}^{(1)}$ which achieves maximum rate can be found using classical water-filling. Similarly, conditions two and three in (3.87) correspond to the bounds on the rates achieved by user 2 and user 3 (R_2 and R_3) respectively when all other users are silent. Hence, the first three conditions are same as the one derived from the vector case (3.80). Condition four corresponds to $\mathcal{S} = \{1,2\}$, thus gives a bound on the sum rate that user 1 and user 2 can together achieve given that user 3 is silent. In this case $\mathcal{H}^{(1,2)} \in \mathbb{C}^{N_R \times N_T \times 2}$ is a sub-tensor of $\mathcal{H}$ as $\mathcal{H}^{(1,2)} = \mathcal{H}_{:,:,1:2}$. The covariance tensor $\mathcal{Q}^{(1,2)} \in \mathbb{C}^{N_T \times 2 \times N_T \times 2}$ is a sub-tensor of $\mathcal{Q}$ given by $\mathcal{Q}^{(1,2)} = \mathcal{Q}_{:,1:2,:,1:2}$. The power constraints are defined as $\sum_n \mathcal{Q}^{(1,2)}_{n,1,n,1} \le P_1$ and $\sum_n \mathcal{Q}^{(1,2)}_{n,2,n,2} \le P_2$ where P_1, P_2 are power budgets for user 1 and 2 respectively. Note that the bound on sum rate $R_1 + R_2$ achieved using this method assumes that user 1 and 2 perform a joint transmission. Hence the sum rate achieved using the tensor approach will be different than the one obtained from iterative water-filling which assumes independent transmission. Similarly, condition five represents the bound on sum rate $R_2 + R_3$ that can be achieved when user 2 and 3 transmit together keeping user 1 silent. In this case $\mathcal{H}^{(2,3)} \in \mathbb{C}^{N_R \times N_T \times 2}$ is a sub-tensor of $\mathcal{H}$ given by $\mathcal{H}^{(2,3)} = \mathcal{H}_{:,:,2:3}$. The covariance tensor $\mathcal{Q}^{(2,3)} \in \mathbb{C}^{N_T \times 2 \times N_T \times 2}$ is a sub-tensor of $\mathcal{Q}$ given by $\mathcal{Q}^{(2,3)} = \mathcal{Q}_{:,2:3,:,2:3}$. The power constraints are defined as $\sum_n \mathcal{Q}^{(2,3)}_{n,1,n,1} \leq P_2$ and $\sum_n \mathcal{Q}^{(2,3)}_{n,2,n,2} \leq P_3$ where P_2, P_3 are power budgets for user 2 and 3 respectively. Similarly condition six represents the bound on sum rate $R_1 + R_3$ that can be achieved when user 1 and 3 transmit together keeping user 2 silent. The last condition represents the bound on sum rate when all the three users are transmitting. Note that the covariance tensors satisfying the power constraints of all the equations in (3.87) can be seen as sub-tensors of the covariance tensor $\mathcal{Q}$. For instance, $\mathcal{Q}_{:,1,:,1} = \mathrm{Q}^{(1)}$ represents the covariance matrix of user 1. But the optimal choice of covariance tensor $\mathcal{Q}$ that achieves the sum capacity under user cooperation can not be obtained from only individual covariance tensors $\mathrm{Q}^{(i)}$ for different users. For instance, the optimal $\mathrm{Q}^{(1)}$ that maximizes $\log\det(\mathrm{I} + \mathrm{H}^{(1)}\mathrm{Q}^{(1)}\mathrm{H}^{(1)H})$ may not be the sub-tensor of the optimal $\mathcal{Q}$ that maximizes $\log\det(\mathrm{I} + \mathcal{H} *_2 \mathcal{Q} *_2 \mathcal{H}^H)$. Thus the capacity region under user coordination is given by the convex hull of the union of all the rate regions over all the choices of covariance tensors which satisfy the power constraints.

We can also use (3.86) to represent the capacity region without user coordination as in (3.80) by assuming the following additional constraints on the covariance tensor:

$$\mathcal{Q}^{(\mathcal{S})}_{n,i,n',i'} = 0, \quad \text{for } i \neq i'. \tag{3.88}$$

To prove this, we consider the expression $\mathcal{H}^{(\mathcal{S})} *_2 \mathcal{Q}^{(\mathcal{S})} *_2 \mathcal{H}^{(\mathcal{S})H}$ from (3.86) for any given $\mathcal{S}$ and show that with (3.88) it reduces to $\sum_{k\in\mathcal{S}} \mathrm{H}^{(k)}\mathrm{Q}^{(k)}\mathrm{H}^{(k)H}$ as in (3.80). We can write :

$$(\mathcal{H}^{(\mathcal{S})} *_2 \mathcal{Q}^{(\mathcal{S})} *_2 \mathcal{H}^{(\mathcal{S})H})_{j,j'} = \sum_{n',i'} (\sum_{n,i} \mathcal{H}^{(\mathcal{S})}_{j,n,i} \mathcal{Q}^{(\mathcal{S})}_{n,i,n',i'}) \mathcal{H}^{(\mathcal{S})H}_{n',i',j'}. \tag{3.89}$$

With (3.88), entries of $\mathcal{Q}^{(\mathcal{S})}$ are zero for all $i \neq i'$. Hence we can write (3.89) as

$$(\mathcal{H}^{(\mathcal{S})} *_2 \mathcal{Q}^{(\mathcal{S})} *_2 \mathcal{H}^{(\mathcal{S})H})_{j,j'} = \sum_{i=1}^{|\mathcal{S}|} \Big(\sum_{n'=1}^{N_T} (\sum_{n=1}^{N_T} \mathcal{H}^{(\mathcal{S})}_{j,n,i} \mathcal{Q}^{(\mathcal{S})}_{n,i,n',i}) \mathcal{H}^{(\mathcal{S})H}_{n',i,j'} \Big). \tag{3.90}$$

Note that each user in the set of all users $\mathcal{U} = \{1, \ldots, K\}$ is known by its index k, i.e.

first user, second user, kth user and so on. The variable i denotes the index of a user in set $\mathcal{S}$ where $\mathcal{S} \subseteq \mathcal{U}$. For instance if $\mathcal{S} = \{3, 4, \ldots, K\}$, then the third user ($k = 3$) is at index $i = 1$ in $\mathcal{S}$. Hence we replace the index i with the user number k. The entities $\mathcal{H}^{(\mathcal{S})}_{:,:,i}$ and $\mathcal{Q}^{(\mathcal{S})}_{:,i,:,i}$ are the channel sub-tensor and covariance matrix of the user i in set $\mathcal{S}$, and equivalently of the kth user in set of all users. Hence we get $\mathcal{H}^{(\mathcal{S})}_{j,n,i} = \mathrm{H}^{(k)}_{j,n}$ for $k \in \mathcal{S}$, and the covariance as $\mathcal{Q}_{n,i,n',i} = \mathrm{Q}^{(k)}_{n,n'}$ for $k \in \mathcal{S}$. Thus we can write (3.90) as :

$$(\mathcal{H}^{(\mathcal{S})} *_2 \mathcal{Q}^{(\mathcal{S})} *_2 \mathcal{H}^{(\mathcal{S})H})_{j,j'} = \sum_{k\in\mathcal{S}} \Big(\sum_{n'=1}^{N_T} (\sum_{n=1}^{N_T} \mathrm{H}^{(k)}_{j,n} \mathrm{Q}^{(k)}_{n,n'}) \mathrm{H}^{(k)H}_{n,j'} \Big) \tag{3.91}$$

$$= \sum_{k\in\mathcal{S}} (\mathrm{H}^{(k)} \mathrm{Q}^{(k)} \mathrm{H}^{(k)H})_{j,j'}. \tag{3.92}$$

Substituting (3.92) with the additional constraints (3.88) into (3.86) gives us the capacity region from (3.80) which assumes no user coordination.

Now we present a few numerical examples to illustrate the concepts. Consider a MU MIMO MAC scenario where K users with $N_T = 2$ antennas each are transmitting to a base station equipped with $N_R = 10$ antennas. The noise vector at the receiver is circularly symmetric complex Gaussian with zero mean and identity covariance matrix. The channel entries are realizations of circularly symmetric complex Gaussian random variables with zero mean and unit variance and these realizations are known at the transmitters and receiver. The results presented are averaged over 100 different channel realizations. Each user has an individual power budget P_k. The total transmit power is $P = \sum_k P_k$. We assume that all the users have the same power constraint, i.e. $P_k = P/K$ and plot the sum capacity obtained through the tensor approach achieved by K users against total power in Figure 3.27. The sum capacity is found by solving the optimization problem in (3.83)-(3.85). We employ the proposed solution from Algorithm 1 to approximate the optimal input covariance tensor which achieves the sum capacity. The covariance tensor obtained from Algorithm 1 is further used to approximate the sum capacity given by $\log\det(\mathcal{H} *_2 \mathcal{Q} *_2 \mathcal{H}^H + \mathrm{I})$. This approach assumes that all the users coordinate for transmission. It can be seen that for a fixed number of users, the sum capacity increases with an increase in the total transmit power. Furthermore, for a fixed total transmit power, the sum capacity increases when the number of users increases. Especially at higher transmit powers, the increased number of users lead to a significant increase in the sum capacity.

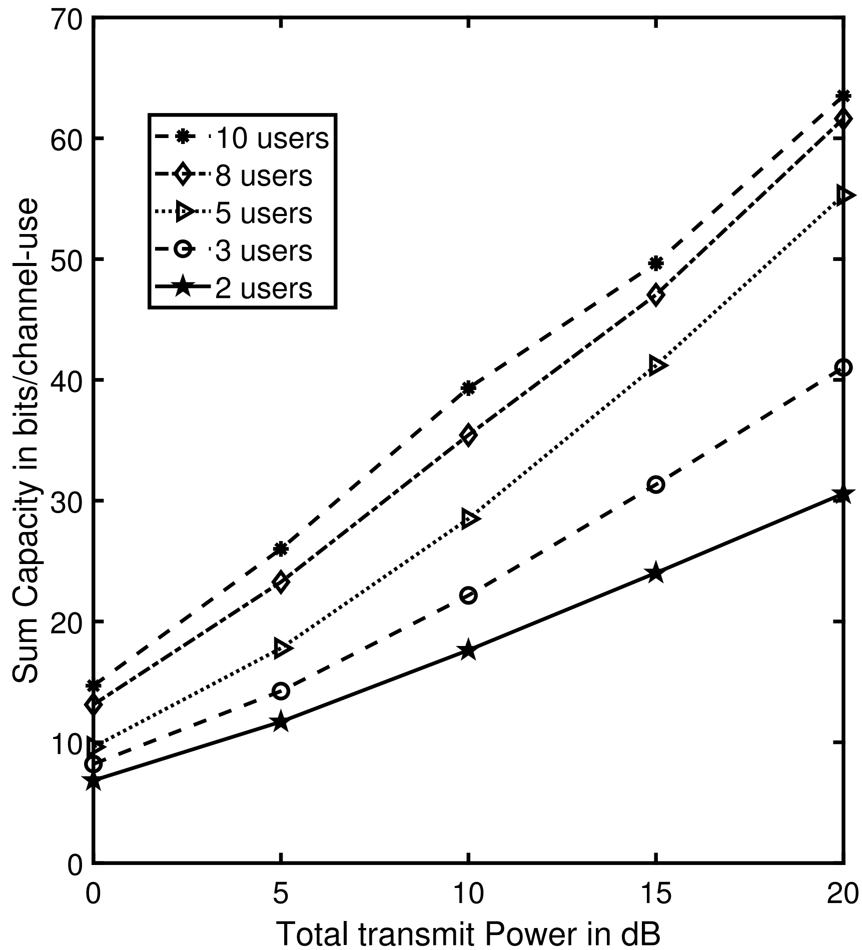


Fig. 3.27: Sum capacity vs transmit power for MU MIMO MAC.

The results of Figure 3.27 can be compared with the iterative water-filling approach where different users despite having channel state information of other users transmit independently. Figure 3.28 presents the sum capacity under coordinated users and independent users against the number of users for two different values of total power. The sum capacity under independent users is calculated using the iterative water-filling approach from [171, Algorithm 1]. It can be observed that as the number of users increases, there is a significant difference in achievable rate of coordinated users as compared to the independent users.

This shows that user cooperation captured in the input covariance tensor structure can increase the sum capacity substantially.

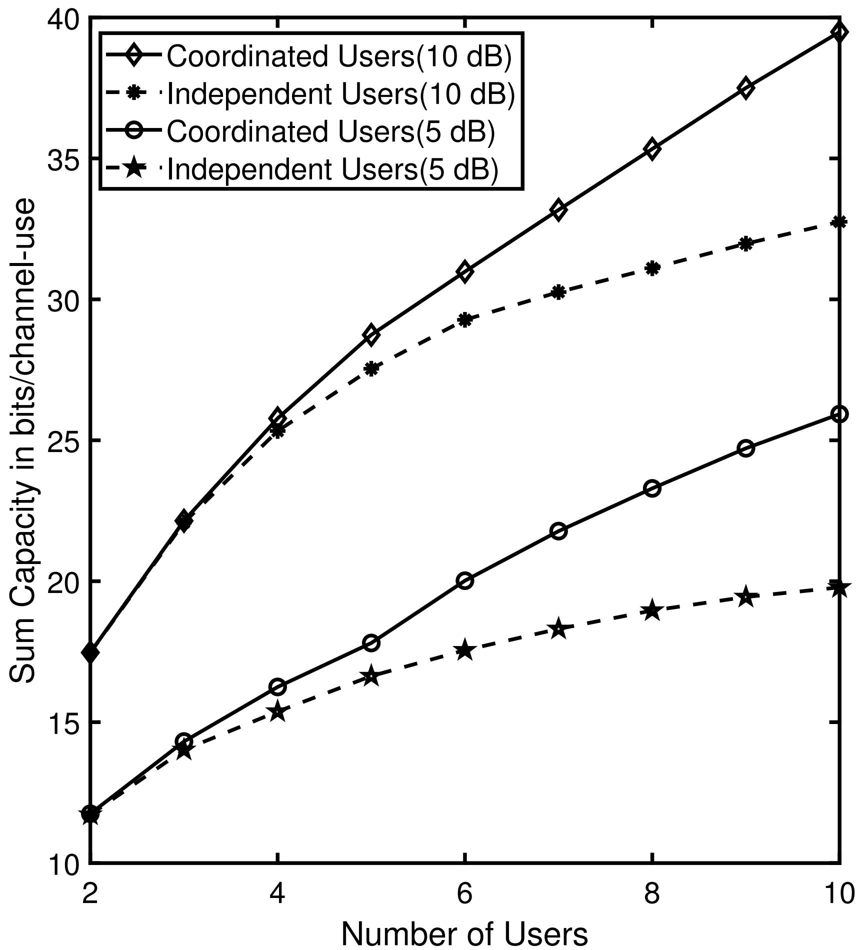


Fig. 3.28: Sum capacity vs users for MU MIMO MAC.

Further the capacity region of a 2 users MIMO MAC is presented in Figure 3.29 for transmit power budgets $P_1 = P_2 = 5$ dB. We assume the base station has $N_R = 8$ antennas, and 2 different settings for transmit antennas $N_T = 4$ and 8. The capacity region obtained from the tensor formulation for two users can be expressed as :

$$\mathcal{C}_{\mathcal{R}} = \bigcup^{-}_{\substack{\mathrm{tr}(\mathrm{Q}^{(1)}) \leq P_1, \\ \mathrm{tr}(\mathrm{Q}^{(2)}) \leq P_2, \\ \sum_{n=1}^{N_T} \mathcal{Q}_{n,k,n,k} \leq P_k, k=1,2.}} \left\{ \begin{array}{c} R_1 \leq \log\det\left(\mathrm{I} + \mathrm{H}^{(1)}\mathrm{Q}^{(1)}\mathrm{H}^{(1)H}\right) \\ R_2 \leq \log\det\left(\mathrm{I} + \mathrm{H}^{(2)}\mathrm{Q}^{(2)}\mathrm{H}^{(2)H}\right) \\ R_1 + R_2 \leq \log\det\left(\mathrm{I} + \mathcal{H} *_2 \mathcal{Q} *_2 \mathcal{H}^H\right) \end{array} \right\} \tag{3.93}$$

For the two users case, finding the optimum covariance that maximizes the achievable rate of user 1 assuming user 2 is silent reduces to a single user MIMO scenario. Hence the bounds on individual rates R_1 and R_2 in (3.93) and (3.81) are the same. Note however that the bound on the sum rate $R_1 + R_2$ differs. With the additional constraint on covariance tensor from (3.88), the bound on the sum rate in (3.93) translates to :

$$R_1 + R_2 \leq \log\det\left(\mathrm{I} + \mathcal{H} *_2 \mathcal{Q} *_2 \mathcal{H}^H\right) \tag{3.94}$$

$$= \log\det\left(\mathrm{I} + \mathrm{H}^{(1)}\mathrm{Q}^{(1)}\mathrm{H}^{(1)H} + \mathrm{H}^{(2)}\mathrm{Q}^{(2)}\mathrm{H}^{(2)H}\right) \tag{3.95}$$

where (3.94) to (3.95) follow from (3.92). Hence, with additional constraint on covariance $\mathcal{Q}$ as defined by (3.88), the sum rate bound in (3.93) reduces to the sum rate bound in (3.81), and thus all the three bounds in (3.93) depend only on $\mathrm{Q}^{(1)}, \mathrm{Q}^{(2)}$. The additional constraint corresponds to the transmit scheme where each user acts independent of the other user. In such a case, the capacity region is characterized by a pair of covariance matrices $(\mathrm{Q}^{(1)}, \mathrm{Q}^{(2)})$ which satisfies the power constraints. In general, the capacity region of the 2 users case corresponding to (3.93) is characterized by a triplet $(\mathrm{Q}^{(1)}, \mathrm{Q}^{(2)}, \mathcal{Q})$ where $\mathcal{Q}$ is the transmit covariance tensor which prescribes a joint transmission scheme. The transmit covariance matrix of individual users $\mathrm{Q}^{(1)}, \mathrm{Q}^{(2)}$ form the sub-tensors of $\mathcal{Q}$. However the optimal $\mathrm{Q}^{(1)}, \mathrm{Q}^{(2)}$ which maximizes R_1, R_2 respectively may not be the sub-tensors of the optimal $\mathcal{Q}$ which maximizes $R_1 + R_2$. The optimal $\mathrm{Q}^{(1)}, \mathrm{Q}^{(2)}$ are found assuming the other user to be silent for transmission, whereas the optimal $\mathcal{Q}$ is found assuming joint transmission by both the users.

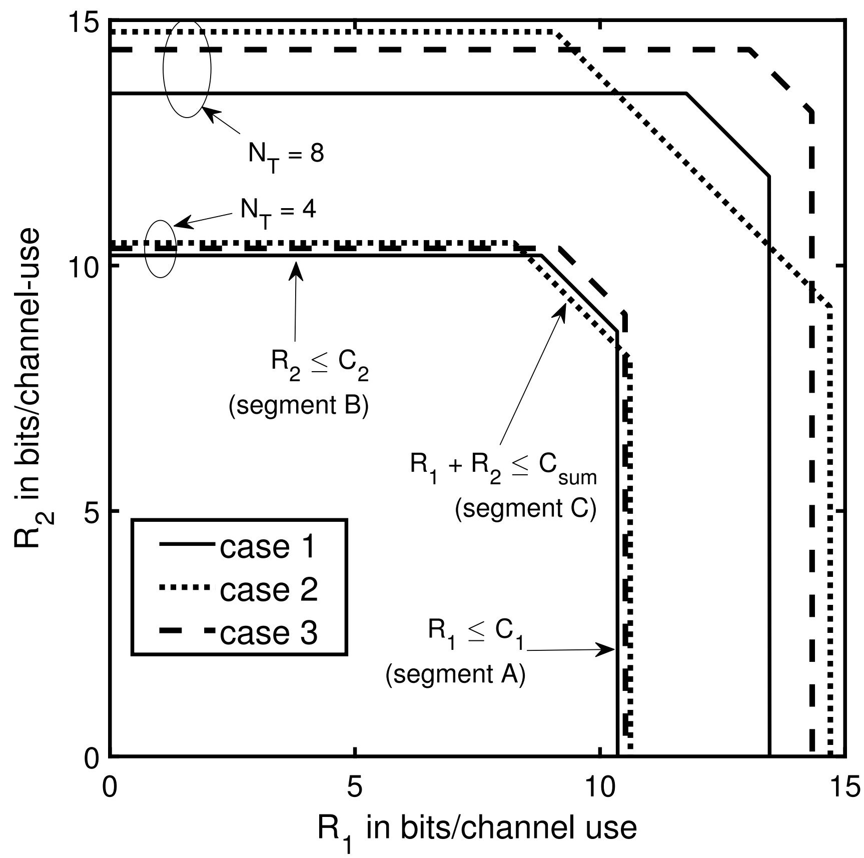


Fig. 3.29: Capacity region of a 2 users MIMO MAC.

The rate regions in (3.93) and (3.81) forms a pentagon on a two dimensional R_1, R_2 plane. The capacity region is determined by the convex hull of the union of all such pentagons obtained through different choices of covariances which satisfy the constraints. In Figure 3.29, the solid line (case 1) represents the pentagon corresponding to (3.81) where $(\mathrm{Q}^{(1)}, \mathrm{Q}^{(2)})$ are obtained via iterative water-filling from [171, Algorithm 1] to maximize the sum rate with independent transmissions. The dotted line (case 2) represents the pentagon corresponding to (3.81) where $(\mathrm{Q}^{(1)}, \mathrm{Q}^{(2)})$ are obtained via conventional water-filling for single user MIMO to maximize R_1 and R_2 individually. Similarly, different $(\mathrm{Q}^{(1)}, \mathrm{Q}^{(2)})$ will correspond to different pentagons based on (3.81). The capacity region with independent users is obtained as a convex hull of the union of all such pentagons associated with different covariances which satisfy the constraints in (3.81). Thus the convex hull of the regions of case 1 and 2 gives an inner bound to the capacity region with independent user transmission. The capacity region signifies the bounds on the rate of transmission by each user.

The dashed line (case 3) represents the pentagon corresponding to (3.93) where $\mathcal{Q}$ is approximated using Algorithm 1 to maximize the sum rate, and $\mathsf{Q}^{(1)}, \mathsf{Q}^{(2)}$ are the sub-tensors of the $\mathcal{Q}$ obtained from Algorithm 1. However such a choice of $\mathsf{Q}^{(1)}$ and $\mathsf{Q}^{(2)}$ may not maximize the individual rates of user 1 and 2. This is reflected in Figure 3.29 as the dotted line shows a larger bound on individual rates as compared to other cases in both the vertical and horizontal segments, A and B. Case 2 can be seen as another scenario for (3.93) where $\mathsf{Q}^{(1)}, \mathsf{Q}^{(2)}$ are chosen to maximize the individual rates and the joint covariance tensor $\mathcal{Q}$ has structure $\mathcal{Q}_{:,i,:,j} = \mathsf{Q}^{(i)}$ for $i = j$ and 0 for $i \neq j$ with $i, j = 1, 2$. Such a covariance tensor does not maximize the sum rate but only individual rates and still satisfies all the constraints in (3.93). The capacity region with user coordination is given by the convex hull of the union of pentagons associated with case 3 and case 2 along with every pentagon associated with different covariances which satisfy the constraints in (3.93). Thus, a convex hull of the regions of case 2 and 3 gives an inner bound to the capacity region with user coordination. The bound on the sum achievable rate indicated by segment C in Figure 3.29 is lowest in case 2, followed by the case where transmit covariances are chosen via iterative water-filling (case 1), and is largest when covariance is obtained using the tensor approach assuming user coordination (case 3). Moreover, this difference increases as N_T increases from 4 to 8, as observed in the figure. Hence it is seen that the boundary of the capacity region expands in segment C with user coordination as opposed to the independent users transmissions.

3.4.2 MIMO Interference Channels

We now consider the MU MIMO Interference Channels (IC) scenario where both the transmit and receive side have user separation. Consider K transmit devices having N_T antennas each that are communicating with their respective K receive devices having N_R antennas each. Such a channel model assumes that all the transmitting devices are communicating with their respective receivers while creating interference to all the other receivers. Finding the exact capacity region of a general K user interference channel is still an ongoing effort [172, 173]. For the 2 users scenario, capacity bounds have been discussed in [174, 175] and references within, under certain assumptions regarding interference such as Z interference channels (one of the two receive users does not experience interference), strong interference, and noisy interference. The expression 'noisy interference' refers to conditions where the sum capacity can be achieved by treating interference as noise [176]. In [176, Theorem 2], a set of conditions involving the channel and transmit covariance matrices are established which are sufficient for interference to be treated as noise in a 2 users MIMO IC. Such conditions ensure that the indirect links are much weaker than the direct links, and power allocation is such that the received power of the message on the indirect link for each user is very low compared to the received power of the message via the direct link. Further, it is shown in [177] that under strong interference, each receiver can jointly decode the signal and the interference to achieve the sum capacity. As an extension of the two users case, [178] derives the conditions under which treating noise as interference can achieve the sum capacity for a K users IC. Most of these works assume no coordination among the source transmitters or the destinations. If different transmitting users coordinate for transmission, and receivers coordinate for reception then interference can be treated as information bearing entity which can be harnessed using the tensor framework.

The system model for a MIMO interference network is given by [179]:

$$\underline{\mathbf{y}}^{(k)} = \mathrm{H}^{(k,k)}\underline{\mathbf{x}}^{(k)} + \sum_{\substack{u \\ u \neq k}} \mathrm{H}^{(k,u)}\underline{\mathbf{x}}^{(u)} + \underline{\mathbf{n}}^{(k)} \tag{3.96}$$

for $k = 1, \ldots, K$ and $\underline{\mathbf{x}}^{(k)} \in \mathbb{C}^{N_T \times 1}$ is the vector transmitted by source k. Also, $\underline{\mathbf{y}}^{(k)}, \underline{\mathbf{n}}^{(k)} \in \mathbb{C}^{N_R \times 1}$ are the received signal and noise vectors at destination k. The matrix $\mathrm{H}^{(k,k)} \in \mathbb{C}^{N_R \times N_T}$ is the direct channel between source k and destination k and $\mathrm{H}^{(k,u)} \in \mathbb{C}^{N_R \times N_T}$ is the cross-channel matrix between source u and destination k. For each transmitting source, there is an individual power constraint defined as $\mathrm{tr}(\mathrm{Q}^{(k)}) \leq P_k$, where $\mathrm{Q}^{(k)} \in \mathbb{C}^{N_T \times N_T}$ is the covariance matrix of vector $\underline{\mathbf{x}}^{(k)}$ and P_k denotes the power budget. Such interference networks can be thought of as a tensor communication link. The input can be represented as a matrix $\mathbf{X} \in \mathbb{C}^{N_T \times K}$ where each $\underline{\mathbf{x}}^{(k)}$ forms a column of the matrix $\mathbf{X}$. Similarly the received signal and noise can be represented using matrices $\mathbf{Y}, \mathbf{N} \in \mathbb{C}^{N_R \times K}$ where each $\underline{\mathbf{y}}^{(k)}$ and $\underline{\mathbf{n}}^{(k)}$ form columns of the matrices $\mathbf{Y}$ and $\mathbf{N}$. The overall channel between such an input and output can be represented as a fourth order tensor $\mathcal{H} \in \mathbb{C}^{N_R \times K \times N_T \times K}$ where $\mathcal{H}_{:,k,:,u} = \mathrm{H}^{(k,u)}$. Subsequently the interference network system model can be represented in tensor form as :

$$\mathbf{Y} = \mathcal{H} *_2 \mathbf{X} + \mathbf{N}. \tag{3.97}$$

Note that (3.97) differs from MIMO MAC specified by (3.82) in the sense that in (3.82) the channel is a third order tensor and thus the output is a vector. On the other hand in (3.97) the channel is a fourth order tensor which accounts for the user separation at the receiver side as well and thus the output is a matrix or an order-2 tensor. The power constraints on the input can be defined in a similar manner as for (3.82). Assuming the noise covariance is an identity tensor $\mathcal{I}$ of size $N_R \times K \times N_R \times K$, the tensor formulation can be used to specify the channel capacity as

$$\max_{\mathcal{Q}} \ \log \det\Big(\mathcal{H} *_2 \mathcal{Q} *_2 \mathcal{H}^H + \mathcal{I}\Big) \tag{3.98}$$

$$s.t. \ \sum_{n=1}^{N_T} \mathcal{Q}_{n,k,n,k} \leq P_k \quad \forall k, \quad \mathcal{Q} \succeq 0. \tag{3.99}$$

Note that capacity obtained from the proposed tensor formulation assumes that all the sources cooperate for transmission and all the destinations cooperate for reception.

Next we consider an example with two users interference channel and compare the sum rate achieved using the tensor framework which assumes user cooperation, with the upper bound on rate suggested in [174] while treating interference as noise. We consider the same example from [174] consisting of a system composed of two transmitters and two receivers, equipped with N_T antennas each. The model introduces a positive scalar $a \geq 0$ to control the interference power :

$$\underline{\mathbf{y}}^{(1)} = \mathrm{H}^{(1,1)}\underline{\mathbf{x}}^{(1)} + a \cdot \mathrm{H}^{(1,2)}\underline{\mathbf{x}}^{(2)} + \underline{\mathbf{n}}^{(1)}, \tag{3.100}$$

$$\underline{\mathbf{y}}^{(2)} = a \cdot \mathrm{H}^{(2,1)}\underline{\mathbf{x}}^{(1)} + \mathrm{H}^{(2,2)}\underline{\mathbf{x}}^{(2)} + \underline{\mathbf{n}}^{(2)}. \tag{3.101}$$

Such a system of equations can be equivalently represented using (3.97) where the channel $\mathcal{H}$ is a tensor of size $N_T \times 2 \times N_T \times 2$ and the input, output and noise are matrices of size $N_T \times$

2 each. We find the capacity of such MIMO interference channels with user coordination using the tensor framework where the optimal input covariance is approximated using Algorithm 1.

For the numerical example, in (3.100), (3.101) we take $a = 1/\sqrt{3}$ and the channel entries are i.i.d. zero mean unit variance Gaussian random variables as used in [174]. The results are averaged over 100 different channel realizations. These channel realizations are known at the transmitters and the receivers. We find the sum rate achieved via the tensor framework assuming user coordination, and denote it using R_T. We compare R_T with the capacity of K parallel non-interfering channels found using standard water-filling approach and denote it as R_U. In [174], R_U has been used as an upper bound on the achievable sum rate while treating interference as noise. Figures 3.30-3.34 present a comparison between R_T and R_U. To calculate R_U, we find the transmit covariance matrix $\mathrm{Q}^{(k)}$ for each user input $\underline{\mathbf{x}}^{(k)}$ based on MIMO water-filling corresponding to the channel $\mathrm{H}^{(k,k)}$, and set $R_U = \sum_k \log\det(\mathrm{I} + \mathrm{H}^{(k,k)}\mathrm{Q}^{(k)}\mathrm{H}^{(k,k)H})$. Also, R_T is calculated using the objective function in (3.98) where the optimal covariance tensor $\mathcal{Q}$ is approximated using Algorithm 1.

Figure 3.30 compares R_T and R_U for a 2 users case with different power constraints $P_1 = P_2 = P$. The achievable sum rates with the two approaches are plotted against the number of antennas N_T. The sum rates for both the cases increase as N_T increases. The achievable rate with user coordination as ensured in the tensor framework, R_T is higher than the upper bound on the sum rate, R_U from [174]. It can also be observed in Figure 3.30 that as P increases, the gap between R_T and R_U increases as well. The tensor approach shows that the presence of interference can in fact give higher achievable sum rates if the transmit and receive operations are performed jointly by all the transmitting and receiving users respectively. Hence the sum rate achieved via the tensor approach, that allows cooperation at the transmitter and receiver sides, can be higher than the sum rate achieved in the absence of interference.

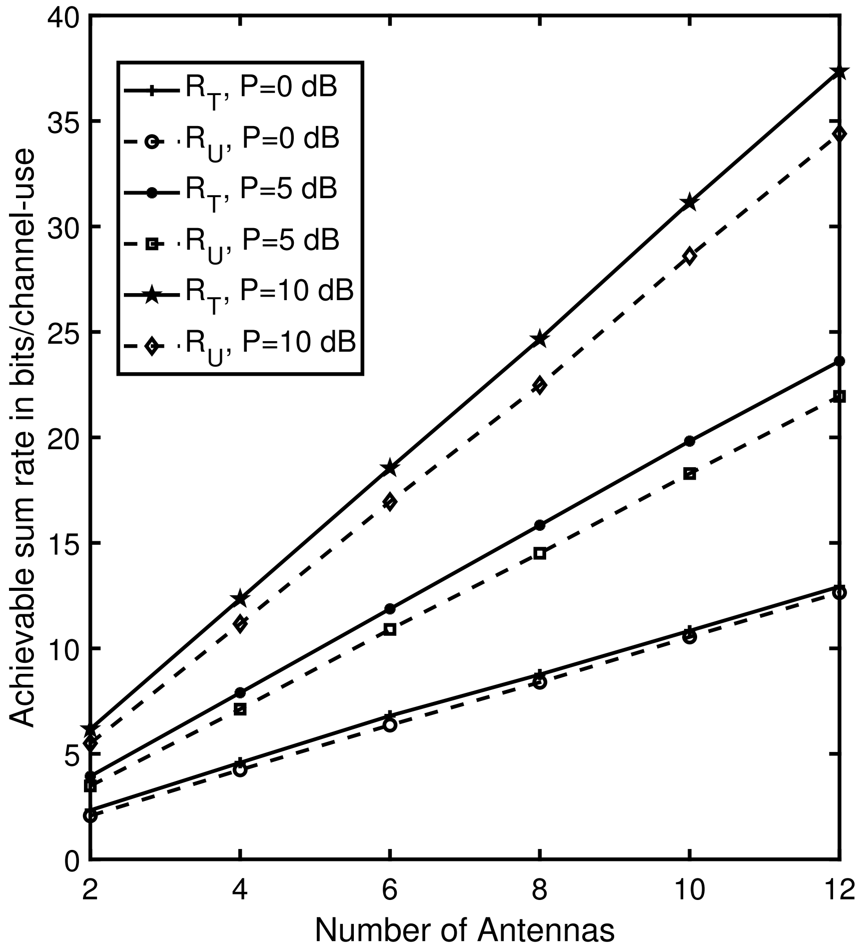


Fig. 3.30: Achievable sum rate vs number of antennas for 2 users MIMO IC.

This phenomenon is further observed in Figure 3.31 which presents the sum rate against the number of users K for a multi-user MIMO interference scenario from (3.96) with $N_T = 2$. Each users' receive signal contains a desired signal and information from $K-1$ interfering links whose power is controlled by a scalar factor a as described in the two users case. The result presented is for $a = 1/\sqrt{3}$ and for two different total power budgets $P = 5, 10$ dB with individual power constraints as $P_k = P/K$. It can be seen that R_T is always larger than R_U. Furthermore, as the number of users grows, the difference between R_T and R_U also widens, which demonstrates the advantage of considering interference as an information bearing entity rather than noise, through the tensor framework.

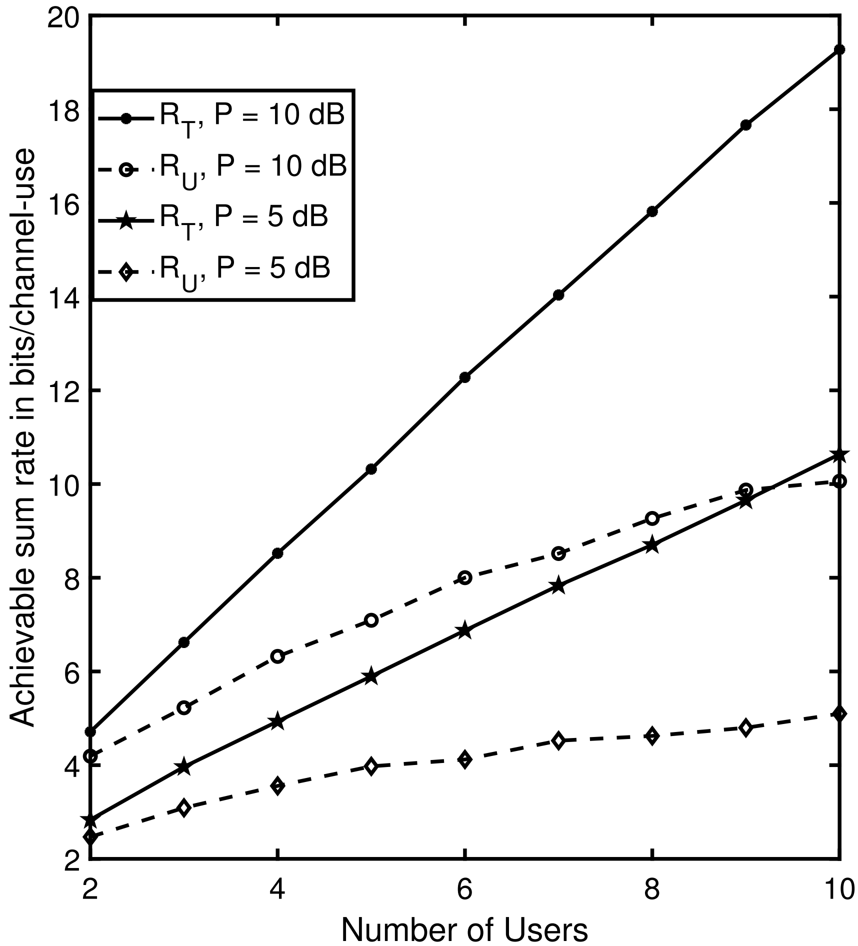


Fig. 3.31: Achievable sum rate vs number of users for K-user MIMO IC with $N_T = 2$.

To further elaborate on the role of interference, Figure 3.32 presents the sum rate R_T of the two users scenario achieved using the tensor framework for different interference power and compares it with R_U. The solid lines represent R_T and dashed lines represent R_U. The result is plotted against the interference coupling power gain defined as $G_1 = 10 \log_{10} a^2$ dB [174] for different number of antennas N_T at each device, and with power constraints $P_k = P/K$ for $k = 1, 2$ where $K = 2$ and P is set to 10 dB. It is seen clearly that higher interference leads to higher sum rate R_T using the tensor approach. Also, R_T increases with N_T and the gap between sum rate for different number of antennas widens with increasing interference. However, R_U is always lower than R_T and does not change with G_1. Since R_U

is calculated by assuming zero interference, it does not vary with changing the interference power. At very low interference power, we see that R_T and R_U are almost same. The difference between R_T and R_U starts to be significant (more than 1 bit/channel-use) at an interference power G_1 of approximately -2 dB when $N_T = 2$, and around -6 dB when $N_T = 8$.

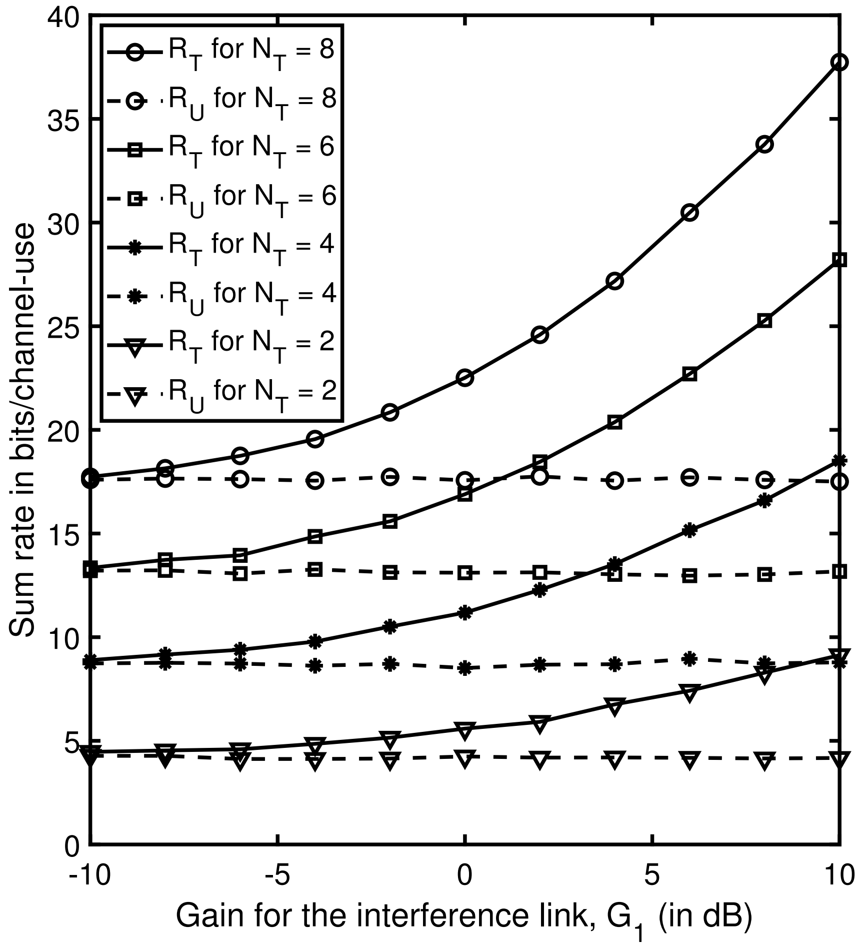


Fig. 3.32: Achievable sum rate vs interference power for 2 users MIMO IC.

Next we consider a 3 users system specified by:

$$\underline{\mathbf{y}}^{(1)} = \mathrm{H}^{(1,1)}\underline{\mathbf{x}}^{(1)} + a \cdot \mathrm{H}^{(1,2)}\underline{\mathbf{x}}^{(2)} + b \cdot \mathrm{H}^{(1,3)}\underline{\mathbf{x}}^{(3)} + \underline{\mathbf{n}}^{(1)} \tag{3.102}$$

$$\underline{\mathbf{y}}^{(2)} = a \cdot \mathrm{H}^{(2,1)}\underline{\mathbf{x}}^{(1)} + \mathrm{H}^{(2,2)}\underline{\mathbf{x}}^{(2)} + b \cdot \mathrm{H}^{(2,3)}\underline{\mathbf{x}}^{(3)} + \underline{\mathbf{n}}^{(2)} \tag{3.103}$$

$$\underline{\mathbf{y}}^{(3)} = a \cdot \mathrm{H}^{(3,1)}\underline{\mathbf{x}}^{(1)} + b \cdot \mathrm{H}^{(3,2)}\underline{\mathbf{x}}^{(2)} + \mathrm{H}^{(3,3)}\underline{\mathbf{x}}^{(3)} + \underline{\mathbf{n}}^{(3)} \tag{3.104}$$

Such a system model can be represented using (3.97) with $\mathbf{Y}, \mathbf{X}$ and $\mathbf{N}$ as $N_T \times 3$ matrices each and $\mathcal{H}$ as an $N_T \times 3 \times N_T \times 3$ tensor. The number of antennas at each device is denoted by N_T. Each destination user receives signals from a direct link and 2 interfering links whose power is controlled by scalar factors a and b respectively. In Figure 3.33 we present R_T and R_U against interference power with three users for $N_T = 2$. The power constraints for each user is same, i.e. $P_k = P/K$ where $K = 3$, and P is set to 5 dB. In Figure 3.33 we have $G_1 = 10\log_{10} a^2$ dB and $G_2 = 10\log_{10} b^2$ dB, where G_1, G_2 represent the strength of the 2 interfering links for each user. It can be seen that R_T is low when the interference power of both links is weak. With increasing strength of the interfering links, we get higher sum rate R_T. The curve for R_U does not vary with change in G_1 and G_2 and is always lower than R_T. Similar observation can be made in Figure 3.34 which represents R_T and R_U for $N_T = 4$. On comparing Figures 3.33 and 3.34, we see that R_T and R_U increase as N_T increases from 2 to 4. Notice that difference in R_T for $N_T = 2$ and 4 is wider for large values of G_1, G_2, i.e. when the interference gets stronger. The tensor framework allows to treat interfering terms as information bearing components, resulting in higher rates with increasing G_1, G_2 and N_T.

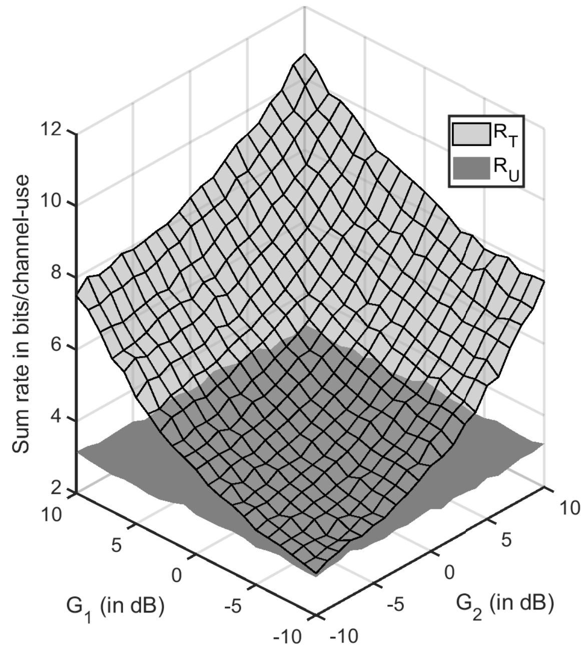


Fig. 3.33: Achievable sum rate vs interference power for 3 users MIMO IC with $N_T = 2$.

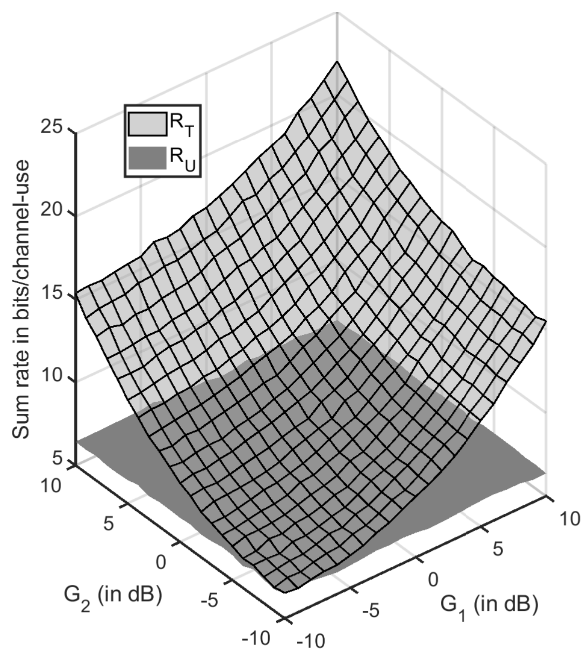


Fig. 3.34: Achievable sum rate vs interference power for 3 users MIMO IC with $N_T = 4$.

3.5 Chapter Summary

In this chapter, we formulated and solved the problem of finding the Shannon capacity of higher order tensor channels. In particular, the tensor framework's ability to mathematically represent a family of domain specific power constraints is utilized. The optimal solution is developed using the KKT conditions under such power constraints. An algorithmic approach is presented to approximate the optimum input covariance for any SNR setting. It was shown that for a fixed tensor channel, an exponential increase in channel multiplexing gain can be achieved with increase in the number of domains. Through numerical examples, the channel capacity behavior is analysed for different channel orders and dimensions. It was also shown that the capacity of tensor channels decreases as the correlation among the channel components increases. An example of MIMO GFDM system was presented to illustrate how the impact of various transmit and receive filter parameters on the equivalent channel capacity, and domain trade-offs can be analyzed in the tensor framework. The tensor formalism allows to characterize the capacity of multi-user MIMO systems having per user power constraints for any number of users. In case of multi-user MAC and IC channels, it was shown that the tensor approach leads to a cooperative users approach which provides higher sum rates as compared to the independent users approach. The tensor framework allows to capture the user cooperation in the form of a joint covariance tensor across all the domains. It is shown that such an improvement in achievable rates through user cooperation becomes even more significant as the number of users grows or the power of the interfering links increases.

Chapter 4

MMSE Estimation of Tensors

In this chapter, we develop a framework for MMSE estimation which is concerned with estimating a signal in tensor form from a tensor based noisy observation. We present the notions of best MMSE estimation, and also the tensor multi-linear and widely multi-linear estimators. The notion of the error covariance tensor associated with the MMSE estimator, as presented in this chapter, can be used to establish a relationship with the gradient of the mutual information, thereby linking the information and estimation measures, as will be shown in Chapter 5.

The commonly used methods for tensor estimation found in literature rely on the Tucker or PARAFAC (CP) decompositions. The Tucker product based technique is also known as the n-mode Wiener filtering approach [180], and aims to find N separate factor matrices along each mode of the order N tensor to be estimated. The mode-n product between the observed noisy tensor and the factor matrices is then used to find the estimate. Such an approach has found applications in various areas such as image processing [180], speech processing [181], and communication systems [80]. However, the additional assumption that the multi-linear estimator is separable across all the modes, makes the Tucker approach sub-optimal within the class of multi-linear estimators. Estimation using CP model relies on matrix unfolding of the observed tensor and uses an alternating least squares method to find the separate factor tensors [182]. Such an estimation technique assumes a finite rank decomposition of the tensor to be estimated, and is often employed for tensor completion problems [183]. Further, it has applications in communication systems for joint channel and symbol estimation if the signal can be assumed to have a low rank CP structure [82]. In this chapter, we will first establish the best MMSE estimators, and then use the Einstein product to find a multi-linear operator where no such separability or rank constraints are assumed on the tensor to be estimated.

4.1 Tensor Framework for Estimation

Consider the problem of estimating a complex tensor $\mathcal{X} \in \mathbb{C}^{I_1 \times \ldots \times I_N}$ from an observed complex tensor $\mathcal{Y} \in \mathbb{C}^{J_1 \times \ldots \times J_M}$. Throughout this chapter, we assume that the observed tensor and the tensor to be estimated have zero mean. We first establish an Orthogonality principle for tensors through the following theorem :

Theorem 3. *Let $g : \mathbb{C}^{J_1 \times \ldots \times J_M} \to \mathbb{C}^{I_1 \times \ldots \times I_N}$ be a tensor valued function of a tensor such that $\hat{\mathcal{X}} = g(\mathcal{Y}) \in \mathbb{C}^{I_1 \times \ldots \times I_N}$ is an estimator of tensor $\mathcal{X} \in \mathbb{C}^{I_1 \times \ldots \times I_N}$ based on the observation tensor $\mathcal{Y} \in \mathbb{C}^{J_1 \times \ldots \times J_M}$. We define the error tensor as $\mathcal{E} = \mathcal{X} - g(\mathcal{Y})$, then if :*

$$\mathbb{E}[\langle \mathcal{E}, h(\mathcal{Y}) \rangle] = 0 \quad \textit{for any } h : \mathbb{C}^{J_1 \times \ldots \times J_M} \to \mathbb{C}^{I_1 \times \ldots \times I_N} \tag{4.1}$$

then,

$$\mathbb{E}[||\mathcal{E}||^2] \leq \mathbb{E}[||\mathcal{X} - h(\mathcal{Y})||^2]. \tag{4.2}$$

Proof of Theorem 3 is provided in Appendix B.3. Using similar line of proof as for Theorem 3, the following corollaries can be established :

Corollary 3.1. Let $g, h : \mathbb{C}^{J_1 \times \ldots \times J_M} \to \mathbb{C}^{I_1 \times \ldots \times I_N}$ be tensor valued functions of tensors such that $g(\mathcal{Y}) = \mathcal{A}_1 *_M \mathcal{Y} + \mathcal{A}_2 *_M \mathcal{Y}^*$ and $h(\mathcal{Y}) = \mathcal{B}_1 *_M \mathcal{Y} + \mathcal{B}_2 *_M \mathcal{Y}^*$ where $\mathcal{A}_1, \mathcal{A}_2, \mathcal{B}_1, \mathcal{B}_2 \in \mathbb{C}^{I_1 \times \ldots I_N \times J_1 \times \ldots \times J_M}$. Let $g(\mathcal{Y})$ be an estimator of tensor $\mathcal{X} \in \mathbb{C}^{I_1 \times \ldots \times I_N}$ based on the observation tensor $\mathcal{Y} \in \mathbb{C}^{J_1 \times \ldots \times J_M}$. Then for the error tensor $\mathcal{E} = \mathcal{X} - g(\mathcal{Y})$, if,

$$\mathbb{E}[\langle \mathcal{E}, h(\mathcal{Y}) \rangle] = 0 \qquad \text{for any } \mathcal{B}_1, \mathcal{B}_2 \in \mathbb{C}^{I_1 \times \ldots I_N \times J_1 \times \ldots \times J_M} \tag{4.3}$$

then,

$$\mathbb{E}[||\mathcal{E}||^2] \leq \mathbb{E}[||\mathcal{X} - h(\mathcal{Y})||^2]. \tag{4.4}$$

Corollary 3.2. Let $g, h : \mathbb{C}^{J_1\times\ldots\times J_M} \to \mathbb{C}^{I_1\times\ldots\times I_N}$ be tensor valued functions of tensors such that $g(\mathcal{Y}) = \mathcal{A} *_M \mathcal{Y}$ and $h(\mathcal{Y}) = \mathcal{B} *_M \mathcal{Y}$ where $\mathcal{A}, \mathcal{B} \in \mathbb{C}^{I_1\times\ldots I_N\times J_1\times\ldots\times J_M}$. Let $g(\mathcal{Y})$ be an estimator of tensor $\mathcal{X} \in \mathbb{C}^{I_1\times\ldots\times I_N}$ based on the observation tensor $\mathcal{Y} \in \mathbb{C}^{J_1\times\ldots\times J_M}$. Then for the error tensor $\mathcal{E} = \mathcal{X} - g(\mathcal{Y})$, if

$$\mathbb{E}[\langle \mathcal{E}, h(\mathcal{Y})\rangle] = 0 \quad \text{for any } \mathcal{B} \in \mathbb{C}^{I_1\times\ldots I_N\times J_1\times\ldots\times J_M} \tag{4.5}$$

then,

$$\mathbb{E}[||\mathcal{E}||^2] \leq \mathbb{E}[||\mathcal{X} - h(\mathcal{Y})||^2]. \tag{4.6}$$

4.1.1 Best MMSE Estimation of Tensors

The objective is to find the function $g(\mathcal{Y})$ which is the best estimator of $\mathcal{X}$ in mean square error sense. Let $h(\mathcal{Y})$ be any other estimator. Then we have :

$$\mathbb{E}[\langle \mathcal{X} - g(\mathcal{Y}), h(\mathcal{Y})\rangle] = \mathbb{E}_{\mathcal{Y}}\Big[\mathbb{E}[\langle \mathcal{X} - g(\mathcal{Y}), h(\mathcal{Y})\rangle \mid \mathcal{Y}]\Big] \tag{4.7}$$

$$= \mathbb{E}_{\mathcal{Y}}\Big[\mathbb{E}[(\mathcal{X} - g(\mathcal{Y})) *_N h(\mathcal{Y})^* \mid \mathcal{Y}]\Big]. \tag{4.8}$$

Conditioned on $\mathcal{Y}$, we can take $h(\mathcal{Y})^*$ and $g(\mathcal{Y})$ outside of the inner expectation.

$$\mathbb{E}[\langle \mathcal{X} - g(\mathcal{Y}), h(\mathcal{Y})\rangle] = \mathbb{E}_{\mathcal{Y}}\Big[\mathbb{E}[(\mathcal{X} - g(\mathcal{Y})) \mid \mathcal{Y}] *_N h(\mathcal{Y})^*\Big] \tag{4.9}$$

$$= \mathbb{E}_{\mathcal{Y}}\Big[(\mathbb{E}[\mathcal{X} \mid \mathcal{Y}] - g(\mathcal{Y})) *_N h(\mathcal{Y})^*\Big]. \tag{4.10}$$

From orthogonality principle, we know that (4.10) has to be 0 for any $h(\mathcal{Y})$ to minimize the mean square error. Thus $g(\mathcal{Y}) = \mathbb{E}[\mathcal{X} \mid \mathcal{Y}]$ is the best MMSE estimator of tensor $\mathcal{X}$ from $\mathcal{Y}$:

$$\hat{\mathcal{X}} = \mathbb{E}[\mathcal{X} \mid \mathcal{Y}]. \tag{4.11}$$

Since $\mathbb{E}[\hat{\mathcal{X}}] = \mathbb{E}[\mathbb{E}[\mathcal{X} \mid \mathcal{Y}]] = \mathbb{E}[\mathcal{X}]$, the error $\mathcal{E} = \mathcal{X} - \hat{\mathcal{X}}$, is a zero mean tensor with the associated covariance as :

$$\mathcal{Q}_{\mathcal{E}} = \mathbb{E}[(\mathcal{X} - \hat{\mathcal{X}}) \circ (\mathcal{X} - \hat{\mathcal{X}})^*] \tag{4.12}$$

$$= \mathbb{E}_{\mathcal{X},\mathcal{Y}}[(\mathcal{X} - \mathbb{E}[\mathcal{X} \mid \mathcal{Y}]) \circ (\mathcal{X} - \mathbb{E}[\mathcal{X} \mid \mathcal{Y}])^*]. \tag{4.13}$$

Note that (4.13) represents the order $2N$ covariance tensor of the error when conditional mean estimator is used to estimate an order N tensor. The trace of such a covariance tensor gives us the Mean Squared Error (MSE) in estimation as :

$$\mathbb{E}[||\boldsymbol{\mathcal{E}}||^2] = \mathbb{E}\Big[\sum_{i_1,\dots,i_N} |\boldsymbol{\mathcal{E}}_{i_1,\dots,i_N}|^2\Big] = \sum_{i_1,\dots,i_N} \mathbb{E}[|\boldsymbol{\mathcal{E}}_{i_1,\dots,i_N}|^2] = \mathrm{tr}(\mathcal{Q}_{\boldsymbol{\mathcal{E}}}) \tag{4.14}$$

Notice that similar to the vector case [121], conditioning on $\boldsymbol{\mathcal{Y}}$ is same as conditioning on $\boldsymbol{\mathcal{Y}}^*$, i.e. $\mathbb{E}[\boldsymbol{\mathcal{X}} \mid \boldsymbol{\mathcal{Y}}] = \mathbb{E}[\boldsymbol{\mathcal{X}} \mid \boldsymbol{\mathcal{Y}}^*]$. Hence the estimator conditioned on both $\boldsymbol{\mathcal{Y}}$ and $\boldsymbol{\mathcal{Y}}^*$ gives no extra information as compared to the one conditioned only on $\boldsymbol{\mathcal{Y}}$. Also the conditional mean estimator of $\boldsymbol{\mathcal{X}}^*$ would just be the complex conjugate of the conditional mean estimator of $\boldsymbol{\mathcal{X}}$.

4.1.2 Widely Multi-linear and Multi-linear MMSE Estimation of Tensors

A function f between linear spaces V and W, denoted as $f : V \to W$, over the same field $\mathbb{K}$ is said to be a linear map if $f(u+v) = f(u) + f(v)$ and $f(cu) = cf(u)$ for any $u, v \in V$ and scalar $c \in \mathbb{K}$ [184]. Consider a simple linear relation over complex field as $y = hx$ where $x, y, h \in \mathbb{C}$. Such a relation can be written as a transformation in terms of the real and imaginary components of y, x in $\mathbb{R}^2$ as :

$$\begin{bmatrix} y_r \\ y_i \end{bmatrix} = \begin{bmatrix} h_r & -h_i \\ h_i & h_r \end{bmatrix} \begin{bmatrix} x_r \\ x_i \end{bmatrix} \tag{4.15}$$

where x_r, y_r, h_r denote the real parts and x_i, y_i, h_i denote the imaginary parts of x, y, h respectively. Thus, linear transformations on $\mathbb{R}^2$ are linear on complex field $\mathbb{C}$ only if the transforming matrix has the specific structure as shown in (4.15). But not all linear relations on $\mathbb{R}^2$ need to have the same structural constraints on the transforming matrix as in (4.15). Thus, in order to establish a transformation on $\mathbb{C}$ which has a more general linear form in $\mathbb{R}^2$, a widely linear relation can be used where the function depends not only on x, but also its conjugate x^* as $y = g(x) = hx + kx^*$ for $h, k \in \mathbb{C}$. Clearly this relation is not linear in x as it does not satisfy the scalar multiplication property, i.e. $g(cx) = chx + c^*kx^* \neq cg(x)$ for $c \in \mathbb{C}$. However, when expressed in $\mathbb{R}^2$, we can write $g(x)$ as :

$$\begin{bmatrix} y_r \\ y_i \end{bmatrix} = \begin{bmatrix} h_r & -h_i \\ h_i & h_r \end{bmatrix} \begin{bmatrix} x_r \\ x_i \end{bmatrix} + \begin{bmatrix} k_r & -k_i \\ k_i & k_r \end{bmatrix} \begin{bmatrix} x_r \\ -x_i \end{bmatrix} = \begin{bmatrix} h_r + k_r & -h_i + k_i \\ h_i + k_i & h_r - k_r \end{bmatrix} \begin{bmatrix} x_r \\ x_i \end{bmatrix} \tag{4.16}$$

which is a more general form of linear transformation on $\mathbb{R}^2$. Hence the equivalent of a general linear relation in $\mathbb{R}^2$ on $\mathbb{C}$ is the widely linear transformation, also known as linear-conjugate-linear relationship [121]. Essentially, the set of complex linear transformations is a subset of widely linear transformations where $k = 0$. The notion of widely linear transformation was used to define the Widely Linear Minimum Mean Squared Error (WLMMSE) estimate of a complex vector in [121, 185] where the estimate of a vector depends linearly on both the observed vector and its conjugate or in other words depends widely linearly on the observed vector. For complex data estimation, assuming a widely linear dependence instead of linear can significantly improve the performance since the former uses both covariance and pseudo-covariance for estimation, while the latter employs only the covariance [121].

In this section, we restrict ourselves to the class of estimators which depend linearly on the received tensor $\mathcal{Y}$ and its conjugate $\mathcal{Y}^*$. If the estimator depends linearly only on tensor $\mathcal{Y}$, it is called a multi-linear estimate, whereas if the estimator depends linearly on both $\mathcal{Y}$ and $\mathcal{Y}^*$ through different set of coefficients, it is called a widely multi-linear estimate. In order to estimate the tensor $\mathcal{X} \in \mathbb{C}^{I_1 \times \ldots \times I_N}$ from an observed complex tensor $\mathcal{Y} \in \mathbb{C}^{J_1 \times \ldots \times J_M}$ and its conjugate by a multi-linear structure, we are looking for the tensors $\mathcal{A}_1, \mathcal{A}_2 \in \mathbb{C}^{I_1 \times \ldots I_N \times J_1 \times \ldots \times J_M}$ such that the estimator:

$$\hat{\mathcal{X}}_{WL} = \mathcal{A}_1 *_M \mathcal{Y} + \mathcal{A}_2 *_M \mathcal{Y}^* \tag{4.17}$$

satisfies

$$\mathbb{E}[||\mathcal{X} - (\mathcal{A}_1 *_M \mathcal{Y} + \mathcal{A}_2 *_M \mathcal{Y}^*)||^2] \leq \mathbb{E}[||\mathcal{X} - (\mathcal{B}_1 *_M \mathcal{Y} + \mathcal{B}_2 *_M \mathcal{Y}^*)||^2] \tag{4.18}$$

for any other tensors $\mathcal{B}_1, \mathcal{B}_2 \in \mathbb{C}^{I_1 \times \ldots \times I_N \times J_1 \times \ldots \times J_M}$. From Corollary 3.1, we know that the optimal $\mathcal{A}_1$ and $\mathcal{A}_2$ will be such that :

$$\mathbb{E}[\langle (\mathcal{X} - \hat{\mathcal{X}}_{WL}), (\mathcal{B}_1 *_M \mathcal{Y} + \mathcal{B}_2 *_M \mathcal{Y}^*) \rangle] = 0 \tag{4.19}$$

for any choice of $\mathcal{B}_1, \mathcal{B}_2 \in \mathbb{C}^{I_1 \times \ldots \times I_N \times J_1 \times \ldots \times J_M}$. On substituting (4.17) into (4.19), we get :

$$\mathbb{E}[(\mathcal{X} - \mathcal{A}_1 *_M \mathcal{Y} - \mathcal{A}_2 *_M \mathcal{Y}^*) *_N (\mathcal{B}_1 *_M \mathcal{Y} + \mathcal{B}_2 *_M \mathcal{Y}^*)^*] = 0. \tag{4.20}$$

From (2.10), we can write $(\mathcal{B}_1 *_M \mathcal{Y})^* = (\mathcal{Y}^* *_M \mathcal{B}_1^H)$ and $(\mathcal{B}_2 *_M \mathcal{Y}^*)^* = (\mathcal{Y} *_M \mathcal{B}_2^H)$, and hence the left hand side of (4.20) can be written as :

$$
\begin{aligned}
&\mathbb{E}[(\boldsymbol{\mathcal{X}} - \mathcal{A}_1 *_M \boldsymbol{\mathcal{Y}} - \mathcal{A}_2 *_M \boldsymbol{\mathcal{Y}}^*) *_N ((\boldsymbol{\mathcal{Y}}^* *_M \mathcal{B}_1^H) + (\boldsymbol{\mathcal{Y}} *_M \mathcal{B}_2^H))] \\
&= \mathbb{E}[\operatorname{tr}\{(\boldsymbol{\mathcal{X}} - \mathcal{A}_1 *_M \boldsymbol{\mathcal{Y}} - \mathcal{A}_2 *_M \boldsymbol{\mathcal{Y}}^*) \circ ((\boldsymbol{\mathcal{Y}}^* *_M \mathcal{B}_1^H) + (\boldsymbol{\mathcal{Y}} *_M \mathcal{B}_2^H))\}] \quad \text{(from (2.18))} \quad (4.21) \\
&= \mathbb{E}[\operatorname{tr}\{\boldsymbol{\mathcal{X}} \circ \boldsymbol{\mathcal{Y}}^* *_M \mathcal{B}_1^H - \mathcal{A}_1 *_M \boldsymbol{\mathcal{Y}} \circ \boldsymbol{\mathcal{Y}}^* *_M \mathcal{B}_1^H - \mathcal{A}_2 *_M \boldsymbol{\mathcal{Y}}^* \circ \boldsymbol{\mathcal{Y}}^* *_M \mathcal{B}_1^H + \\
&\qquad \boldsymbol{\mathcal{X}} \circ \boldsymbol{\mathcal{Y}} *_M \mathcal{B}_2^H - \mathcal{A}_1 *_M \boldsymbol{\mathcal{Y}} \circ \boldsymbol{\mathcal{Y}} *_M \mathcal{B}_2^H - \mathcal{A}_2 *_M \boldsymbol{\mathcal{Y}}^* \circ \boldsymbol{\mathcal{Y}} *_M \mathcal{B}_2^H\}] \quad (4.22) \\
&= \operatorname{tr}\{\underbrace{\mathbb{E}[\boldsymbol{\mathcal{X}} \circ \boldsymbol{\mathcal{Y}}^*]}_{\mathcal{C}_{\boldsymbol{\mathcal{X}}\boldsymbol{\mathcal{Y}}}} *_M \mathcal{B}_1^H - \mathcal{A}_1 *_M \underbrace{\mathbb{E}[\boldsymbol{\mathcal{Y}} \circ \boldsymbol{\mathcal{Y}}^*]}_{\mathcal{C}_{\boldsymbol{\mathcal{Y}}}} *_M \mathcal{B}_1^H - \mathcal{A}_2 *_M \underbrace{\mathbb{E}[\boldsymbol{\mathcal{Y}}^* \circ \boldsymbol{\mathcal{Y}}^*]}_{\tilde{\mathcal{C}}_{\boldsymbol{\mathcal{Y}}}^*} *_M \mathcal{B}_1^H + \\
&\qquad \underbrace{\mathbb{E}[\boldsymbol{\mathcal{X}} \circ \boldsymbol{\mathcal{Y}}]}_{\tilde{\mathcal{C}}_{\boldsymbol{\mathcal{X}}\boldsymbol{\mathcal{Y}}}} *_M \mathcal{B}_2^H - \mathcal{A}_1 *_M \underbrace{\mathbb{E}[\boldsymbol{\mathcal{Y}} \circ \boldsymbol{\mathcal{Y}}]}_{\tilde{\mathcal{C}}_{\boldsymbol{\mathcal{Y}}}} *_M \mathcal{B}_2^H - \mathcal{A}_2 *_M \underbrace{\mathbb{E}[\boldsymbol{\mathcal{Y}}^* \circ \boldsymbol{\mathcal{Y}}]}_{\mathcal{C}_{\boldsymbol{\mathcal{Y}}}^*} *_M \mathcal{B}_2^H\} \quad (4.23) \\
&= \operatorname{tr}\{\underbrace{(\mathcal{C}_{\boldsymbol{\mathcal{X}}\boldsymbol{\mathcal{Y}}} - \mathcal{A}_1 *_M \mathcal{C}_{\boldsymbol{\mathcal{Y}}} - \mathcal{A}_2 *_M \tilde{\mathcal{C}}_{\boldsymbol{\mathcal{Y}}}^*)}_{\bar{\mathcal{B}}_1} *_M \mathcal{B}_1^H + \underbrace{(\tilde{\mathcal{C}}_{\boldsymbol{\mathcal{X}}\boldsymbol{\mathcal{Y}}} - \mathcal{A}_1 *_M \tilde{\mathcal{C}}_{\boldsymbol{\mathcal{Y}}} - \mathcal{A}_2 *_M \mathcal{C}_{\boldsymbol{\mathcal{Y}}}^*)}_{\bar{\mathcal{B}}_2} *_M \mathcal{B}_2^H\}. \\
&\hfill (4.24)
\end{aligned}
$$

For (4.24) to be 0, for any $\mathcal{B}_1$ and $\mathcal{B}_2$, we need $\bar{\mathcal{B}}_1$ and $\bar{\mathcal{B}}_2$ to be all zero tensors, which gives us the conditions for optimal $\mathcal{A}_1$ and $\mathcal{A}_2$:

$$
\mathcal{C}_{\boldsymbol{\mathcal{X}}\boldsymbol{\mathcal{Y}}} = \mathcal{A}_1 *_M \mathcal{C}_{\boldsymbol{\mathcal{Y}}} + \mathcal{A}_2 *_M \tilde{\mathcal{C}}_{\boldsymbol{\mathcal{Y}}}^* \tag{4.25}
$$

$$
\tilde{\mathcal{C}}_{\boldsymbol{\mathcal{X}}\boldsymbol{\mathcal{Y}}} = \mathcal{A}_1 *_M \tilde{\mathcal{C}}_{\boldsymbol{\mathcal{Y}}} + \mathcal{A}_2 *_M \mathcal{C}_{\boldsymbol{\mathcal{Y}}}^*. \tag{4.26}
$$

Equations (4.25) and (4.26) are systems of multi-linear equations which can be solved for $\mathcal{A}_1$ and $\mathcal{A}_1$ using methods described in [48]. If the inverse of the covariance of $\boldsymbol{\mathcal{Y}}$ exists, then from (4.26), we get $\mathcal{A}_2 = (\tilde{\mathcal{C}}_{\boldsymbol{\mathcal{X}}\boldsymbol{\mathcal{Y}}} - \mathcal{A}_1 *_M \tilde{\mathcal{C}}_{\boldsymbol{\mathcal{Y}}}) *_M \mathcal{C}_{\boldsymbol{\mathcal{Y}}}^{*-1}$ which we can substitute in (4.25) to get :

$$
\begin{aligned}
\mathcal{C}_{\boldsymbol{\mathcal{X}}\boldsymbol{\mathcal{Y}}} &= \mathcal{A}_1 *_M \mathcal{C}_{\boldsymbol{\mathcal{Y}}} + (\tilde{\mathcal{C}}_{\boldsymbol{\mathcal{X}}\boldsymbol{\mathcal{Y}}} - \mathcal{A}_1 *_M \tilde{\mathcal{C}}_{\boldsymbol{\mathcal{Y}}}) *_M \mathcal{C}_{\boldsymbol{\mathcal{Y}}}^{*-1} *_M \tilde{\mathcal{C}}_{\boldsymbol{\mathcal{Y}}}^* && (4.27) \\
&= \mathcal{A}_1 *_M (\mathcal{C}_{\boldsymbol{\mathcal{Y}}} - \tilde{\mathcal{C}}_{\boldsymbol{\mathcal{Y}}} *_M \mathcal{C}_{\boldsymbol{\mathcal{Y}}}^{*-1} *_M \tilde{\mathcal{C}}_{\boldsymbol{\mathcal{Y}}}^*) + (\tilde{\mathcal{C}}_{\boldsymbol{\mathcal{X}}\boldsymbol{\mathcal{Y}}} *_M \mathcal{C}_{\boldsymbol{\mathcal{Y}}}^{*-1} *_M \tilde{\mathcal{C}}_{\boldsymbol{\mathcal{Y}}}^*) && (4.28) \\
&\Rightarrow \mathcal{A}_1 = \left(\mathcal{C}_{\boldsymbol{\mathcal{X}}\boldsymbol{\mathcal{Y}}} - \tilde{\mathcal{C}}_{\boldsymbol{\mathcal{X}}\boldsymbol{\mathcal{Y}}} *_M \mathcal{C}_{\boldsymbol{\mathcal{Y}}}^{*-1} *_M \tilde{\mathcal{C}}_{\boldsymbol{\mathcal{Y}}}^*\right) *_M \mathcal{P}_{\boldsymbol{\mathcal{Y}}}^{-1} && (4.29) \\
&\text{where,} \quad \mathcal{P}_{\boldsymbol{\mathcal{Y}}} = (\mathcal{C}_{\boldsymbol{\mathcal{Y}}} - \tilde{\mathcal{C}}_{\boldsymbol{\mathcal{Y}}} *_M \mathcal{C}_{\boldsymbol{\mathcal{Y}}}^{*-1} *_M \tilde{\mathcal{C}}_{\boldsymbol{\mathcal{Y}}}^*). && (4.30)
\end{aligned}
$$

Also from (4.25) we get $\mathcal{A}_1 = (\mathcal{C}_{\boldsymbol{\mathcal{X}}\boldsymbol{\mathcal{Y}}} - \mathcal{A}_2 *_M \tilde{\mathcal{C}}_{\boldsymbol{\mathcal{Y}}}^*) *_M \mathcal{C}_{\boldsymbol{\mathcal{Y}}}^{-1}$ which we can substitute in (4.26) to get :

$$\mathcal{A}_2 = \left(\tilde{\mathcal{C}}_{\mathcal{XY}} - \mathcal{C}_{\mathcal{XY}} *_M \mathcal{C}_{\mathcal{Y}}^{-1} *_M \tilde{\mathcal{C}}_{\mathcal{Y}}\right) *_M \mathcal{P}_{\mathcal{Y}}^{*-1}. \tag{4.31}$$

Substituting (4.29) and (4.31) into (4.17) gives us the widely multi-linear estimate $\hat{\mathcal{X}}_{WL}$ of the tensor $\mathcal{X}$. The covariance of the corresponding error tensor is :

$$\begin{aligned}\mathcal{Q}_{WL} &= \mathbb{E}[(\mathcal{X} - \hat{\mathcal{X}}_{WL}) \circ (\mathcal{X} - \hat{\mathcal{X}}_{WL})^*] \\ &= \mathcal{C}_{\mathcal{X}} - \mathcal{A}_1 *_M \mathcal{C}_{\mathcal{XY}}^H - \mathcal{A}_2 *_M \tilde{\mathcal{C}}_{\mathcal{XY}}^H.\end{aligned} \tag{4.32}$$

A detailed derivation of (4.32) has been included in Appendix B.4. On substituting $\mathcal{A}_1$ and $\mathcal{A}_2$ from (4.29) and (4.31) respectively, we get :

$$\begin{aligned}\mathcal{Q}_{WL} = \mathcal{C}_{\mathcal{X}} - \left(\mathcal{C}_{\mathcal{XY}} - \tilde{\mathcal{C}}_{\mathcal{XY}} *_M \mathcal{C}_{\mathcal{Y}}^{*-1} *_M \tilde{\mathcal{C}}_{\mathcal{Y}}^*\right) *_M \mathcal{P}_{\mathcal{Y}}^{-1} *_M \mathcal{C}_{\mathcal{XY}}^H - \\ \left(\tilde{\mathcal{C}}_{\mathcal{XY}} - \mathcal{C}_{\mathcal{XY}} *_M \mathcal{C}_{\mathcal{Y}}^{-1} *_M \tilde{\mathcal{C}}_{\mathcal{Y}}\right) *_M \mathcal{P}_{\mathcal{Y}}^{*-1} *_M \tilde{\mathcal{C}}_{\mathcal{XY}}^H.\end{aligned} \tag{4.33}$$

The quantity $\mathcal{A}_1 *_M \mathcal{C}_{\mathcal{XY}}^H + \mathcal{A}_2 *_M \tilde{\mathcal{C}}_{\mathcal{XY}}^H$ can be seen as the cross covariance between the widely multi-linear estimator $\hat{\mathcal{X}}_{WL}$ and the tensor $\mathcal{X}$ (based on (B.11)). Hence, intuitively the error covariance tensor is the difference between the covariance tensor of $\mathcal{X}$ and the cross covariance between $\hat{\mathcal{X}}_{WL}$ and $\mathcal{X}$, i.e. $\mathcal{Q}_{WL} = \mathcal{C}_{\mathcal{X}} - \mathcal{C}_{\hat{\mathcal{X}}_{WL}\mathcal{X}}$.

The corresponding MSE is given by

$$\begin{aligned}\mathrm{MSE}_{WL} &= \mathrm{tr}(\mathcal{Q}_{WL}) = \mathrm{tr}(\mathcal{C}_{\mathcal{X}} - \mathcal{A}_1 *_M \mathcal{C}_{\mathcal{XY}}^H - \mathcal{A}_2 *_M \tilde{\mathcal{C}}_{\mathcal{XY}}^H) \\ &= \mathrm{tr}(\mathcal{C}_{\mathcal{X}} - (\mathcal{C}_{\mathcal{XY}} - \tilde{\mathcal{C}}_{\mathcal{XY}} *_M \mathcal{C}_{\mathcal{Y}}^{*-1} *_M \tilde{\mathcal{C}}_{\mathcal{Y}}^*) *_M \mathcal{P}_{\mathcal{Y}}^{-1} *_M \mathcal{C}_{\mathcal{XY}}^H - (\tilde{\mathcal{C}}_{\mathcal{XY}} - \mathcal{C}_{\mathcal{XY}} *_M \mathcal{C}_{\mathcal{Y}}^{-1} *_M \tilde{\mathcal{C}}_{\mathcal{Y}}) *_M \mathcal{P}_{\mathcal{Y}}^{*-1} *_M \tilde{\mathcal{C}}_{\mathcal{XY}}^H).\end{aligned} \tag{4.34}$$

Next we consider the problem of MMSE estimation where we assume that the estimate of $\mathcal{X}$ depends linearly only on the received tensor $\mathcal{Y}$ and not its conjugate, i.e.

$$\hat{\mathcal{X}}_L = \mathcal{A} *_M \mathcal{Y} \tag{4.35}$$

and we want to find the tensor $\mathcal{A} \in \mathbb{C}^{I_1 \times \ldots \times I_N \times J_1 \times \ldots \times J_M}$ such that $\mathbb{E}[||\mathcal{X} - \mathcal{A} *_M \mathcal{Y}||^2] \leq \mathbb{E}[||\mathcal{X} - \mathcal{B} *_M \mathcal{Y}||^2]$ for any other tensor $\mathcal{B} \in \mathbb{C}^{I_1 \times \ldots \times I_N \times J_1 \times \ldots \times J_M}$. Using Corollary 3.2, we know that the optimal $\mathcal{A}$ will satisfy $\mathbb{E}[(\mathcal{X} - \mathcal{A} *_M \mathcal{Y}) *_N (\mathcal{B} *_M \mathcal{Y})^*] = 0$. Now using the same line of proof as for (4.24) by substituting $\mathcal{A}_2 = 0_{\mathcal{T}}$ and $\mathcal{A}_1 = \mathcal{A}$ we can get the condition for optimal $\mathcal{A}$ as :

$$\mathcal{C}_{\mathfrak{X}\mathfrak{Y}} = \mathcal{A} *_M \mathcal{C}_{\mathfrak{Y}}. \tag{4.36}$$

Equation (4.36) can be solved using methods described in [48] for $\mathcal{A}$. If the inverse of $\mathcal{C}_{\mathfrak{Y}}$ exists, then we have

$$\mathcal{A} = \mathcal{C}_{\mathfrak{X}\mathfrak{Y}} *_M \mathcal{C}_{\mathfrak{Y}}^{-1} \tag{4.37}$$

and the multi-linear MMSE estimate of $\mathfrak{X}$ is given by :

$$\hat{\mathfrak{X}}_L = (\mathcal{C}_{\mathfrak{X}\mathfrak{Y}} *_M \mathcal{C}_{\mathfrak{Y}}^{-1}) *_M \mathfrak{Y}. \tag{4.38}$$

The covariance of the corresponding error tensor is :

$$\mathcal{Q}_L = \mathbb{E}[(\mathfrak{X} - \hat{\mathfrak{X}}_L) \circ (\mathfrak{X} - \hat{\mathfrak{X}}_L)^*] \tag{4.39}$$

which we can solve by substituting $\mathcal{A}_2 = 0_{\mathcal{T}}$ and $\mathcal{A}_1 = (\mathcal{C}_{\mathfrak{X}\mathfrak{Y}} *_M \mathcal{C}_{\mathfrak{Y}}^{-1})$ in (4.34), to get:

$$\mathcal{Q}_L = \mathcal{C}_{\mathfrak{X}} - \mathcal{A} *_M \mathcal{C}_{\mathfrak{X}\mathfrak{Y}}^H = \mathcal{C}_{\mathfrak{X}} - \mathcal{C}_{\mathfrak{X}\mathfrak{Y}} *_M \mathcal{C}_{\mathfrak{Y}}^{-1} *_M \mathcal{C}_{\mathfrak{X}\mathfrak{Y}}^H. \tag{4.40}$$

The quantity $\mathcal{A} *_M \mathcal{C}_{\mathfrak{X}\mathfrak{Y}}^H$ can be seen as the cross covariance between $\hat{\mathfrak{X}}_L$ and $\mathfrak{X}$. Hence, intuitively the error covariance tensor is the difference between the covariance tensor of $\mathfrak{X}$ and the cross covariance between $\hat{\mathfrak{X}}_L$ and $\mathfrak{X}$, i.e. $\mathcal{Q}_L = \mathcal{C}_{\mathfrak{X}} - \mathcal{C}_{\hat{\mathfrak{X}}_L\mathfrak{X}}$. The MSE is given as:

$$\text{MSE}_L = \text{tr}(\mathcal{Q}_L) = \text{tr}(\mathcal{C}_{\mathfrak{X}} - \mathcal{C}_{\mathfrak{X}\mathfrak{Y}} *_M \mathcal{C}_{\mathfrak{Y}}^{-1} *_M \mathcal{C}_{\mathfrak{X}\mathfrak{Y}}^H). \tag{4.41}$$

4.1.3 Comparing Multi-linear and Widely Multi-linear MMSE Estimation

Let $g(\mathfrak{Y}) = \mathcal{A}_1 *_M \mathfrak{Y} + \mathcal{A}_2 *_M \mathfrak{Y}^*$ be a widely multi-linear estimator of $\mathfrak{X}$ based on tensor $\mathfrak{Y}$ and its conjugate $\mathfrak{Y}^*$. Based on Corollary 3.1, we know that the mean square error achieved by the widely multi-linear estimate $g(\mathfrak{Y})$ with $\mathcal{A}_1$ and $\mathcal{A}_2$ given in (4.29) and (4.31) respectively will be less than or equal to the mean square error achieved by any other choice of $\mathcal{A}_1$ and $\mathcal{A}_2$. An alternate choice of $\mathcal{A}_1$ and $\mathcal{A}_2$ can be $\mathcal{A}_1 = \mathcal{A}$ (obtained from (4.37)) and $\mathcal{A}_2 = 0_{\mathcal{T}}$, in which case $g(\mathfrak{Y})$ will represent the multi-linear estimate. Hence the mean square error achieved by a widely multi-linear estimate given by (4.34) is always less than or equal to the mean square error achieved by the multi-linear estimate given by (4.41). The performance difference between the two estimators can be analyzed by comparing their respective error covariance tensors. On substituting $\mathcal{C}_{\mathcal{X}\mathcal{Y}}$ from (4.25) into (4.40), we get :

$$\begin{aligned}
\mathcal{Q}_L &= \mathcal{C}_{\mathcal{X}} - (\mathcal{A}_1 *_M \mathcal{C}_{\mathcal{Y}} + \mathcal{A}_2 *_M \tilde{\mathcal{C}}^*_{\mathcal{Y}}) *_M \mathcal{C}^{-1}_{\mathcal{Y}} *_M (\mathcal{A}_1 *_M \mathcal{C}_{\mathcal{Y}} + \mathcal{A}_2 *_M \tilde{\mathcal{C}}^*_{\mathcal{Y}})^H && (4.42)\\
&= \mathcal{C}_{\mathcal{X}} - (\mathcal{A}_1 + \mathcal{A}_2 *_M \tilde{\mathcal{C}}^*_{\mathcal{Y}} *_M \mathcal{C}^{-1}_{\mathcal{Y}}) *_M (\mathcal{C}_{\mathcal{Y}} *_M \mathcal{A}_1^H + \tilde{\mathcal{C}}_{\mathcal{Y}} *_M \mathcal{A}_2^H) && (4.43)\\
&= \mathcal{C}_{\mathcal{X}} - \mathcal{A}_1 *_M \mathcal{C}_{\mathcal{Y}} *_M \mathcal{A}_1^H - \mathcal{A}_1 *_M \tilde{\mathcal{C}}_{\mathcal{Y}} *_M \mathcal{A}_2^H - \mathcal{A}_2 *_M \tilde{\mathcal{C}}^*_{\mathcal{Y}} *_M \mathcal{A}_1^H -\\
&\qquad \mathcal{A}_2 *_M \tilde{\mathcal{C}}^*_{\mathcal{Y}} *_M \mathcal{C}^{-1}_{\mathcal{Y}} *_M \tilde{\mathcal{C}}_{\mathcal{Y}} *_M \mathcal{A}_2^H. && (4.44)
\end{aligned}$$

Note that we have used the properties $\mathcal{C}_{\mathcal{Y}} = \mathcal{C}^H_{\mathcal{Y}}$ and $\tilde{\mathcal{C}}_{\mathcal{Y}} = \tilde{\mathcal{C}}^T_{\mathcal{Y}}$ in deriving (4.44). Similarly on substituting $\mathcal{C}_{\mathcal{X}\mathcal{Y}}$ from (4.25) and $\tilde{\mathcal{C}}_{\mathcal{X}\mathcal{Y}}$ from (4.26) into (4.32), we get

$$\begin{aligned}
\mathcal{Q}_{WL} &= \mathcal{C}_{\mathcal{X}} - \mathcal{A}_1 *_M (\mathcal{A}_1 *_M \mathcal{C}_{\mathcal{Y}} + \mathcal{A}_2 *_M \tilde{\mathcal{C}}^*_{\mathcal{Y}})^H - \mathcal{A}_2 *_M (\mathcal{A}_1 *_M \tilde{\mathcal{C}}_{\mathcal{Y}} + \mathcal{A}_2 *_M \mathcal{C}^*_{\mathcal{Y}})^H\\
&= \mathcal{C}_{\mathcal{X}} - \mathcal{A}_1 *_M \mathcal{C}_{\mathcal{Y}} *_M \mathcal{A}_1^H - \mathcal{A}_1 *_M \tilde{\mathcal{C}}_{\mathcal{Y}} *_M \mathcal{A}_2^H - \mathcal{A}_2 *_M \tilde{\mathcal{C}}^*_{\mathcal{Y}} *_M \mathcal{A}_1^H - \mathcal{A}_2 *_M \mathcal{C}^*_{\mathcal{Y}} *_M \mathcal{A}_2^H.
\end{aligned} \tag{4.45}$$

Subtracting (4.45) from (4.44), we get

$$\begin{aligned}
\Delta_{\mathcal{Q}} = \mathcal{Q}_L - \mathcal{Q}_{WL} &= \mathcal{A}_2 *_M (\mathcal{C}^*_{\mathcal{Y}} - \tilde{\mathcal{C}}^*_{\mathcal{Y}} *_M \mathcal{C}^{-1}_{\mathcal{Y}} *_M \tilde{\mathcal{C}}_{\mathcal{Y}}) *_M \mathcal{A}_2^H && (4.46)\\
&= \mathcal{A}_2 * \mathcal{P}^*_{\mathcal{Y}} *_M \mathcal{A}_2^H \qquad \text{(from (4.30))} && (4.47)\\
&= (\tilde{\mathcal{C}}_{\mathcal{X}\mathcal{Y}} - \mathcal{C}_{\mathcal{X}\mathcal{Y}} *_M \mathcal{C}^{-1}_{\mathcal{Y}} *_M \tilde{\mathcal{C}}_{\mathcal{Y}}) *_M \mathcal{P}^{*-1}_{\mathcal{Y}} *_M (\tilde{\mathcal{C}}_{\mathcal{X}\mathcal{Y}} - \mathcal{C}_{\mathcal{X}\mathcal{Y}} *_M \mathcal{C}^{-1}_{\mathcal{Y}} *_M \tilde{\mathcal{C}}_{\mathcal{Y}})^H \quad \text{(from (4.31))} && (4.48)
\end{aligned}$$

$$\Delta_e = \mathrm{MSE}_L - \mathrm{MSE}_{WL} = \mathrm{tr}(\mathcal{Q}_L - \mathcal{Q}_{WL}) = \mathrm{tr}(\Delta_{\mathcal{Q}}) \tag{4.49}$$

Since $\mathcal{P}^{*-1}_{\mathcal{Y}}$ is a positive definite tensor, hence the pseudo-diagonal entries of $\Delta_{\mathcal{Q}}$ are always non negative. Hence Δ_e is always non negative. The condition when the two estimates are identical and hence provide the same mean square error, i.e. $\Delta_e = 0$, is formally established in the following Lemma.

Lemma 6. *The widely multi-linear and multi-linear MMSE estimates are identical when :*

$$\tilde{\mathcal{C}}_{\mathcal{X}\mathcal{Y}} = \mathcal{C}_{\mathcal{X}\mathcal{Y}} *_M \mathcal{C}^{-1}_{\mathcal{Y}} *_M \tilde{\mathcal{C}}_{\mathcal{Y}}. \tag{4.50}$$

Proof. Substituting (4.50) in (4.29), we get :

$$\begin{aligned}
\mathcal{A}_1 &= (\mathcal{C}_{\mathcal{XY}} - \mathcal{C}_{\mathcal{XY}} *_M \mathcal{C}_{\mathcal{Y}}^{-1} *_M \tilde{\mathcal{C}}_{\mathcal{Y}} *_M \mathcal{C}_{\mathcal{Y}}^{*-1} *_M \tilde{\mathcal{C}}_{\mathcal{Y}}^{*}) *_M (\mathcal{C}_{\mathcal{Y}} - \tilde{\mathcal{C}}_{\mathcal{Y}} *_M \mathcal{C}_{\mathcal{Y}}^{*-1} *_M \tilde{\mathcal{C}}_{\mathcal{Y}}^{*})^{-1} \\
&= \mathcal{C}_{\mathcal{XY}} *_M (\mathcal{I} - \mathcal{C}_{\mathcal{Y}}^{-1} *_M \tilde{\mathcal{C}}_{\mathcal{Y}} *_M \mathcal{C}_{\mathcal{Y}}^{*-1} *_M \tilde{\mathcal{C}}_{\mathcal{Y}}^{*}) *_M (\mathcal{C}_{\mathcal{Y}} *_M (\mathcal{I} - \mathcal{C}_{\mathcal{Y}}^{-1} *_M \tilde{\mathcal{C}}_{\mathcal{Y}} *_M \mathcal{C}_{\mathcal{Y}}^{*-1} *_M \tilde{\mathcal{C}}_{\mathcal{Y}}^{*}))^{-1} \\
&= \mathcal{C}_{\mathcal{XY}} *_M (\mathcal{I} - \mathcal{C}_{\mathcal{Y}}^{-1} *_M \tilde{\mathcal{C}}_{\mathcal{Y}} *_M \mathcal{C}_{\mathcal{Y}}^{*-1} *_M \tilde{\mathcal{C}}_{\mathcal{Y}}^{*}) *_M (\mathcal{I} - \mathcal{C}_{\mathcal{Y}}^{-1} *_M \tilde{\mathcal{C}}_{\mathcal{Y}} *_M \mathcal{C}_{\mathcal{Y}}^{*-1} *_M \tilde{\mathcal{C}}_{\mathcal{Y}}^{*})^{-1} *_M \mathcal{C}_{\mathcal{Y}}^{-1} \\
&= \mathcal{C}_{\mathcal{XY}} *_M \mathcal{C}_{\mathcal{Y}}^{-1}. \qquad (4.51)
\end{aligned}$$

Also, substituting (4.50) in (4.31), we get $\mathcal{A}_2 = 0_{\mathcal{T}}$. In this case, the widely multi-linear estimate is given as $(\mathcal{C}_{\mathcal{XY}} *_M \mathcal{C}_{\mathcal{Y}}^{-1}) *_M \mathcal{Y} + 0_{\mathcal{T}} *_M \mathcal{Y}^*$, which is the multi-linear estimate from (4.38). □

The condition in Lemma 6 essentially represents a case where the error of the multi-linear estimate $(\hat{\mathcal{X}}_L - \mathcal{X})$ is uncorrelated with $\mathcal{Y}^*$, i.e. $\mathbb{E}[(\hat{\mathcal{X}}_L - \mathcal{X}) \circ (\mathcal{Y}^*)^*] = 0_{\mathcal{T}}$. Substituting (4.38) in this condition we get :

$$\mathbb{E}[((\mathcal{C}_{\mathcal{XY}} *_M \mathcal{C}_{\mathcal{Y}}^{-1}) *_M \mathcal{Y} - \mathcal{X}) \circ (\mathcal{Y}^*)^*] = 0_{\mathcal{T}} \qquad (4.52)$$

$$\Rightarrow (\mathcal{C}_{\mathcal{XY}} *_M \mathcal{C}_{\mathcal{Y}}^{-1}) *_M \mathbb{E}[\mathcal{Y} \circ \mathcal{Y}] - \mathbb{E}[\mathcal{X} \circ \mathcal{Y}] = 0_{\mathcal{T}} \qquad (4.53)$$

$$\Rightarrow \mathcal{C}_{\mathcal{XY}} *_M \mathcal{C}_{\mathcal{Y}}^{-1} *_M \tilde{\mathcal{C}}_{\mathcal{Y}} - \tilde{\mathcal{C}}_{\mathcal{XY}} = 0_{\mathcal{T}} \qquad (4.54)$$

which is same as (4.50). A trivial case where this will be satisfied is when the tensor to be estimated and the observation are jointly proper. Tensors $\mathcal{X}$ and $\mathcal{Y}$ are called jointly proper if they are both individually proper, i.e. $\tilde{\mathcal{C}}_{\mathcal{X}} = \tilde{\mathcal{C}}_{\mathcal{Y}} = 0_{\mathcal{T}}$ and cross-proper, i.e. $\tilde{\mathcal{C}}_{\mathcal{XY}} = 0_{\mathcal{T}}$. In this case, (4.50) is satisfied, hence multi-linear estimate is same as widely multi-linear estimate. However, $\mathcal{X}$ and $\mathcal{Y}$ being jointly proper is not a necessary condition, as even if $\mathcal{X}$ is not proper, i.e. $\tilde{\mathcal{C}}_{\mathcal{X}} \neq 0_{\mathcal{T}}$ but $\tilde{\mathcal{C}}_{\mathcal{Y}} = 0_{\mathcal{T}}$ and $\tilde{\mathcal{C}}_{\mathcal{XY}} = 0_{\mathcal{T}}$, still (4.50) is satisfied making the multi-linear and widely multi-linear estimate same.

Further, if the tensor to be estimated $\mathcal{X}$ is real, the cross pseudo-covariance is given as :

$$\tilde{\mathcal{C}}_{\mathcal{XY}} = \mathbb{E}[\mathcal{X} \circ \mathcal{Y}] = \mathbb{E}[(\mathcal{X} \circ \mathcal{Y}^*)^*] = \mathcal{C}_{\mathcal{XY}}^* \quad (\text{since } \mathcal{X} = \mathcal{X}^*). \qquad (4.55)$$

Substituting (4.55) into (4.25) and comparing with (4.26) shows that $\mathcal{A}_2 = \mathcal{A}_1^*$. Thus for real $\mathcal{X}$, the widely multi-linear MMSE estimate and the associated mean square error (from

(4.34)) are given as :

$$\hat{\mathcal{X}}_{WL} = \mathcal{A}_1 *_M \mathcal{Y} + \mathcal{A}_1^* *_M \mathcal{Y}^* \tag{4.56}$$
$$= 2\Re(\mathcal{A}_1 *_M \mathcal{Y}) \tag{4.57}$$
$$= 2\Re((\mathcal{C}_{\mathcal{X}\mathcal{Y}} - \tilde{\mathcal{C}}_{\mathcal{X}\mathcal{Y}} *_M \mathcal{C}_{\mathcal{Y}}^{*-1} *_M \tilde{\mathcal{C}}_{\mathcal{Y}}^*) *_M \mathcal{P}_{\mathcal{Y}}^{-1} *_M \mathcal{Y}) \tag{4.58}$$

which shows that the widely multi-linear estimate of a real signal is always real irrespective of the observation being complex. The corresponding mean square error is given as:

$$\begin{aligned} \mathrm{MSE}_{WL} &= \mathrm{tr}(\mathcal{C}_{\mathcal{X}} - \mathcal{A}_1 *_M \mathcal{C}_{\mathcal{X}\mathcal{Y}}^H - \mathcal{A}_1^* *_M \mathcal{C}_{\mathcal{X}\mathcal{Y}}^{*H}) \\ &= \mathrm{tr}(\mathcal{C}_{\mathcal{X}} - 2\Re(\mathcal{A}_1 *_M \mathcal{C}_{\mathcal{X}\mathcal{Y}}^H)) \\ &= \mathrm{tr}(\mathcal{C}_{\mathcal{X}} - 2\Re((\mathcal{C}_{\mathcal{X}\mathcal{Y}} - \tilde{\mathcal{C}}_{\mathcal{X}\mathcal{Y}} *_M \mathcal{C}_{\mathcal{Y}}^{*-1} *_M \tilde{\mathcal{C}}_{\mathcal{Y}}^*) *_M \mathcal{P}_{\mathcal{Y}}^{-1} *_M \mathcal{C}_{\mathcal{X}\mathcal{Y}}^H)). \end{aligned} \tag{4.59}$$

For real or proper complex vectors, it is well known that if the signal to be estimated $\underline{\mathbf{x}}$ and the observed vector $\underline{\mathbf{y}}$ are jointly Gaussian random vectors, then the best MMSE estimate is same as the Linear Minimum Mean Squared Error (LMMSE) estimate. However, to be more accurate, we can say that if $\underline{\mathbf{x}}$ and $\underline{\mathbf{y}}$ are two jointly complex Gaussian random vectors, then the best MMSE estimator $\mathbb{E}[\underline{\mathbf{x}}|\underline{\mathbf{y}}]$ is the WLMMSE estimator of $\underline{\mathbf{x}}$ from $\underline{\mathbf{y}}$ [122]. We can extend this result to widely multi-linear MMSE estimate in the form of the following theorem.

Theorem 4. *If $\mathcal{X}$ and $\mathcal{Y}$ are two jointly complex Gaussian tensors, then the best MMSE estimator $\mathbb{E}[\mathcal{X}|\mathcal{Y}]$ is the widely multi-linear estimator of $\mathcal{X}$ from $\mathcal{Y}$.*

A proof of Theorem 4 has been included in Appendix B.5.

4.1.4 Comparison with Tucker based Tensor MMSE filter

The n-mode Wiener filter which makes use of the Tucker product [186] is a commonly used signal processing technique for tensors. Consider an observed tensor $\mathcal{Y} \in \mathbb{C}^{I_1 \times \ldots \times I_N}$. The objective is to estimate $\mathcal{X} \in \mathbb{C}^{I_1 \times \ldots \times I_N}$ based on the observation $\mathcal{Y}$. Note that such class of estimators often assume that the signal to be estimated and observed signals have the same dimensions. The estimator of the signal tensor $\mathcal{X}$ can be represented by N n-mode filters represented via $\mathrm{A}^{(n)} \in \mathbb{C}^{I_n \times I_n}$ using the Tucker product as follows [187]:

$$\hat{\mathcal{X}}_T = \mathcal{Y} \times_1 \mathrm{A}^{(1)} \times_2 \mathrm{A}^{(2)} \times_3 \ldots \times_N \mathrm{A}^{(N)} \tag{4.60}$$

where the criterion for obtaining the optimal n-mode filters $\mathrm{A}^{(n)}$ is the minimization of the mean squared error between $\mathcal{X}$ and $\hat{\mathcal{X}}_T$, defined as :

$$e(\mathrm{A}^{(1)}, \mathrm{A}^{(2)}, \ldots, \mathrm{A}^{(N)}) = \mathbb{E}[||\mathcal{X} - \hat{\mathcal{X}}_T||^2] \tag{4.61}$$

$$= \mathbb{E}[||\mathcal{X} - \mathcal{Y} \times_1 \mathrm{A}^{(1)} \times_2 \mathrm{A}^{(2)} \times_3 \ldots \times_N \mathrm{A}^{(N)}||^2]. \tag{4.62}$$

The optimal choice of the n-mode filter matrices $\mathrm{A}^{(n)}$ which ensures minimum mean-squared error between $\mathcal{X}$ and $\hat{\mathcal{X}}_T$ is calculated using n-mode Wiener filtering method which relies on matrix unfoldings of $\mathcal{Y}$ [186]. Initially, all the factor matrices $\mathrm{A}^{(n)}$ are set to identity matrices. For updating $\mathrm{A}^{(n)}$ for each n, it is assumed that $\mathrm{A}^{(m)}$ for $m \neq n$ are known. An alternative least squares method is used to calculate all the optimal $\mathrm{A}^{(n)}$ where $\mathrm{A}^{(m)}$ for $m \neq n$ is fixed to find $\mathrm{A}^{(n)}$ for all n, and then we repeat for all n until a convergence criterion is met. A detailed derivation of the solution and the algorithm to calculate the optimal n-mode matrix filters is presented in [187, 186]. To the best of our knowledge, the Tucker operator based estimator is presented only for the multi-linear case in the literature but not for the widely multi-linear case. However a simple extension to a widely multi-linear case using Tucker operator would require finding optimal factor matrices $\mathrm{B}^{(n)} \in \mathbb{C}^{I_n \times I_n}$ to operate on the conjugate of $\mathcal{Y}$ as well, i.e.

$$\hat{\mathcal{X}}_{WT} = \mathcal{Y} \times_1 \mathrm{A}^{(1)} \times_2 \ldots \times_N \mathrm{A}^{(N)} + \mathcal{Y}^* \times_1 \mathrm{B}^{(1)} \times_2 \ldots \times_N \mathrm{B}^{(N)} \tag{4.63}$$

such that the mean square error between $\mathcal{X}$ and $\hat{\mathcal{X}}_{WT}$ is minimized. Notice that both (4.60) and (4.63) can be seen as special cases of the widely multi-linear estimator from (4.17). Hence the Tucker product based estimator can be seen as a specific case of the MMSE estimator using the Einstein product with an additional constraint. On writing (4.60) element-wise,

$$\hat{\mathcal{X}}_{Ti_1,\ldots,i_N} = \sum_{j_N=1}^{I_N} \cdots \sum_{j_1=1}^{I_1} \mathcal{Y}_{j_1,\ldots,j_N} \cdot \mathrm{A}^{(1)}_{i_1,j_1} \cdot \mathrm{A}^{(2)}_{i_2,j_2} \cdots \mathrm{A}^{(N)}_{i_N,j_N}. \tag{4.64}$$

We define a tensor $\mathcal{A} \in \mathbb{C}^{I_1 \times \ldots \times I_N \times I_1 \times \ldots \times I_N}$ such that

$$\mathcal{A}_{i_1,\ldots,i_N,j_1,\ldots,j_N} = \mathrm{A}^{(1)}_{i_1,j_1} \cdot \mathrm{A}^{(2)}_{i_2,j_2} \cdots \mathrm{A}^{(N)}_{i_N,j_N} \tag{4.65}$$

In this case we can re-write (4.64) as :

$$\hat{\mathcal{X}}_{Ti_1,\ldots,i_N} = \sum_{j_N=1}^{I_N} \cdots \sum_{j_1=1}^{I_1} \mathcal{Y}_{j_1,\ldots,j_N} \cdot \mathcal{A}_{i_1,\ldots,i_N,j_1,\ldots,j_N}$$
$$\Rightarrow \hat{\mathcal{X}}_T = \mathcal{A} *_N \mathcal{Y} \tag{4.66}$$

The solution for the optimal tensor $\mathcal{A}$ which minimizes the mean square-error between $\mathcal{X}$ and $\hat{\mathcal{X}}_T$ in (4.66) is the multi-linear MMSE estimator as given by (4.37). Similarly (4.63) can be equivalently written as (4.17) with constraints $\mathcal{A}_{1i_1,\ldots,i_N,j_1,\ldots,j_N} = \mathrm{A}^{(1)}_{i_1,j_1} \cdots \mathrm{A}^{(N)}_{i_N,j_N}$ and $\mathcal{A}_{2i_1,\ldots,i_N,j_1,\ldots,j_N} = \mathrm{B}^{(1)}_{i_1,j_1} \cdots \mathrm{B}^{(N)}_{i_N,j_N}$. Hence the Tucker multi-linear MMSE estimator can be seen as a special case of the Einstein product based multi-linear MMSE estimator with an additional constraint that the tensor $\mathcal{A}$ can be written as in (4.65). Similarly, the Tucker widely multi-linear MMSE estimator can be seen as a constrained case of the widely multi-linear MMSE estimator from (4.17). The constraint (4.65) expresses the tensor $\mathcal{A}$ as a rearranged outer product of N factor matrices $\mathrm{A}^{(n)}$. This extra constraint makes the performance of Tucker operator based estimators sub-optimal within the class of multi-linear estimators. The Tucker or the n-mode filtering approach aims to find factor matrices along each mode separately, and thereby assumes that the optimal multi-linear estimator can be expressed using the product of such factor matrix elements. But the proposed estimator based on the Einstein product finds the best multi-linear estimator with no constraints or assumptions regarding separability of the estimator across different modes, thereby providing better mean square error performance.

Complexity Analysis: Even though Tucker operator based estimator is sub-optimal, it has a computational advantage over the more general multi-linear estimator using the Einstein product. Finding the estimator for multi-linear MMSE estimation of order N tensor requires inverting the covariance tensor of size $I_1 \times I_2 \times \ldots I_N \times I_1 \times I_2 \times \ldots \times I_N$, which has a computational complexity of $\mathcal{O}((I_1 \cdots I_N)^3)$. The Tucker approach while admitting sub-optimality breaks the tensor estimation problem into N smaller linear estimation problems which requires inverting N matrices of size $I_n \times I_n$ for $n = 1, \ldots, N$ where the complexity of each matrix inversion is $\mathcal{O}(I_n^3)$ [80]. The computational cost for the two methods is presented in Table 4.1. Hence, as N increases, the Tucker method provides a low complexity solution but with sub-optimal performance. In section 4.2.2 we present numerical examples illustrating the loss of performance due to the sub-optimality of Tucker approach as N grows, and compare it with the Einstein product approach.

Table 4.1: Complexity comparison for tensor estimation.

Method	Complexity
Using Tucker product	$\mathcal{O}(I_1^3 + I_2^3 + \cdots + I_N^3)$
Using Einstein product	$\mathcal{O}((I_1 \cdot I_2 \cdots I_N)^3)$

Since the complexity of the Einstein product approach primarily depends on the tensor inversion operation, approaches discussed in Appendix B.7 can be employed to reduce the time complexity of finding the tensor inverse.

4.2 Applications of Tensor MMSE Estimation

A standard task for a receiver in a communication system is to estimate the transmitted signal based on the noisy observation received through the channel. Hence for a multi-domain communication system where the input and output signals are modelled using tensors as in (2.35), the tensor based MMSE estimation techniques developed in this chapter can be employed at the receiver.

For a known channel $\mathcal{H}$, assuming $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{N}}$ to be independent and zero mean, using the steps for deriving (3.4), we can write the received covariance, cross covariance, received pseudo-covariance and cross pseudo-covariance tensors as :

$$\mathcal{C}_{\boldsymbol{\mathcal{Y}}} = \mathcal{H} *_N \mathcal{C}_{\boldsymbol{\mathcal{X}}} *_N \mathcal{H}^H + \mathcal{C}_{\boldsymbol{\mathcal{N}}} \tag{4.67}$$

$$\mathcal{C}_{\boldsymbol{\mathcal{X}}\boldsymbol{\mathcal{Y}}} = \mathcal{C}_{\boldsymbol{\mathcal{X}}} *_N \mathcal{H}^H \tag{4.68}$$

$$\tilde{\mathcal{C}}_{\boldsymbol{\mathcal{Y}}} = \mathcal{H} *_N \tilde{\mathcal{C}}_{\boldsymbol{\mathcal{X}}} *_N \mathcal{H}^T + \tilde{\mathcal{C}}_{\boldsymbol{\mathcal{N}}} \tag{4.69}$$

$$\tilde{\mathcal{C}}_{\boldsymbol{\mathcal{X}}\boldsymbol{\mathcal{Y}}} = \tilde{\mathcal{C}}_{\boldsymbol{\mathcal{X}}} *_N \mathcal{H}^T \tag{4.70}$$

respectively. Here $\mathcal{C}_{\boldsymbol{\mathcal{X}}}, \tilde{\mathcal{C}}_{\boldsymbol{\mathcal{X}}}, \mathcal{C}_{\boldsymbol{\mathcal{N}}},$ and $\tilde{\mathcal{C}}_{\boldsymbol{\mathcal{N}}}$ denote the input covariance, input pseudo-covariance, noise covariance and noise pseudo-covariance tensors respectively. Substituting (4.67) and

(4.68) into (4.38) and (4.41) gives the receiver structure based on multi-linear MMSE estimation and the associated mean square error respectively. Similarly substituting (4.67), (4.68), (4.69) and (4.70) into (4.29) and (4.31) gives us tensors $\mathcal{A}_1$ and $\mathcal{A}_2$ which on substituting into (4.17) and (4.34) yields the receiver structure based on widely multi-linear MMSE estimation and the associated mean square error respectively.

Let us assume that the transmitted tensor contains independent elements normalized to unit power such that $\mathcal{C}_{\boldsymbol{\mathcal{X}}} = \mathcal{I}_N$ which is an identity tensor of size $I_1 \times \ldots \times I_N \times I_1 \times \ldots \times I_N$. Let the noise be additive circular Gaussian noise with mean zero and variance σ_n^2, such that $\mathcal{C}_{\boldsymbol{\mathcal{N}}} = \sigma_n^2 \mathcal{I}_M$ and $\tilde{\mathcal{C}}_{\boldsymbol{\mathcal{N}}} = 0_{\mathcal{T}}$. Hence the mean square error from widely multi-linear estimation (based on (4.34)) can be given as :

$$\begin{aligned}\text{MSE} = &\operatorname{tr}(\mathcal{I}_N) - \operatorname{tr}\Big((\mathcal{H}^H - \tilde{\mathcal{C}}_{\boldsymbol{\mathcal{X}}} *_N \mathcal{H}^T *_M (\mathcal{H} *_N \mathcal{H}^H + \sigma_n^2 \mathcal{I}_M)^{*-1} *_M (\mathcal{H} *_N \tilde{\mathcal{C}}_{\boldsymbol{\mathcal{X}}} *_N \mathcal{H}^T)^*) *_M \\ &\mathcal{P}_{\boldsymbol{\mathcal{Y}}}^{-1} *_M \mathcal{H}\Big) - \operatorname{tr}\Big((\tilde{\mathcal{C}}_{\boldsymbol{\mathcal{X}}} *_N \mathcal{H}^T - \mathcal{H}^H *_M (\mathcal{H} *_N \mathcal{H}^H + \sigma_n^2 \mathcal{I}_M)^{-1} *_M \mathcal{H} *_N \tilde{\mathcal{C}}_{\boldsymbol{\mathcal{X}}} *_N \mathcal{H}^T) *_M \\ &\mathcal{P}_{\boldsymbol{\mathcal{Y}}}^{*-1} *_M \mathcal{H}^* *_N \tilde{\mathcal{C}}_{\boldsymbol{\mathcal{X}}}^H\Big) \end{aligned} \tag{4.71}$$

where $\mathcal{P}_{\boldsymbol{\mathcal{Y}}}$ is given by (4.30). The MSE performance difference between receivers employing widely multi-linear and multi-linear estimators in such multi-domain communication systems can be found using (4.49) as :

$$\begin{aligned}\Delta_e = \operatorname{tr}\Big[&\Big(\tilde{\mathcal{C}}_{\boldsymbol{\mathcal{X}}} *_N \mathcal{H}^T - \mathcal{H}^H *_M (\mathcal{H} *_N \mathcal{H}^H + \sigma_n^2 \mathcal{I}_M)^{-1} *_M \mathcal{H} *_N \tilde{\mathcal{C}}_{\boldsymbol{\mathcal{X}}} *_N \mathcal{H}^T\Big) *_M \mathcal{P}_{\boldsymbol{\mathcal{Y}}}^{*-1} *_M \\ &\Big(\tilde{\mathcal{C}}_{\boldsymbol{\mathcal{X}}} *_N \mathcal{H}^T - \mathcal{H}^H *_M (\mathcal{H} *_N \mathcal{H}^H + \sigma_n^2 \mathcal{I}_M)^{-1} *_M \mathcal{H} *_N \tilde{\mathcal{C}}_{\boldsymbol{\mathcal{X}}} *_N \mathcal{H}^T\Big)^H\Big]. \end{aligned} \tag{4.72}$$

Notice that if $\tilde{\mathcal{C}}_{\boldsymbol{\mathcal{X}}} = 0_{\mathcal{T}}$ which is when $\boldsymbol{\mathcal{X}}$ is proper, then Δ_e is also zero as widely multi-linear estimator reduces to a multi-linear estimator in this case. The MSE expression from (4.71) simplifies to :

$$\text{MSE} = \operatorname{tr}\left(\mathcal{I}_N - \mathcal{H}^H *_M (\mathcal{H} *_N \mathcal{H}^H + \sigma_n^2 \mathcal{I}_M)^{-1} *_M \mathcal{H}\right) \tag{4.73}$$

which is same as the MSE from multi-linear estimation (from (4.41)) with $\mathcal{C}_{\boldsymbol{\mathcal{X}}\boldsymbol{\mathcal{Y}}} = \mathcal{H}^H$ and $\mathcal{C}_{\boldsymbol{\mathcal{Y}}} = (\mathcal{H} *_N \mathcal{H}^H + \sigma_n^2 \mathcal{I}_M)$.

In the subsequent sections, we present numerical examples to illustrate the concept of Tensor multi-linear (TL) and Tensor widely multi-linear (TWL) estimators in the context of multi-domain communication systems. We use (4.17) at the receiver for TWL estimation with $\mathcal{A}_1$ and $\mathcal{A}_2$ calculated from (4.29) and (4.31). Also, (4.38) is used at the receiver for TL estimation. The received covariance, cross covariance, pseudo-covariance and cross pseudo-covariance are calculated based on (4.67)-(4.70). The elements of input tensor are generated with zero mean variance σ_s^2 and noise tensor with zero mean variance σ_n^2. The SNR is defined as σ_s^2/σ_n^2. For all the examples, we keep $\sigma_s^2 = 1$ and vary σ_n^2 to attain different SNR values. We use Monte Carlo simulations where input, channel and noise are randomly generated with channel realizations known at the receiver. The results are averaged over $N_{ch} = 500$ channel realizations for each SNR, with $N_{in} = 100$ noise and input realizations for each channel realization. The performance is evaluated in terms of mean square error normalized with respect to the number of transmit tensor elements, i.e.

$$\text{MSE} = \frac{1}{N_{ch}} \frac{1}{N_{in}} \sum_{k=1}^{N_{ch}} \sum_{l=1}^{N_{in}} \frac{||\mathbf{\mathfrak{X}}^{(k,l)} - \hat{\mathbf{\mathfrak{X}}}^{(k,l)}||^2}{\text{numel}[\mathbf{\mathfrak{X}}^{(k,l)}]} \tag{4.74}$$

where $\mathbf{\mathfrak{X}}^{(k,l)}$ and $\hat{\mathbf{\mathfrak{X}}}^{(k,l)}$ denote the actual and the estimated tensors at the kth channel and lth input run for a fixed SNR. The number of transmit elements are denoted by numel$[\mathbf{\mathfrak{X}}^{(k,l)}]$.

4.2.1 Example with Gaussian input signals

In (2.35), let the input tensor, $\mathbf{\mathfrak{X}} \in \mathbb{C}^{4\times4\times4}$ contain elements drawn from i.i.d. zero mean unit variance improper complex Gaussian distribution. Let each element of $\mathbf{\mathfrak{X}}$ be denoted by $x = a + ib$ where a and b are real scalar random variables. To generate a and b for simulation such that they are correlated with coefficient ρ, we consider the vector $\underline{\mathbf{p}} = [a, b]^T$ and its correlation matrix $\mathrm{R} = [\mathrm{var}(a), \mathrm{cov}(a, b); \mathrm{cov}(b, a), \mathrm{var}(b)]$, where $\mathrm{cov}(a, b) = \rho\sqrt{\mathrm{var}(a)\mathrm{var}(b)}$. The Cholesky decomposition of R is given as $\mathrm{R} = \mathrm{LL}^T$. We generate vector $\underline{\mathbf{q}} = [c, d]^T$ where c and d are uncorrelated zero mean and variance 1/2 Gaussian scalars. Then $\underline{\mathbf{p}} = \mathrm{L}\underline{\mathbf{q}}$ generates a vector with entries a and b such that $\mathrm{var}(a) = \mathrm{var}(b) = 1/2$ and they are correlated with coefficient ρ. All the elements of $\mathbf{\mathfrak{X}}$ designated as $x = a + ib$ are generated independently using this Cholesky Decomposition method. Hence input pseudo-covariance $\tilde{\mathfrak{C}}_{\mathbf{\mathfrak{X}}}$ is a pseudo-diagonal tensor where all its non-zero entries are $\mathbb{E}[x^2] = \mathrm{var}(a) - \mathrm{var}(b) + i\cdot 2\mathrm{cov}(a, b)$. Thus for different ρ, we get different $\tilde{\mathfrak{C}}_{\mathbf{\mathfrak{X}}}$. The input covariance is $\mathfrak{C}_{\mathbf{\mathfrak{X}}} = \sigma_s^2 \mathfrak{I}$ where $\mathfrak{I}$ is an order 6 identity tensor and $\sigma_s^2 = \mathrm{var}(a) + \mathrm{var}(b) = 1$. Furthermore

$\mathcal{N} \in \mathbb{C}^{4\times4\times4}$ is an order three received noise tensor with zero mean proper complex Gaussian entries and independent of input signal with covariance $\mathcal{C}_{\mathcal{N}} = \sigma_n^2 \mathcal{I}$ and pseudo-covariance $\tilde{\mathcal{C}}_{\mathcal{N}} = 0_{\mathcal{T}}$. The channel $\mathcal{H} \in \mathbb{C}^{4\times4\times4\times4\times4\times4}$ contains i.i.d. zero mean unit variance proper complex Gaussian entries. The objective is to estimate $\mathcal{X}$ based on the observation $\mathcal{Y}$.

Figure 4.1 plots the MSE against ρ at 10 dB SNR. It can be seen that for a low magnitude of ρ, i.e when the signal is close to being proper, both TL and TWL estimation results in almost same mean squared error, but as the magnitude of ρ increases the TWL estimator performs much better than the TL estimator. The mean squared error essentially remains flat for TL estimation when changing ρ as it does not take into account the pseudo-covariance. As observed in Figure 4.1, the mean squared error follows quite well the theoretical mean squared error calculated from (4.41) and (4.34), which validates our simulation set up. Further, Figure 4.2 presents the mean squared error against SNR for specific values of ρ. With increase in SNR, mean squared error reduces but the TWL estimator performs much better than the TL estimator for higher values of ρ. The TL estimator performance does not change with ρ as it does not depend on the pseudo-covariance. However, for a given SNR the TWL estimator performance improves as ρ increases since it uses the correlation between the real and imaginary components of the tensor for estimation.

4.2.2 Example of Tucker based MMSE Estimation

We now compare the Tucker product based MMSE estimator with the Einstein product based multi-linear MMSE estimator. In a multi-domain communication system, consider input $\mathcal{X}$, output $\mathcal{Y}$ and noise $\mathcal{N}$ are order N tensors where dimension of each individual domain is 3. Hence the channel $\mathcal{H}$ is an order $2N$ tensor where all its modes have dimension 3. We assume that the channel can be written in terms of N factor matrices $\mathrm{H}^{(1)}, \mathrm{H}^{(2)}, \ldots, \mathrm{H}^{(N)}$ of size 3×3 each as in (4.65). Hence the system model can be written in two equivalent forms as :

$$\mathcal{Y} = \mathcal{X} \times_1 \mathrm{H}^{(1)} \times_2 \mathrm{H}^{(2)} \times_3 \ldots \times_N \mathrm{H}^{(N)} + \mathcal{N} \tag{4.75}$$

$$\mathcal{Y} = \mathcal{H} *_N \mathcal{X} + \mathcal{N} \tag{4.76}$$

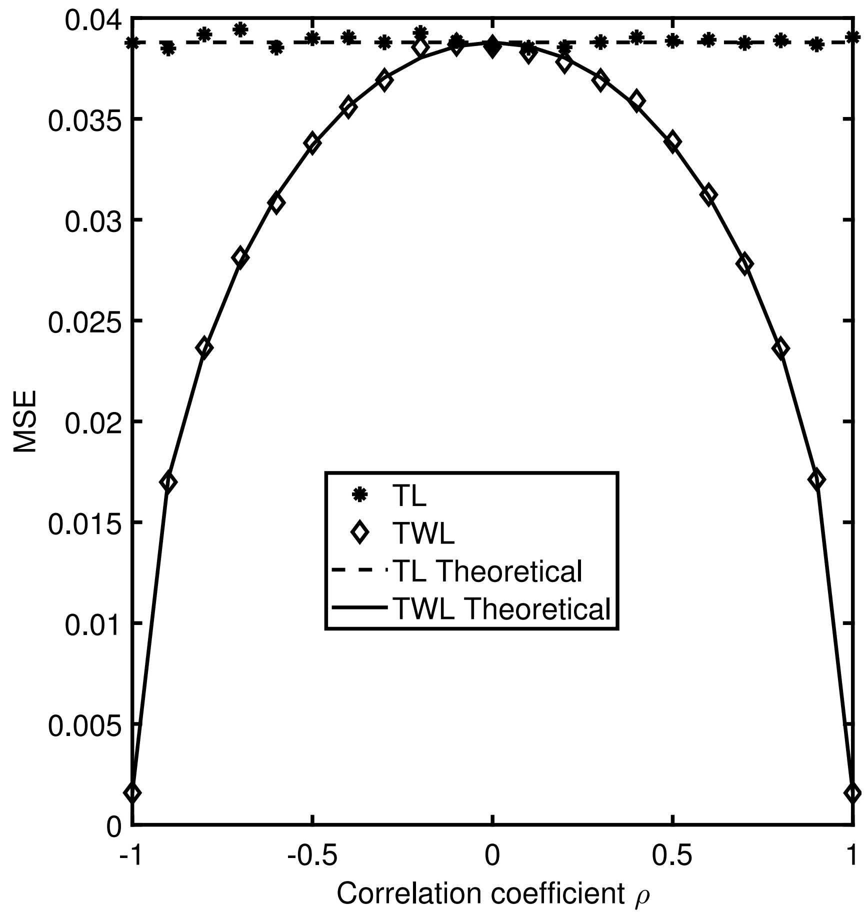


Fig. 4.1: MSE vs correlation coefficient between real and imaginary parts of Gaussian input at 10 dB SNR.

where $\mathcal{H}_{i_1,i_2,\ldots,i_N,j_1,j_2,\ldots,j_N} = \mathrm{H}^{(1)}_{i_1,j_1} \cdot \mathrm{H}^{(2)}_{i_2,j_2} \cdots \mathrm{H}^{(N)}_{i_N,j_N}$. The equivalence of these system models can be established by writing the equations element-wise as shown in (4.64)-(4.66). For simulations, $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{N}}$ are generated using circularly symmetric complex Gaussian distribution with covariance as scaled identity tensor $\sigma_s^2\mathcal{I}$ and $\sigma_n^2\mathcal{I}$ respectively, with $\sigma_s^2 = 1$. The components of $\mathrm{H}^{(1)}, \mathrm{H}^{(2)}, \ldots, \mathrm{H}^{(N)}$ are i.i.d. drawn from proper complex Gaussian distribution with zero mean and unit variance. The objective is to estimate $\boldsymbol{\mathcal{X}}$ based on observing $\boldsymbol{\mathcal{Y}}$. The Tucker operator based estimator finds factor matrices $\mathrm{A}^{(1)}, \mathrm{A}^{(2)}, \ldots, \mathrm{A}^{(N)}$ such that the estimate is given by (4.60). For the channel model (4.75) the factor matrices $\mathrm{A}^{(n)}$ are

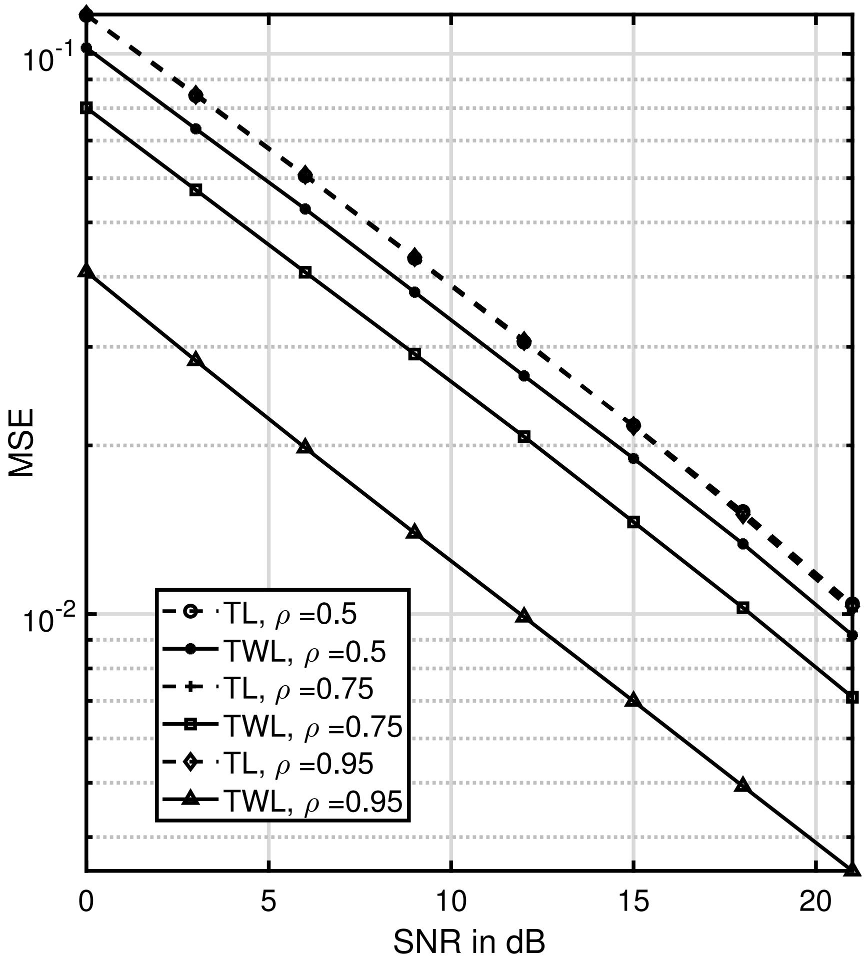


Fig. 4.2: MSE vs SNR with Gaussian input for different ρ.

given as [80] :

$$\mathrm{A}^{(n)} = \mathrm{R}_{\mathcal{X}}^{(n)}\mathrm{H}^{(n)H}(\mathrm{H}^{(n)}\mathrm{R}_{\mathcal{X}}^{(n)}\mathrm{H}^{(n)H} + \mathrm{R}_{\mathcal{N}}^{(n)})^{-1} \tag{4.77}$$

for $n = 1, 2, \ldots, N$ where $\mathrm{R}_{\mathcal{X}}^{(n)} = \mathbb{E}[\mathbf{X}_{(n)}\mathbf{X}_{(n)}^H]$ and $\mathrm{R}_{\mathcal{N}}^{(n)} = \mathbb{E}[\mathbf{N}_{(n)}\mathbf{N}_{(n)}^H]$. The quantities $\mathbf{X}_{(n)}$ and $\mathbf{N}_{(n)}$ are the n-mode matrix unfoldings of $\mathcal{X}$ and $\mathcal{N}$ respectively. The mean square error achieved by this estimator is compared with the mean square error achieved by the Einstein product based multi-linear MMSE estimator of (4.35) where $\mathcal{A}$ is given by (4.37). The MSE plot is presented in Figure 4.3 for three different values of $N = 2, 3, 4$.

The solid lines in Figure 4.3 correspond to the MSE when the Einstein product method is used, and the dashed lines correspond to the MSE when the Tucker method is used.

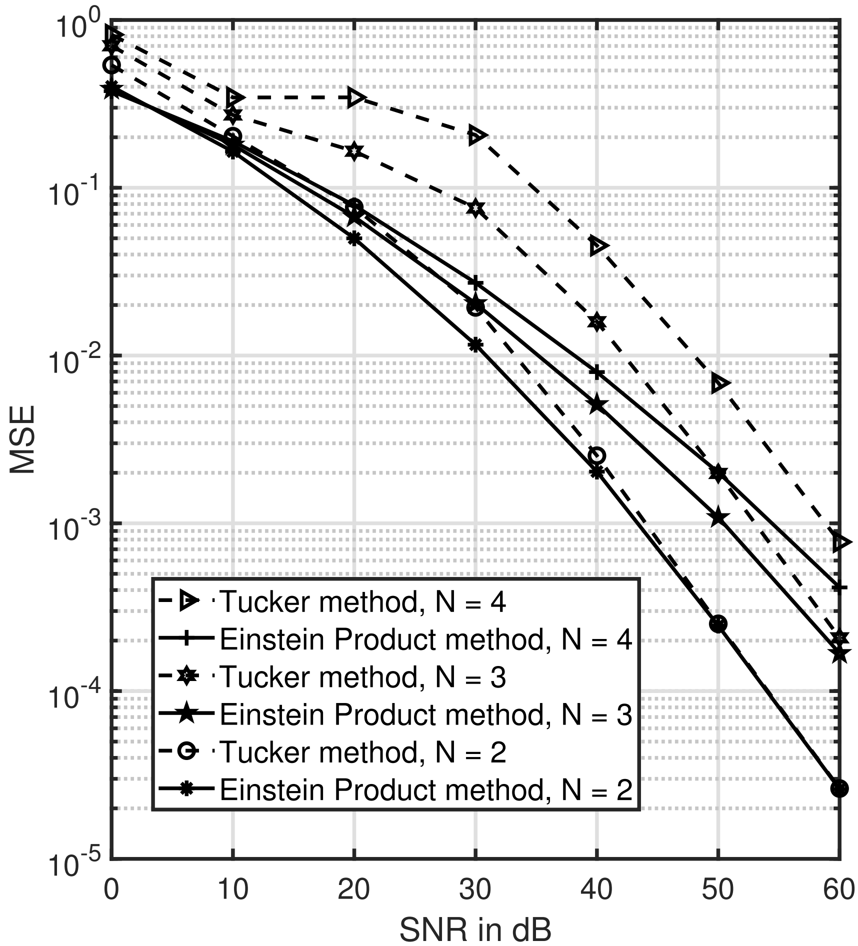


Fig. 4.3: MSE vs SNR for different estimation techniques.

It can be observed that in all the three cases the multi-linear MMSE estimator based on the Einstein product achieves lower mean squared error than the Tucker operator based estimator, and they perform similar at high SNR. Further, it can be observed that as N increases, the difference between the performance of Tucker method and Einstein product method also widens. For $N = 2$, the performance gap between the two cases is small especially at high SNR, but for $N = 4$ the gap is significant. This shows that for higher order tensors, assuming the multi-linear estimator to be separable across all the domains makes the performance more sub-optimal. This is further apparent in Figure 4.4 where the MSE performance for the two methods is plotted against N for a fixed SNR of 30 dB. For $N = 1$, both the methods reduce to standard vector based LMMSE solution, hence they perform exactly same. As can be observed in Figure 4.4, with increasing N the performance of the Tucker approach can be substantially worse than the Einstein product method. However, it is to be noted that the computational complexity of Tucker approach is less than the Einstein product method. Hence there is an inherent trade-off between the Einstein product method and the Tucker approach where the former provides much better performance, but the latter has lower complexity.

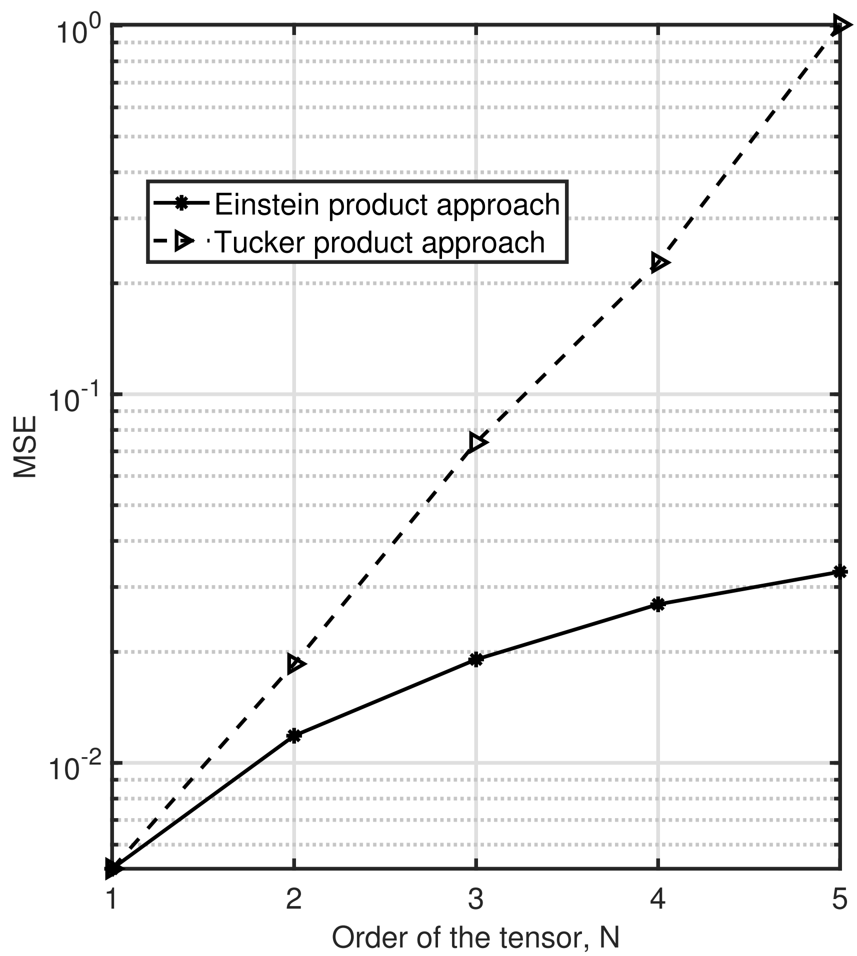


Fig. 4.4: MSE vs tensor order for different estimation techniques at 30 dB SNR.

4.2.3 Estimation of Tensors in TT format

The number of elements in a tensor grows exponentially with its number of domains. In addition, the dimensionality of each domain may be large as well. Hence tensor-based systems often generate large data which pose a challenge on their storage complexity. For instance, in modern multi-domain communication systems, the dimensions of domains such as space (antenna) or frequency (sub-carriers) can be significantly large in a massive MIMO or multi-carrier scheme. However, with the help of tensor tools such as TT decomposition, effective storage mechanisms have been proposed in literature for large data. The TT decomposition approach is not only used when the data has a natural multi-way structure, but also when the data to be stored is a large vector or matrix. In such cases, the large matrix or vector is converted to a tensor for storage efficiency through Tensorization [57] (refer to section 2.1.3 for details on TT decomposition). The tensor-based technique presented in this chapter offers itself as an effective mechanism for data estimation when tensors are expressed and stored in TT format. This is because the MMSE estimation methods proposed in this chapter do not rely on any tensor to vector/matrix transformation, but make use of the Einstein Product.

An algorithm to compute the Einstein Product for tensors in TT format without reconstructing the whole tensor is presented in [120]. Through an example, we now show the application of tensor MMSE estimator for tensor stored in TT format. Consider the system model from (2.35) where $\boldsymbol{\mathcal{Y}}, \boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{N}}$ are order 3 tensors of size $5 \times 5 \times 5$ each and $\mathcal{H}$ is an order 6 tensor of size $5 \times 5 \times 5 \times 5 \times 5 \times 5$. We use the multi-linear MMSE tensor $\mathcal{A}$ from (4.37) to find the estimate $\hat{\boldsymbol{\mathcal{X}}}$ at the receiver. For the numerical example, $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{N}}$ are generated using i.i.d. zero mean proper complex Gaussian distribution with covariance as $\sigma_s^2 \mathcal{I}$ and $\sigma_n^2 \mathcal{I}$ respectively, with $\sigma_s^2 = 1$. The channel contains i.i.d. zero mean unit variance complex Gaussian entries and is normalized to provide unit power gain at the receiver. We show the MSE results when the observed tensor $\boldsymbol{\mathcal{Y}}$ and the multi-linear MMSE tensor $\mathcal{A}$ are instead stored in their TT formats $\bar{\boldsymbol{\mathcal{Y}}}$ and $\bar{\mathcal{A}}$ at the receiver with accuracy ϵ. Algorithm 1 from [62] is employed to calculate the TT decompositions. The estimation in tensor's original format is given as $\hat{\boldsymbol{\mathcal{X}}} = \mathcal{A} *_3 \boldsymbol{\mathcal{Y}}$. However, since we assume $\boldsymbol{\mathcal{Y}}$ and $\mathcal{A}$ are stored in

TT formats, we find the estimate $\hat{\mathcal{X}}$ also in TT format denoted as $\bar{\hat{\mathcal{X}}}$ using $\bar{\mathcal{Y}}$ and $\bar{\mathcal{A}}$ as :

$$\bar{\hat{\mathcal{X}}} = \bar{\mathcal{A}} *_3 \bar{\mathcal{Y}} \tag{4.78}$$

which can be written in terms of their cores as :

$$\bar{\hat{\mathcal{X}}}_{i,j,k} = \sum_{r_0,r_1,r_2,r3} \hat{\mathcal{X}}^{(1)}_{r_0,i,r_1} \cdot \hat{\mathcal{X}}^{(2)}_{r_1,j,r_2} \cdot \hat{\mathcal{X}}^{(3)}_{r_2,k,r_3} \tag{4.79}$$

$$= (\bar{\mathcal{A}} *_3 \bar{\mathcal{Y}})_{i,j,k} = \sum_{l,m,n} \bar{\mathcal{A}}_{i,j,k,l,m,n} \cdot \bar{\mathcal{Y}}_{l,m,n} \tag{4.80}$$

$$= \sum_{l,m,n} \Big(\sum_{p_0,p_1,\ldots,p_6} \mathcal{A}^{(1)}_{p_0,i,p_1} \cdot \mathcal{A}^{(2)}_{p_1,j,p_2} \cdot \mathcal{A}^{(3)}_{p_2,k,p_3} \cdots \mathcal{A}^{(6)}_{p_5,n,p_6} \Big) \cdot \Big(\sum_{s_0,s_1,s_2,s3} \mathcal{Y}^{(1)}_{s_0,l,s_1} \cdot \mathcal{Y}^{(2)}_{s_1,m,s_2} \cdot \mathcal{Y}^{(3)}_{s_2,n,s_3} \Big). \tag{4.81}$$

The objective is to estimate the TT cores $\hat{\mathcal{X}}^{(1)}, \hat{\mathcal{X}}^{(2)}$ and $\hat{\mathcal{X}}^{(3)}$ using the TT cores of $\bar{\mathcal{Y}}$ and $\bar{\mathcal{A}}$ without explicitly constructing the whole tensor at any stage. We use Algorithm 1 from [120] for this purpose which finds the TT cores of $\bar{\hat{\mathcal{X}}}$ using the TT cores of $\bar{\mathcal{Y}}$ and $\bar{\mathcal{A}}$. From the estimated cores we reconstruct $\bar{\hat{\mathcal{X}}}$ and compare it with the transmitted tensor $\mathcal{X}$ to calculate the mean squared error in estimation. Figure 4.5 presents the MSE against SNR in dB for different accuracy ϵ with which TT decomposition is calculated. We compare this result with the MSE achieved when all the tensors are taken in their original non-decomposed formats, labelled as 'original case' in the figure. We can observe that for lower value of ϵ such as $\epsilon = 0.01$, the MSE achieved by the multi-linear MMSE estimation performed on tensors in TT format is indistinguishable from the case when tensors are used in original format. The performance degradation in MSE is observed at high SNR as ϵ increases. A small value of ϵ indicates an almost exact TT decomposition whereas a larger value of ϵ indicates a larger approximation error in the TT decomposed format. Further, Figure 4.6 presents the MSE against the accuracy ϵ for a fixed SNR of 20 dB. The MSE for the original case is unaffected by ϵ. For estimation in TT format, the MSE is almost same as the original case for small ϵ but after a certain value (> 0.03 in this case), the MSE increases. This increase in MSE is due to the approximation tolerance set in the computation of TT format by fixing ϵ. Hence multi-linear MMSE estimation can be used for tensors stored in TT format, and its MSE performance remains the same as that of the original case if the TT decomposition is computed with high accuracy, i.e. low ϵ. Also, since the Einstein product can be implemented directly on the cores, the proposed estimation technique does not require reconstructing the original tensors.

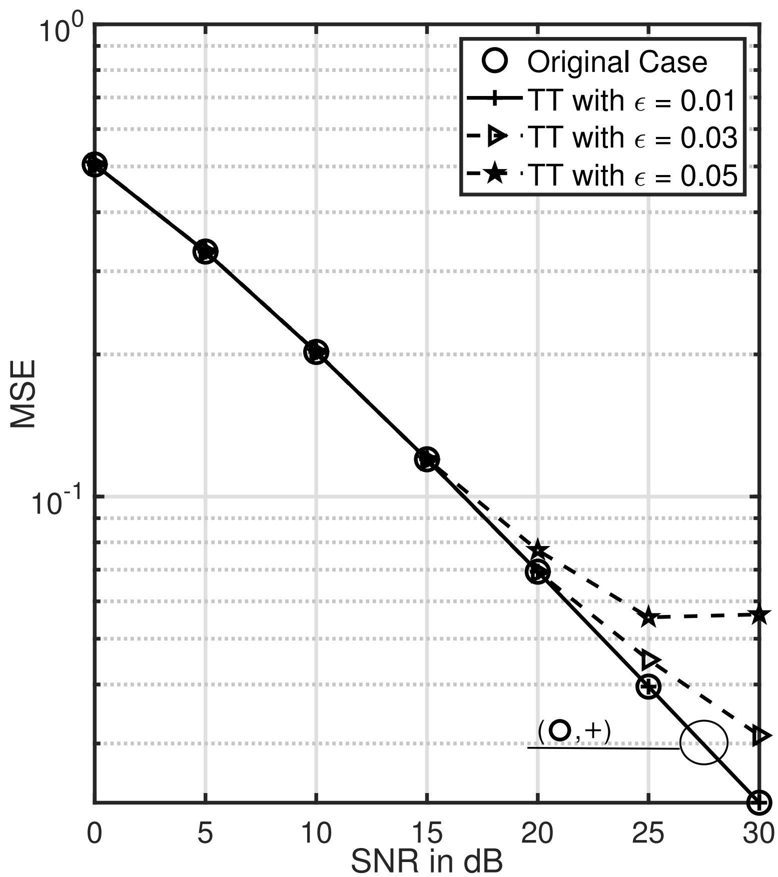


Fig. 4.5: MSE vs SNR for estimation of tensor in TT format.

4.2.4 Tensor Estimation for MIMO OFDM System

Consider the vector based system model for MIMO OFDM from (2.42). A common approach used in MIMO OFDM receiver design is to assume that there is no inter-carrier interference, i.e. $\check{\mathrm{H}}^{(p,q)} = 0$ if $p \neq q$, and perform LMMSE estimation on a per sub-carrier basis [188, 189, 190]. Assuming that the input $\underline{\check{\mathbf{x}}}^{(p)}$ have i.i.d. zero mean unit variance ele-

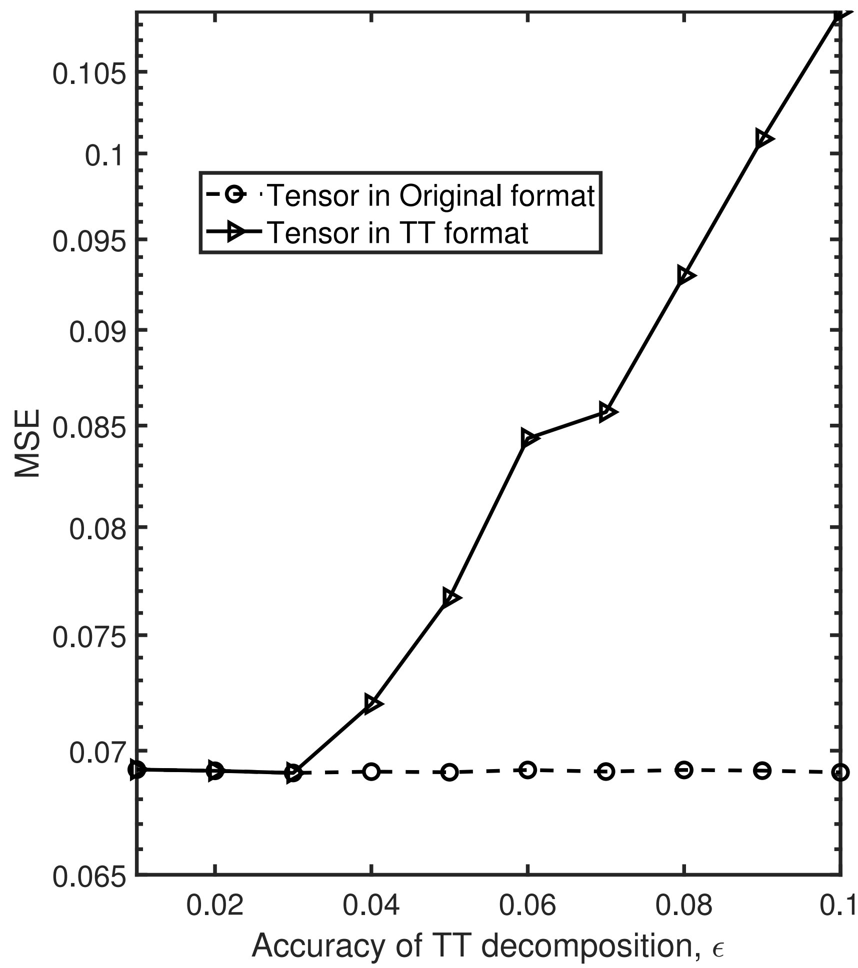


Fig. 4.6: MSE vs accuracy of the tensor TT format.

ments such that the input covariance is an identity matrix, I for each p, and is independent of noise $\check{\underline{\mathbf{n}}}^{(p)}$, the receiver structure with per sub-carrier estimation is given as [188]:

$$\hat{\check{\underline{\mathbf{x}}}}^{(p)} = \check{\mathrm{H}}^{(p,p)H} \cdot (\check{\mathrm{H}}^{(p,p)} \cdot \check{\mathrm{H}}^{(p,p)H} + \mathrm{C}_N)^{-1} \cdot \check{\underline{\mathbf{y}}}^{(p)} \tag{4.82}$$

for $p = 1, \ldots, N_{sc}$, and $\mathrm{C}_N \in \mathbb{C}^{N_R \times N_R}$ is the noise covariance matrix. The per sub-carrier estimation in (4.82) is based on the standard LMMSE filter used for matrix based systems. If we ignore the ICI completely and assume noise to be circularly symmetric white Gaussian with σ_n^2 variance elements, then $\mathrm{C}_N = \sigma_n^2 \cdot \mathrm{I}$. Alternately, one can treat the inter-carrier interference term in (2.42) also as noise, in which case $\mathrm{C}_N = (\sigma_n^2 \cdot \mathrm{I} + \sum_{q \neq p} \check{\mathrm{H}}^{(p,q)} \cdot \check{\mathrm{H}}^{(p,q)H})$. This approach however does not make good use of the ICI terms to extract signal information. In several scenarios, such as in high mobility systems, the channel is doubly selective leading to strong ICI in which case ignoring the interference terms or treating them as noise would lead to sub-optimal performance.

Alternately, consider the tensor based system model for MIMO OFDM from (2.44) where the channel is represented as a fourth order tensor $\check{\mathcal{H}} \in \mathbb{C}^{N_R \times N_{sc} \times N_T \times N_{sc}}$. The ICI is reflected in the elements of $\check{\mathcal{H}}_{n_r,p,n_t,q}$ when $p \neq q$. With such a system model in place, one can use the tensor based receiver structure from (4.17) or (4.38) as the tensor formulation provides an easy method to take into account the information provided by the interfering terms across all the domains.

We present simulation results using TWL and TL estimation for a MIMO OFDM system with $N_{sc} = 64$ sub-carriers, 2 transmit and 2 receive antennas. The channel between each transmit and receive pair of antennas is generated as in [191, 192]. The channel impulse response matrix between n_tth transmit and n_rth receive antenna denoted as $\bar{\mathrm{H}}^{(n_r,n_t)} \in \mathbb{C}^{N_{sc} \times N_{sc}}$ is generated by employing a two tap multipath ($L = 2$) fading channel following Jakes' model [193] using exponential power profile, $\sigma_l^2 = \frac{\exp(-l/L)}{\sum_{l=0}^{L-1} \exp(-l/L)}$ where σ_l^2 represents the variance of the lth channel tap. This matrix is further converted to the frequency domain using the DFT matrix $\mathrm{W} \in \mathbb{C}^{N_{sc} \times N_{sc}}$ with elements $\mathrm{W}_{m,n} = 1/\sqrt{N_{sc}} \exp(-j2\pi mn/N_{sc})$, which then forms the sub-tensor of the frequency domain channel tensor as $\check{\mathcal{H}}_{n_r,:,n_t,:} = \mathrm{W}\bar{\mathrm{H}}^{(n_r,n_t)}\mathrm{W}^H$. The channels are generated for different values of Doppler d, normalized to the OFDM symbol rate, to induce inter-carrier interference. Unless otherwise stated, we take $d = 0.2$. All the results presented were calculated using Monte Carlo simulations with averaging over 500 channel realizations, and at least 100 bit errors were collected for each channel realization to calculate Bit Error Rate (BER). The MSE / BER results are plotted against received SNR per bit.

Figure 4.7 presents the performance of TL and TWL estimators when input is drawn from a Binary Phase Shift Keying (BPSK) constellation which makes the input improper. For the simulation, all the elements in the input tensor are uncorrelated and drawn from a BPSK constellation with unit energy, thereby making the input covariance $\mathcal{C}_{\mathfrak{X}}$ and pseudo-covariance $\tilde{\mathcal{C}}_{\mathfrak{X}}$ as identity tensors. To ensure that the input symbols are uncorrelated,

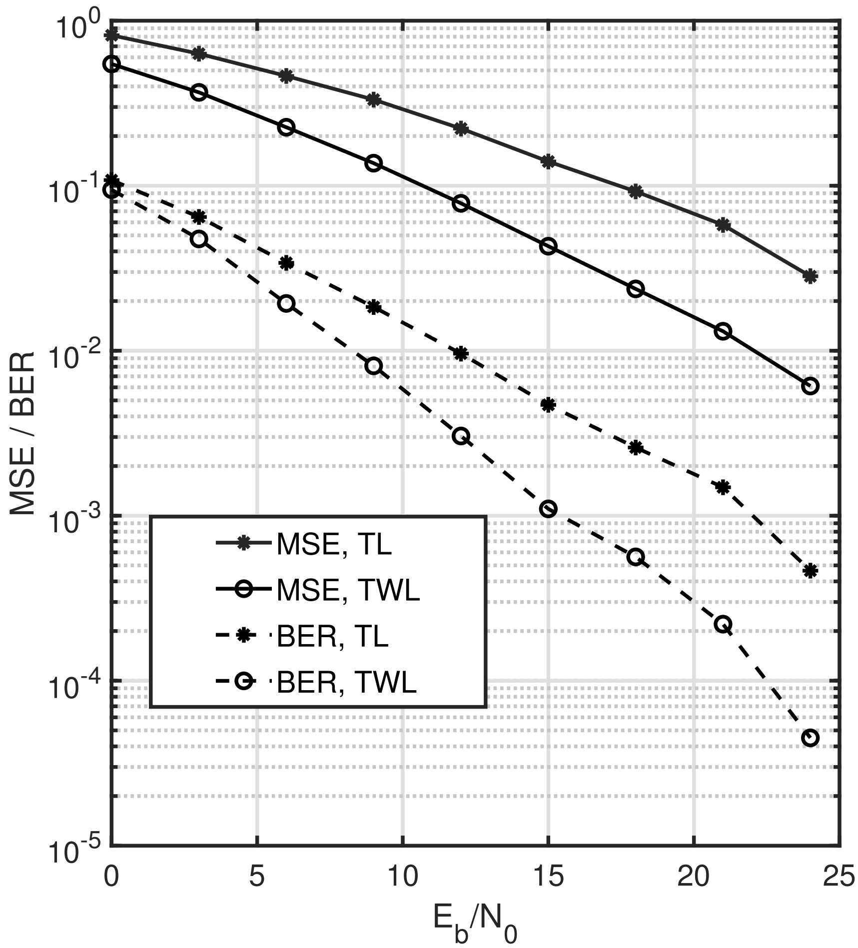


Fig. 4.7: MSE and BER vs E_b/N_0 for 2×2 MIMO OFDM with BPSK input.

one can use symbol interleaving which is commonly used in practice [194]. The received covariance, cross covariance, pseudo-covariance and cross pseudo-covariance are calculated using (4.67)-(4.70) respectively. For TWL estimation, (4.17) is used at the receiver where $\mathcal{A}_1$ and $\mathcal{A}_2$ are given by (4.29) and (4.31). For TL estimation, (4.38) is used at the receiver. The output of the estimator in both cases is passed through a BPSK demodulator to determine the transmitted symbols and calculate the BER. Since BPSK is an improper constellation, we can clearly see in Figure 4.7 that the widely multi-linear MMSE estimator from (4.17) outperforms multi-linear MMSE estimator from (4.38).

Next, we compare estimation in MIMO OFDM with 4-Quadrature Amplitude Modu-

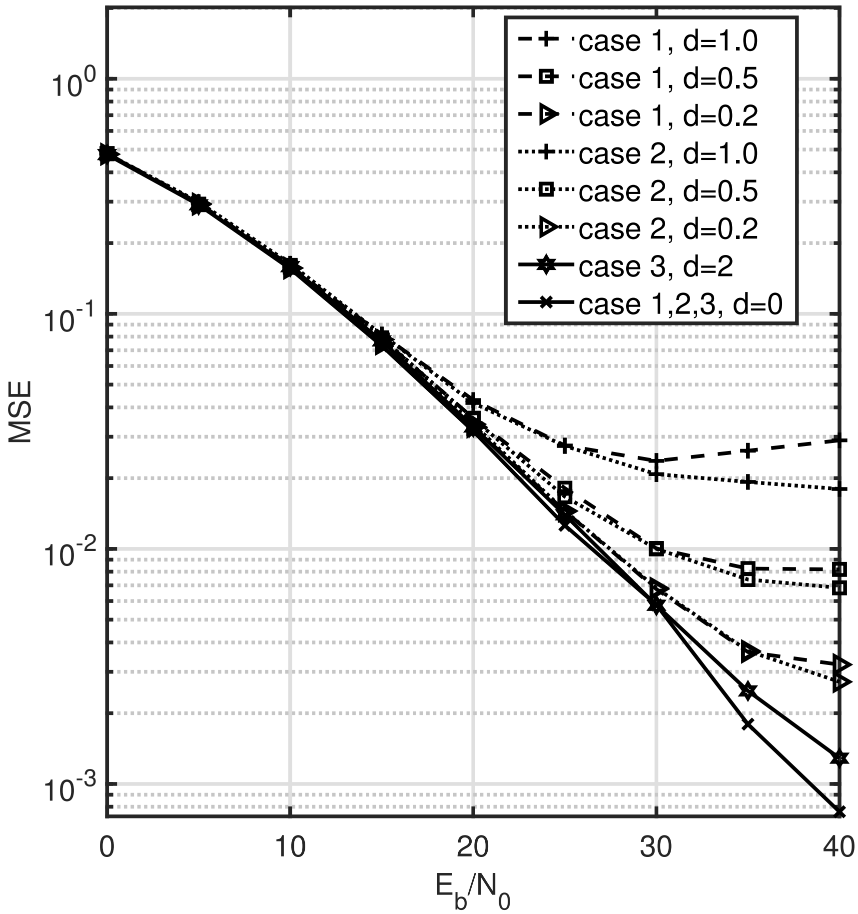


Fig. 4.8: MSE vs E_b/N_0 for 2×2 MIMO OFDM with 4QAM input for different Doppler values d.

lation (QAM) using the tensor multi-linear estimation and the per sub-carrier estimation. We consider three different scenarios based on the estimation technique employed by the receiver :

Case 1 : Per sub-carrier estimation from (4.82) where interference terms are completely ignored, such that $\mathrm{C}_N = \sigma_n^2 \cdot \mathrm{I}$.

Case 2 : Per sub-carrier estimation from (4.82) with interference treated as noise, such that $\mathrm{C}_N = (\sigma_n^2 \cdot \mathrm{I} + \sum_{q \neq p} \check{\mathrm{H}}^{(p,q)} \cdot \check{\mathrm{H}}^{(p,q)H})$.

Case 3 : Multi-linear MMSE estimation from (4.38) with $\mathfrak{C}_{\mathfrak{Y}}$ and $\mathfrak{C}_{\mathfrak{X}\mathfrak{Y}}$ calculated from (4.67) and (4.68) respectively.

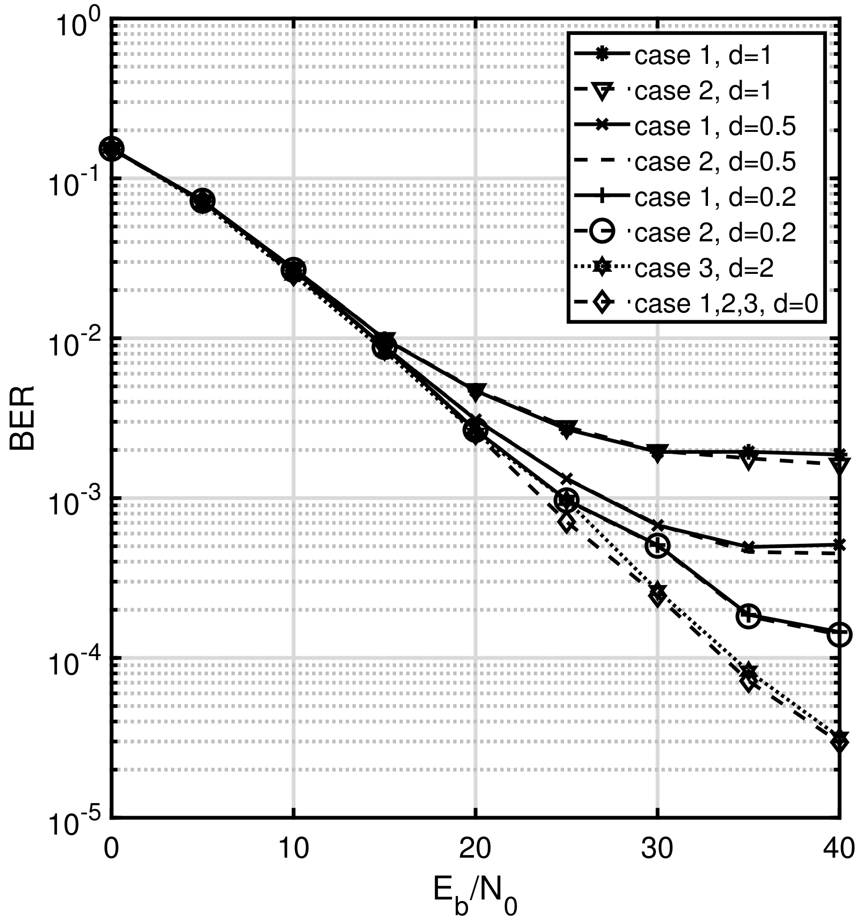


Fig. 4.9: BER vs E_b/N_0 for 2×2 MIMO OFDM with 4QAM input for different Doppler values d.

Figures 4.8 and 4.9 present the MSE and BER for the three cases with different values of the Doppler parameter d. It can be seen that as d increases, there is a significant performance degradation for case 1 and case 2 as compared to case 3. Case 2 performs slightly better than case 1 at higher SNRs as it accounts for interference albeit as noise. It can be observed in Figure 4.8 that for case 1 at $d = 1$, the mean squared error increases at very high SNR. This is because at high SNR, the receiver of case 1 acts like a zero forcing receiver which tries to invert the channel while completely ignoring the interference terms.

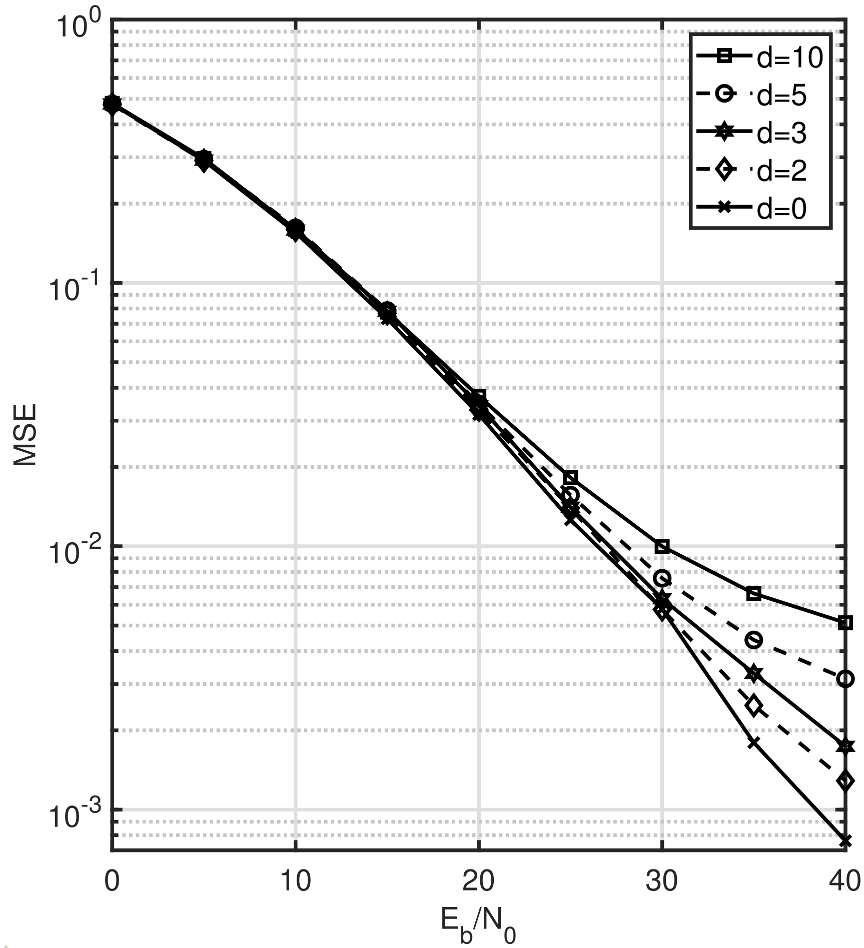


Fig. 4.10: MSE vs E_b/N_0 for 2×2 MIMO OFDM system with 4QAM modulation for higher Doppler values d.

Hence for high values of d when the interfering terms are dominant in the received signal, the channel inversion further amplifies the interference part of the received signal leading to a higher mean squared error. This is easily remedied by making simultaneous use of the information from all the domains with the tensor multi-linear MMSE estimator, as done with case 3. Also, case 3 shows robustness to a change in d, as the MSE or BER performance do not change significantly between $d = 0$ to $d = 2$. The robustness of the multi-linear MMSE estimation can be further observed in Figure 4.10 where the MSE results for case 3 are presented for very high values of Doppler parameter, d. It can be seen that only at extremely large values of Doppler, does the performance degrades. Such large values of the Doppler may not be very practical but are presented here only for a comparison and to demonstrate the behavior of the proposed estimator in extreme cases.

4.3 Chapter Summary

This chapter considered MMSE estimation techniques for tensor based signals. A unified framework for estimation of complex tensors, proper or improper, has been developed using the Einstein product. The multi-linear and widely multi-linear MMSE estimation techniques for multi-domain signals have been formulated while keeping the multi-way structure of the signals intact. The error covariance associated with such estimations has been characterized as a higher order tensor. We compared the proposed estimator using the Einstein product with the Tucker approach. The Tucker product based estimator offers suboptimal performance while providing lower computational complexity. The estimator using Einstein product has a higher computational complexity but provides much better MSE performance. We showed that the proposed estimator can also be used in applications where tensors are stored in TT formats. Also, as an example, we considered MIMO OFDM in a doubly selective channel using the tensor framework. The channel was represented using an order four tensor, accounting for inter-carrier and inter-antenna interference in a single framework. It was shown that the tensor based MMSE estimation outperforms per sub-carrier estimation for MIMO OFDM especially when the inter-carrier interference is high.

Chapter 5

Capacity of Tensor Channels Under Discrete Input Signal Constraints

In view of a tensor-based system model for multi-domain communication systems, we have established in Chapter 3 that the capacity of a tensor channel with additive circularly symmetric complex Gaussian noise under a family of power constraints is achieved when the input is also circularly symmetric complex Gaussian distributed. The optimal input covariance depends on the specific power constraints at the transmitter. For a wide variety of power constraints such as per domain or per element power constraints, the capacity achieving input covariance can be approximated using Algorithm 1. However, in a practical communication system, the input is often drawn from discrete signalling constellations rather than a Gaussian distribution. Thus it is of significant interest to consider the problem of maximizing the mutual information between the input and the output tensors when the input elements are drawn from a given discrete distribution. In this chapter, we investigate how the additional constraints on the input constellations affect the capacity.

For the MIMO matrix channel case, the problem of maximizing mutual information with discrete inputs has been handled using the relationship between the mutual information and minimum mean square error, popularly known as the I-MMSE relation as derived in [104]. The I-MMSE relation links the gradient of the mutual information with the total

MSE obtained from an estimator based on the conditional expectation. The I-MMSE relationship holds for any input distribution so far as the input-output pair are linked via additive circularly symmetric Gaussian noise. Since closed form expressions for mutual information are not easy to derive when the input is drawn from discrete constellations, the I-MMSE relation comes to play as it provides a mechanism to bypass the need of finding the exact mutual information expression. Using the principles employed in I-MMSE derivation from [104], the gradient of mutual information with respect to a variety of system parameters such as the channel matrix and the precoder, has been derived in [107] for vector Gaussian channels. The problem of power allocation in parallel non-interfering Gaussian channels is solved [105] using the I-MMSE relation leading to a mercury water-filling solution. Furthermore, for a general MIMO channel with interfering terms, [106] uses the relation between the gradient of the mutual information with respect to a linear precoding matrix derived in [107] to obtain an optimal linear precoder.

So far, the problem of finding a precoder or input covariance that maximizes the mutual information with discrete inputs has only been addressed in the context of scalar or vector inputs, where only a single domain (space/antenna) is taken into account. More so, within the space domain, most of the work in literature assumes only a sum power constraint and not per antenna power constraints. In this chapter, we consider the problem of finding a multi-linear input precoder which maximizes the input-output mutual information in a tensor based multi-domain communication system. Furthermore, we consider not just the sum power constraint but a family of power constraints as defined in section 3.2.1 which includes the per domain or per element power constraints. In this chapter, we combine the results from Chapter 3 and 4 to handle the case of discrete inputs while maximizing the mutual information. We first derive the gradient of mutual information with respect to the channel tensor, and exploit its relation with the MMSE tensor for our purpose.

5.1 The Tensor I-MMSE relation

In this section we will derive the gradient of the mutual information with respect to several parameters such as the channel and the precoder, and express them using the MMSE tensor.

Consider the system model from (2.35) where an order-N input $\mathcal{X} \in \mathbb{C}^{I_1 \times \ldots \times I_N}$ and order-M output $\mathcal{Y} \in \mathbb{C}^{J_1 \times \ldots \times J_M}$ are connected via a multi-linear channel as :

$$\mathcal{Y} = \mathcal{H} *_N \mathcal{X} + \mathcal{N} \tag{5.1}$$

with $\mathcal{N}$ representing the received noise tensor of same size as $\mathcal{Y}$. We assume that $\mathcal{N}$ is circularly symmetric complex Gaussian distributed having independent zero mean and unit variance components such that its covariance is an identity tensor.

The best MMSE estimator, given by the conditional mean, has been derived for the tensor setting in Chapter 4 along with the associated error covariance tensor. We denote the order-$2N$ error covariance tensor using $\mathcal{Q}_{\mathcal{E}} \in \mathbb{C}^{I_1 \times \ldots \times I_N \times I_1 \times \ldots \times I_N}$, which can be written as (based on (4.13)) :

$$\begin{aligned}
\mathcal{Q}_{\mathcal{E}} &= \mathbb{E}[(\mathcal{X} - \mathbb{E}[\mathcal{X}|\mathcal{Y}]) \circ (\mathcal{X} - \mathbb{E}[\mathcal{X}|\mathcal{Y}])^*] && (5.2)\\
&= \mathbb{E}[\mathcal{X} \circ \mathcal{X}^*] + \mathbb{E}\Big[\mathbb{E}[\mathcal{X}|\mathcal{Y}] \circ \mathbb{E}[\mathcal{X}^*|\mathcal{Y}]\Big] - \mathbb{E}\Big[\mathbb{E}[\mathcal{X}|\mathcal{Y}] \circ \mathcal{X}^*\Big] - \mathbb{E}\Big[\mathcal{X} \circ \mathbb{E}[\mathcal{X}^*|\mathcal{Y}]\Big] && (5.3)\\
&= \mathbb{E}\Big[\mathcal{X} \circ \mathcal{X}^*\Big] + \mathbb{E}\Big[\mathbb{E}[\mathcal{X}|\mathcal{Y}] \circ \mathbb{E}[\mathcal{X}^*|\mathcal{Y}]\Big] - \mathbb{E}\Big[\mathbb{E}[\mathcal{X}|\mathcal{Y}] \circ \mathbb{E}[\mathcal{X}^*|\mathcal{Y}]\Big] - \mathbb{E}\Big[\mathbb{E}[\mathcal{X}|\mathcal{Y}] \circ \mathbb{E}[\mathcal{X}^*|\mathcal{Y}]\Big] && (5.4)\\
&= \mathbb{E}\Big[\mathcal{X} \circ \mathcal{X}^*\Big] - \mathbb{E}\Big[\mathbb{E}[\mathcal{X}|\mathcal{Y}] \circ \mathbb{E}[\mathcal{X}^*|\mathcal{Y}]\Big]. && (5.5)
\end{aligned}$$

The MMSE tensor $\mathcal{Q}_{\mathcal{E}}$ associated with the conditional mean estimator can be linked with the gradient of the input-output mutual information as shown in the following theorem :

Theorem 5. *In the system model* (5.1)*, if $\mathcal{N}$ is circularly symmetric complex Gaussian with zero mean and identity covariance tensor, then the gradient of the input-output mutual information with respect to the channel tensor is given by*

$$\nabla_{\mathcal{H}} \mathcal{I}(\mathcal{X}; \mathcal{Y}) = \mathcal{H} *_N \mathcal{Q}_{\mathcal{E}} \tag{5.6}$$

where $\mathcal{Q}_{\mathcal{E}}$ is the MMSE tensor defined in (5.5)*.*

Proof. The proof of this theorem when the input and output are vectors, thus the channel is a matrix, is provided in [107]. The same proof can be extended to the tensor case as shown in Appendix B.9. □

Now consider a more general system model:

$$\boldsymbol{\mathcal{Y}} = \mathcal{H} *_N \mathcal{P} *_N \boldsymbol{\mathcal{X}} + \boldsymbol{\mathcal{N}} \tag{5.7}$$

where $\mathcal{P} \in \mathbb{C}^{I_1 \times \ldots \times I_N \times I_1 \times \ldots \times I_N}$ denotes a multi-linear precoding tensor. If we assume input tensor $\boldsymbol{\mathcal{X}}$ is zero mean with covariance $\bar{\mathcal{Q}} = \mathbb{E}[\boldsymbol{\mathcal{X}} \circ \boldsymbol{\mathcal{X}}^*]$, then the transmit covariance would be given by

$$\mathcal{Q} = \mathbb{E}[(\mathcal{P} *_N \boldsymbol{\mathcal{X}}) \circ (\mathcal{P} *_N \boldsymbol{\mathcal{X}})^*] \tag{5.8}$$

$$= \mathbb{E}[\mathcal{P} *_N \boldsymbol{\mathcal{X}} \circ \boldsymbol{\mathcal{X}}^* *_N \mathcal{P}^H] \tag{5.9}$$

$$= \mathcal{P} *_N \bar{\mathcal{Q}} *_N \mathcal{P}^H. \tag{5.10}$$

If the input before any precoding consists of i.i.d. zero mean unit variance elements such that $\bar{\mathcal{Q}} = \mathcal{I}_N$, then the transmit covariance is given as $\mathcal{Q} = \mathcal{P} *_N \mathcal{P}^H$. A direct corollary of Theorem 5 which can be established using the chain rule of differentiation is as follows :

Corollary 5.1. For the system model in (5.7) where $\mathcal{P} \in \mathbb{C}^{I_1 \times \ldots \times I_N \times \ldots \times I_N}$ denotes a multi-linear precoder operator at the input, the noise $\boldsymbol{\mathcal{N}}$ is circularly symmetric complex Gaussian with zero mean and covariance given by identity tensor, and the input $\boldsymbol{\mathcal{X}}$ is arbitrary distributed with covariance $\bar{\mathcal{Q}}$ such that the transmit covariance is given by $\mathcal{Q} = \mathcal{P} *_N \bar{\mathcal{Q}} *_N \mathcal{P}^H$, we have

$$\nabla_{\mathcal{P}} \mathcal{I}(\boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{Y}}) = \mathcal{H}^H *_M \mathcal{H} *_N \mathcal{P} *_N \mathcal{Q}_{\boldsymbol{\mathcal{E}}} \tag{5.11}$$

and

$$\nabla_{\mathcal{Q}} \mathcal{I}(\boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{Y}}) *_N \mathcal{P} *_N \bar{\mathcal{Q}} = \mathcal{H}^H *_M \mathcal{H} *_N \mathcal{P} *_N \mathcal{Q}_{\boldsymbol{\mathcal{E}}} \tag{5.12}$$

where $\mathcal{Q}_{\boldsymbol{\mathcal{E}}}$ is the MMSE tensor defined in (5.5).

Proof. Let $\bar{\mathcal{H}} = \mathcal{H} *_N \mathcal{P}$, then we can write $\boldsymbol{\mathcal{Y}} = \bar{\mathcal{H}} *_N \boldsymbol{\mathcal{X}} + \boldsymbol{\mathcal{N}}$. Thus, from Theorem 5 we know :

$$\nabla_{\bar{\mathcal{H}}} \mathcal{I}(\boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{Y}}) = \bar{\mathcal{H}} *_N \mathcal{Q}_{\boldsymbol{\mathcal{E}}} = \mathcal{H} *_N \mathcal{P} *_N \mathcal{Q}_{\boldsymbol{\mathcal{E}}}. \tag{5.13}$$

Using chain rule of differentiation (see Lemma 12 in Appendix B.8) we have :

$$\nabla_{\mathcal{P}} \mathcal{I}(\boldsymbol{\mathcal{X}};\boldsymbol{\mathcal{Y}}) = \mathcal{H}^H *_M \nabla_{\bar{\mathcal{H}}} \mathcal{I}(\boldsymbol{\mathcal{X}};\boldsymbol{\mathcal{Y}}) = \mathcal{H}^H *_M \mathcal{H} *_N \mathcal{P} *_N \mathcal{Q}_{\boldsymbol{\mathcal{E}}} \tag{5.14}$$

which proves (5.11). Also since $\mathcal{Q} = \mathcal{P} *_N \bar{\mathcal{Q}} *_N \mathcal{P}^H$, we can apply the chain rule (see Lemma 13 in Appendix B.8) to get :

$$\nabla_{\mathcal{P}} \mathcal{I}(\boldsymbol{\mathcal{X}};\boldsymbol{\mathcal{Y}}) = \nabla_{\mathcal{Q}} \mathcal{I}(\boldsymbol{\mathcal{X}};\boldsymbol{\mathcal{Y}}) *_N \mathcal{P} *_N \bar{\mathcal{Q}}. \tag{5.15}$$

Substituting (5.14) into (5.15), we get :

$$\nabla_{\mathcal{Q}} \mathcal{I}(\boldsymbol{\mathcal{X}};\boldsymbol{\mathcal{Y}}) *_N \mathcal{P} *_N \bar{\mathcal{Q}} = \mathcal{H}^H *_M \mathcal{H} *_N \mathcal{P} *_N \mathcal{Q}_{\boldsymbol{\mathcal{E}}} \tag{5.16}$$

which proves (5.12). □

The results of Corollary 5.1 give us the gradient of the mutual information with respect to the input precoder and the transmit covariance in terms of the MMSE tensor. So even in cases where we do not have a closed form expression of the mutual information, these gradient results can be exploited to optimize the mutual information as shown in the next section.

5.2 Maximizing the Mutual Information for arbitrary inputs

Consider the system model from (5.1). Assuming that the input elements are drawn from a zero mean signal constellation with a fixed probability distribution, the objective is to determine the transmit covariance $\mathcal{Q}$ that maximizes the mutual information $\mathcal{I}(\boldsymbol{\mathcal{X}};\boldsymbol{\mathcal{Y}})$ subject to a given power constraint. Hence we can define the optimization problem as:

$$\max_{\mathcal{Q}} \quad \mathcal{I}(\boldsymbol{\mathcal{X}};\boldsymbol{\mathcal{Y}}) \tag{5.17}$$

$$s.t. \quad \sum_{\underline{i}_r} \mathcal{Q}_{\underline{i},\underline{i}} \leq P_{\underline{i}_c} \quad \forall \underline{i}_c, \tag{5.18}$$

$$\mathcal{Q} \succeq 0. \tag{5.19}$$

For an explanation of the notations describing the power constraints, refer to section 3.2.1. Unlike the case of Gaussian input, the optimization problem in (5.17)-(5.19) is not necessarily a concave function maximization since the mutual information between the input and the output for any given input constellation may not yield a concave objective function. More so, closed form expression of the mutual information for any given constellation may not be known. However, the KKT conditions provide a set of necessary conditions for the optimal input covariance to be a critical point to any optimization problem [195, 150]. Hence we will now derive the necessary conditions for the optimal covariance using the KKT conditions.

5.2.1 Conditions for Optimal Covariance

Let $\mathcal{M} \succeq 0$ be the Lagrange multiplier tensor for the positive semi-definite constraint from (5.19) of size $I_1 \times \dots I_N \times I_1 \times \dots I_N$. Let $\mu_{\underline{i}_c} \geq 0$ for all $\underline{i}_c$ be the Lagrange multipliers corresponding to all the linear constraints from (5.18). Then the Lagrangian functional can be defined as :

$$\mathcal{L}(\mathcal{Q}, \{\mu_{\underline{i}_c}\}, \mathcal{M}) = -\mathcal{I}(\boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{Y}}) + \sum_{\underline{i}_c} \mu_{\underline{i}_c} (\sum_{\underline{i}_r} \mathcal{Q}_{\underline{i},\underline{i}} - P_{\underline{i}_c}) - \operatorname{tr}(\mathcal{M} *_N \mathcal{Q}). \tag{5.20}$$

Consider a pseudo-diagonal tensor $\mathcal{B}$ of same size as the input covariance, such that its non-zero elements are $\mathcal{B}_{\underline{i},\underline{i}} = \mu_{\underline{i}_c}, \forall \underline{i}_r$. For instance, if $\underline{i}_c = (i_1, i_2)$, then $\mathcal{B}_{i_1,\dots,i_N,i_1,\dots,i_N} = \mu_{i_1,i_2}$ for all $(i_3, \dots, i_N)$. Then we get

$$\sum_{\underline{i}_c} \mu_{\underline{i}_c} \cdot \sum_{\underline{i}_r} \mathcal{Q}_{\underline{i},\underline{i}} = \sum_{\underline{i}_c} \sum_{\underline{i}_r} \mu_{\underline{i}_c} \cdot \mathcal{Q}_{\underline{i},\underline{i}} = \sum_{\underline{i}} \mathcal{B}_{\underline{i},\underline{i}} \cdot \mathcal{Q}_{\underline{i},\underline{i}} = \operatorname{tr}(\mathcal{B} *_N \mathcal{Q}). \tag{5.21}$$

Based on (5.21), we can re-write the Lagrangian from (5.20) as :

$$\mathcal{L}(\mathcal{Q}, \{\mu_{\underline{i}_c}\}, \mathcal{M}) = -\mathcal{I}(\boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{Y}}) - \sum_{\underline{i}_c} \mu_{\underline{i}_c} P_{\underline{i}_c} + \operatorname{tr}(\mathcal{B} *_N \mathcal{Q}) - \operatorname{tr}(\mathcal{M} *_N \mathcal{Q}) \tag{5.22}$$

and the derivative of Lagrangian with respect to $\mathcal{Q}$ can be written as :

$$\nabla_{\mathcal{Q}} \mathcal{L} = -\nabla_{\mathcal{Q}} \mathcal{I}(\boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{Y}}) + \mathcal{B} - \mathcal{M} \tag{5.23}$$

The first KKT condition is obtained by setting the $\nabla_{\mathcal{Q}} \mathcal{L}$ from (5.23) to zero as :

$$-\nabla_{\mathcal{Q}} \mathcal{I}(\boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{Y}}) + \mathcal{B} - \mathcal{M} = 0_{\mathcal{J}}. \tag{5.24}$$

The KKT conditions also include complementary slackness equations corresponding to each constraint. These equations are obtained by setting every function in the Lagrangian functional from (5.20) to 0 except the objective function. A detailed explanation has been provided in Appendix A.1. Thus we can write the complementary slackness equations as

$$\operatorname{tr}(\mathcal{M} *_N \mathcal{Q}) = 0, \tag{5.25}$$

$$\mu_{\underline{i}_c}\Big(\sum_{\underline{i}_r} \mathcal{Q}_{\underline{i},\underline{i}} - P_{\underline{i}_c}\Big) = 0, \quad \forall \underline{i}_c. \tag{5.26}$$

Since $\mathcal{M} \succeq 0$ and $\mathcal{Q} \succeq 0$, using Lemma 9 from Appendix A.1, we can write (5.25) as :

$$\mathcal{M} *_N \mathcal{Q} = 0_{\mathcal{T}}. \tag{5.27}$$

Also notice that all the entries of $\mathcal{B}$, i.e $\mu_{\underline{i}_c}$, will be strictly greater than 0 at optimum because the inequality constraint must be met with equality at optimum. Since $\mu_{\underline{i}_c} > 0$, (5.26) can be written as as :

$$\sum_{\underline{i}_r} \mathcal{Q}_{\underline{i},\underline{i}} - P_{\underline{i}_c} = 0, \quad \forall \underline{i}_c. \tag{5.28}$$

Now consider the relation between the gradient of the mutual information and the MMSE tensor from Corollary 5.1 (assuming $\mathcal{P} = \mathcal{I}_N$, i.e. an identity tensor, and thus $\mathcal{Q} = \bar{\mathcal{Q}}$ in (5.12)), we can write

$$\nabla_{\mathcal{Q}} \mathcal{I}(\boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{Y}}) *_N \mathcal{Q} = \mathcal{H}^H *_M \mathcal{H} *_N \mathcal{Q}_{\boldsymbol{\mathcal{E}}}. \tag{5.29}$$

We can rewrite (5.24) as :

$$\nabla_{\mathcal{Q}} \mathcal{I}(\boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{Y}}) = \mathcal{B} - \mathcal{M} \tag{5.30}$$

$$\Rightarrow \nabla_{\mathcal{Q}} \mathcal{I}(\boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{Y}}) *_N \mathcal{Q} = \mathcal{B} *_N \mathcal{Q} - \mathcal{M} *_N \mathcal{Q}. \tag{5.31}$$

On equating (5.29) and (5.31) , we get :

$$\mathcal{H}^H *_M \mathcal{H} *_N \mathcal{Q}_{\boldsymbol{\mathcal{E}}} = \mathcal{B} *_N \mathcal{Q} - \underbrace{\mathcal{M} *_N \mathcal{Q}}_{=0_{\mathcal{T}},\ \text{from}(5.27)}. \tag{5.32}$$

Thus we have

$$\mathcal{H}^H *_M \mathcal{H} *_N \mathcal{Q}_{\boldsymbol{\mathcal{E}}} = \mathcal{B} *_N \mathcal{Q} \tag{5.33}$$

$$\Rightarrow \mathcal{Q} = \mathcal{B}^{-1} *_N \mathcal{H}^H *_M \mathcal{H} *_N \mathcal{Q}_{\boldsymbol{\mathcal{E}}} \tag{5.34}$$

where the elements of tensor $\mathcal{B}$ are found to satisfy the power constraints in (5.28). The relation derived in (5.34) when reduced to a matrix channel with sum power constraints (for which $\mathcal{B}$ is a scaled identity matrix) at the input reduces to the result of Theorem 1 from [109]. Thus (5.34) generalizes the optimum covariance result for MIMO matrix channel under sum power constraint, to a higher order tensor channel model under a family of power constraints.

5.2.2 Case of Gaussian signalling

If the input $\boldsymbol{\mathfrak{X}}$ is taken to be circularly symmetric complex Gaussian distributed with covariance $\mathcal{Q}$, then based on (3.13), the mutual information is given by $\log[\det(\mathcal{H} *_N \mathcal{Q} *_N \mathcal{H}^H + \mathcal{I}_M)]$. Hence the gradient of the mutual information for the Gaussian input case with respect to $\mathcal{Q}$ is given as (based on (2.27)) :

$$\nabla_{\mathcal{Q}} \mathcal{I}(\boldsymbol{\mathfrak{X}};\boldsymbol{\mathfrak{Y}}) = \mathcal{H}^H *_M (\mathcal{I}_M + \mathcal{H} *_N \mathcal{Q} *_N \mathcal{H}^H)^{-1} *_M \mathcal{H}. \tag{5.35}$$

Also, from Chapter 4 we know that when the input and output are jointly circular complex Gaussian, the best estimator (conditional mean estimator) is same as the multi-linear MMSE estimator. The MMSE tensor corresponding to the multi-linear MMSE estimation is given as (based on (4.40)) :

$$\mathcal{Q}_{\boldsymbol{\mathcal{E}}} = \mathcal{Q} - \mathcal{Q} *_N \mathcal{H}^H *_M (\mathcal{I}_M + \mathcal{H} *_N \mathcal{Q} *_N \mathcal{H}^H)^{-1} *_M \mathcal{H} *_N \mathcal{Q}. \tag{5.36}$$

From (5.36) we can write :

$$\mathcal{Q}_{\boldsymbol{\mathcal{E}}} *_N \mathcal{H}^H = \mathcal{Q} *_N \mathcal{H}^H - \mathcal{Q} *_N \mathcal{H}^H *_M (\mathcal{I}_M + \underbrace{\mathcal{H} *_N \mathcal{Q} *_N \mathcal{H}^H}_{\mathcal{A}})^{-1} *_M \underbrace{\mathcal{H} *_N \mathcal{Q} *_N \mathcal{H}^H}_{\mathcal{A}}. \tag{5.37}$$

Based on the property, $(\mathcal{I}_M + \mathcal{A})^{-1} *_M \mathcal{A} = (\mathcal{I}_M + \mathcal{A})^{-1} *_M (\mathcal{I}_M + \mathcal{A} - \mathcal{I}_M) = \mathcal{I}_M - (\mathcal{I}_M + \mathcal{A})^{-1}$, we have :

$$\mathcal{Q}_{\boldsymbol{\mathcal{E}}} *_N \mathcal{H}^H = \mathcal{Q} *_N \mathcal{H}^H - \mathcal{Q} *_N \mathcal{H}^H *_M \Big(\mathcal{I}_M - (\mathcal{I}_M + \mathcal{H} *_N \mathcal{Q} *_N \mathcal{H}^H)^{-1}\Big) \tag{5.38}$$

$$= \cancel{\mathcal{Q} *_N \mathcal{H}^H} - \cancel{\mathcal{Q} *_N \mathcal{H}^H} + \mathcal{Q} *_N \mathcal{H}^H *_M (\mathcal{I}_M + \mathcal{H} *_N \mathcal{Q} *_N \mathcal{H}^H)^{-1} \tag{5.39}$$

$$\Rightarrow \mathcal{Q}_{\boldsymbol{\mathcal{E}}} *_N \mathcal{H}^H *_M \mathcal{H} = \mathcal{Q} *_N \mathcal{H}^H *_M (\mathcal{I}_M + \mathcal{H} *_N \mathcal{Q} *_N \mathcal{H}^H)^{-1} *_M \mathcal{H}. \tag{5.40}$$

Taking Hermitian on both sides and noting that $\mathcal{Q}_{\boldsymbol{\mathcal{E}}}$ and $(\mathcal{I}_M + \mathcal{H} *_N \mathcal{Q} *_N \mathcal{H}^H)^{-1}$ are both Hermitian tensors, we get :

$$\mathcal{H}^H *_M \mathcal{H} *_N \mathcal{Q}_{\boldsymbol{\mathcal{E}}} = \mathcal{H}^H *_M (\mathcal{I}_M + \mathcal{H} *_N \mathcal{Q} *_N \mathcal{H}^H)^{-1} *_M \mathcal{H} *_N \mathcal{Q} \tag{5.41}$$

$$= \nabla_{\mathcal{Q}} \mathcal{I}(\boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{Y}}) *_N \mathcal{Q}, \quad \text{(using (5.35))}. \tag{5.42}$$

which is the same as (5.29). This validates the relationship between the gradient of mutual information with respect to the input covariance and the MMSE tensor for the specific case of Gaussian signalling in which case we have closed form expressions for the mutual information and the MMSE tensor.

5.3 Solving for the Optimal Input Precoder

It is to be noted that while (5.34) states an equation which must be satisfied by the optimal input covariance tensor, it does not uniquely identifies the same. Since the MMSE tensor $\mathcal{Q}_{\boldsymbol{\mathcal{E}}}$ in (5.34) would itself depend on the transmit covariance tensor, thus (5.34) does not provide a direct equation to solve for the input covariance. In order to solve for the optimal transmit covariance, we may adopt an iterative approach using a gradient ascent method. Consider the system model from (5.7), with input $\boldsymbol{\mathcal{X}}$ containing i.i.d. zero mean unit variance entries. Thus the tensor $\mathcal{P}$ can be seen as a multi-linear precoder tensor which also does power allocation given the specific power constraints. Given the precoder tensor, the transmit covariance $\mathcal{Q}$ can be seen as $\mathcal{Q} = \mathcal{P} *_N \mathcal{P}^H$ (based on (5.10)). Thus the optimization problem in (5.17) can be recast using the multi-linear precoder $\mathcal{P}$ as the optimizing variable. Our objective is to find a $\mathcal{P}$ that maximizes the mutual information $\mathcal{I}(\boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{Y}})$ for a given input distribution subject to the power constraints, and can be described in terms of $\mathcal{P}$ as :

$$\max_{\mathcal{P}} \;\; \mathcal{I}(\boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{Y}}) \tag{5.43}$$

$$s.t. \;\; \sum_{\underline{i}_r} (\mathcal{P} *_N \mathcal{P}^H)_{\underline{i},\underline{i}} \leq P_{\underline{i}_c} \quad \forall \underline{i}_c. \tag{5.44}$$

Let $\mathbb{P}$ denote the set of all such tensors which satisfy the constraints in (5.44), known as the feasible set. We use a gradient ascent approach to find the optimal $\mathcal{P}$ using the following iterative equation :

$$\mathcal{P}^{(k+1)} = \left[\mathcal{P}^{(k)} + \nu \nabla_{\mathcal{P}} \mathcal{I}^{(k)}\right]_{\mathbb{P}} \tag{5.45}$$

where $[\cdot]_{\mathbb{P}}$ denotes the projection onto the feasible set $\mathbb{P}$ such that that power constraints are satisfied, i.e. $\sum_{\underline{i}_r} (\mathcal{P}^{(k+1)} *_N \mathcal{P}^{(k+1)H})_{\underline{i},\underline{i}} = P_{\underline{i}_c}, \forall \underline{i}_c$. The projection of any $\mathcal{P}^{(0)}$ onto the feasible set $\mathbb{P}$ can be mathematically defined as $[\mathcal{P}^{(0)}]_{\mathbb{P}} = \arg \min_{\mathcal{P} \in \mathbb{P}} ||\mathcal{P} - \mathcal{P}^{(0)}||^2$. The variable k denotes the iteration index. The constant ν denotes the step size taken for the gradient ascent. Substituting $\nabla_{\mathcal{P}} \mathcal{I}$ from (5.11) into (5.45) we get

$$\mathcal{P}^{(k+1)} = \left[\mathcal{P}^{(k)} + \nu \cdot \mathcal{H}^H *_M \mathcal{H} *_N \mathcal{P}^{(k)} *_N \mathcal{Q}_{\boldsymbol{\mathcal{E}}}^{(k)}\right]_{\mathbb{P}}. \tag{5.46}$$

Equation (5.46) gives an iterative procedure to find the precoder. For low SNRs, any initial guess of $\mathcal{P}^{(0)}$ which satisfies the power constraints with equality can be made. For instance, in general the initial guess for $\mathcal{P}^{(0)}$ could be a pseudo-diagonal tensor corresponding to a uniform power allocation at the input in case of sum power constraints. Note that since the mutual information may not be a concave function for several discrete input distributions, the iterative method from (5.46) would in general provide only a necessary condition for the optimal input precoder. One approach in such cases is to consider running the gradient ascent iterative equation using several starting guesses of $\mathcal{P}^{(0)}$, and then select the stationary point precoder which offers the maximum mutual information as used in [106]. However, an alternate approach could be devised using the fact that the mutual information for discrete inputs is a concave function at low enough SNRs [109]. Thus in order to obtain an optimal $\mathcal{P}$ for any given SNR, we use a similar approach as presented in [109]. First we determine the unique globally optimal input precoder $\mathcal{P}$ using (5.46) for a low enough SNR with any starting guess. The low SNR ensures that the optimization problem is concave hence the solution is indeed globally optimal. Further, we increase the SNR gradually where the optimal precoder for the higher SNR is computed by taking the precoder corresponding to the lower SNR as the starting point. It is assumed that the mutual information as a function of SNR does not change drastically with a small change in SNR. Thus, choosing the globally optimum value at a low SNR as a starting point for a slightly higher SNR ensures that the starting point remains in the proximity of the global optimum for the higher SNR case. Such a technique is sometimes used for non concave function optimization [196, 109].

Further, to implement the iterative equation from (5.46) to find the optimal $\mathcal{P}$, we need to address the following questions : how to compute the projection operation $[\cdot]_{\mathbb{P}}$, how to compute the MMSE tensor, and how to evaluate the achievable mutual information for performance comparison in the absence of any closed form expressions. We will now discuss all these steps in detail.

5.3.1 Determining the projection onto the feasible set

The projection of the iterative precoder onto the set of feasible precoders satisfying the sum power constraints for MIMO channels has been cast as a convex optimization problem in [107]. Similarly, the projection of a tensor $\mathcal{P}^{(0)}$ onto the feasible set $\mathbb{P}$ satisfying the family of power constraints denoted by $[\cdot]_{\mathbb{P}}$, can be cast as a convex optimization problem :

$$\min_{\mathcal{P}} \quad ||\mathcal{P} - \mathcal{P}^{(0)}||^2 \tag{5.47}$$

$$\text{s.t.} \ \sum_{\underline{i}_r} (\mathcal{P} *_N \mathcal{P}^H)_{\underline{i},\underline{i}} = P_{\underline{i}_c}, \forall \underline{i}_c. \tag{5.48}$$

Such a projection problem can be solved using the KKT conditions [150], or using software tools such as CVX [152]. Here we present the solution using the KKT conditions.

Let $l_{\underline{i}_c} \geq 0$ for all $\underline{i}_c$ be the Lagrange multipliers corresponding to all the linear constraints from (5.48). Then the Lagrangian functional corresponding to (5.47)-(5.48) can be defined as :

$$\mathcal{L}(\mathcal{P}, \{l_{\underline{i}_c}\}) = ||\mathcal{P} - \mathcal{P}^{(0)}||^2 + \sum_{\underline{i}_c} l_{\underline{i}_c} (\sum_{\underline{i}_r} (\mathcal{P} *_N \mathcal{P}^H)_{\underline{i},\underline{i}} - P_{\underline{i}_c}). \tag{5.49}$$

We arrange the values $\{l_{\underline{i}_c}\}$ in a pseudo-diagonal tensor $\mathcal{F}$ of same size as the input precoder such that its non-zero entries are $\mathcal{F}_{\underline{i},\underline{i}} = l_{\underline{i}_c}, \forall \underline{i}_r$. For instance, if $\underline{i}_c = (i_1, i_2)$, then

$\mathcal{F}_{i_1,\dots,i_N,i_1,\dots,i_N} = l_{i_1,i_2}$ for all $(i_3,\dots,i_N)$. Then we get

$$\sum_{\underline{i}_c} l_{\underline{i}_c} \cdot \sum_{\underline{i}_r} (\mathcal{P} *_N \mathcal{P}^H)_{\underline{i},\underline{i}} = \sum_{\underline{i}_c} \sum_{\underline{i}_r} l_{\underline{i}_c} \cdot (\mathcal{P} *_N \mathcal{P}^H)_{\underline{i},\underline{i}} \tag{5.50}$$

$$= \sum_{\underline{i}} \mathcal{F}_{\underline{i},\underline{i}} \cdot (\mathcal{P} *_N \mathcal{P}^H)_{\underline{i},\underline{i}} \tag{5.51}$$

$$= \operatorname{tr}(\mathcal{F} *_N \mathcal{P} *_N \mathcal{P}^H). \tag{5.52}$$

Based on (5.52), we can re-write the Lagrangian from (5.49) as :

$$\mathcal{L}(\mathcal{P}, \{l_{\underline{i}_c}\}) = ||\mathcal{P} - \mathcal{P}^{(0)}||^2 - \sum_{\underline{i}_c} l_{\underline{i}_c} P_{\underline{i}_c} + \operatorname{tr}(\mathcal{F} *_N \mathcal{P} *_N \mathcal{P}^H). \tag{5.53}$$

To find the derivative of the Lagrangian with respect to $\mathcal{P}$, note that

$$(\mathcal{P} *_N \mathcal{P}^H)_{i_1,\dots,i_N,j_1,\dots,j_N} = \sum_{i'_1,\dots,i'_N} \mathcal{P}_{i_1,\dots,i_N,i'_1,\dots,i'_N} \mathcal{P}^H_{i'_1,\dots,i'_N,j_1,\dots,j_N} = \sum_{i'_1,\dots,i'_N} \mathcal{P}_{i_1,\dots,i_N,i'_1,\dots,i'_N} \mathcal{P}^*_{j_1,\dots,j_N,i'_1,\dots,i'_N}. \tag{5.54}$$

Thus the trace is calculated as

$$\operatorname{tr}(\mathcal{P} *_N \mathcal{P}^H) = \sum_{i_1,\dots,i_N} \sum_{i'_1,\dots,i'_N} \mathcal{P}_{i_1,\dots,i_N,i'_1,\dots,i'_N} \mathcal{P}^*_{j_1,\dots,j_N,i'_1,\dots,i'_N} \tag{5.55}$$

and its derivative is calculated as

$$[\nabla_{\mathcal{P}} \operatorname{tr}(\mathcal{P} *_N \mathcal{P}^H)]_{k_1,\dots,k_N,k'_1,\dots,k'_N} = \frac{\partial}{\partial \mathcal{P}^*_{k_1,\dots,k_N,k'_1,\dots,k'_N}} \operatorname{tr}(\mathcal{P} *_N \mathcal{P}^H) = \mathcal{P}_{k_1,\dots,k_N,k'_1,\dots,k'_N} \tag{5.56}$$

which proves $\nabla_{\mathcal{P}} \operatorname{tr}(\mathcal{P} *_N \mathcal{P}^H) = \mathcal{P}$. Since $||\mathcal{P} - \mathcal{P}^{(0)}||^2$ is same as $\operatorname{tr}((\mathcal{P} - \mathcal{P}^{(0)}) *_N (\mathcal{P} - \mathcal{P}^{(0)})^H)$, we have $\nabla_{\mathcal{P}} ||\mathcal{P} - \mathcal{P}^{(0)}||^2 = \mathcal{P} - \mathcal{P}^{(0)}$. Also, based on the chain rule of derivatives (see Lemma 13 in Appendix B.8), we have $\nabla_{\mathcal{P}} \operatorname{tr}(\mathcal{F} *_N \mathcal{P} *_N \mathcal{P}^H) = \mathcal{F} *_N \mathcal{P}$. Thus the derivative of the Lagrangian from (5.53) with respect to $\mathcal{P}$ can be written as :

$$\nabla_{\mathcal{P}} \mathcal{L} = \mathcal{P} - \mathcal{P}^{(0)} + \mathcal{F} *_N \mathcal{P}. \tag{5.57}$$

The KKT condition is achieved by setting (5.57) to an all zero tensor which gives us :

$$(\mathcal{F} + \mathcal{I}_N) *_N \mathcal{P} = \mathcal{P}^{(0)} \tag{5.58}$$

Since $(\mathcal{F} + \mathcal{I}_N)$ is a pseudo-diagonal tensor with non-zero elements, we denote its inverse as $\mathcal{Z} = (\mathcal{F} + \mathcal{I}_N)^{-1}$. The pseudo-diagonal elements of tensor $\mathcal{Z}$ are represented as :

$$\mathcal{Z}_{\underline{i},\underline{i}} = z_{\underline{i}_c} = 1/(l_{\underline{i}_c} + 1), \forall \underline{i}_r \tag{5.59}$$

where $l_{\underline{i}_c}$ are the pseudo-diagonal elements of tensor $\mathcal{F}$. From (5.58) the tensor $\mathcal{P}$ can be written using $\mathcal{Z}$ as

$$\mathcal{P} = \mathcal{Z} *_N \mathcal{P}^{(0)}. \tag{5.60}$$

Thus the projection is given by the Einstein product between the tensors $\mathcal{Z}$ and $\mathcal{P}^{(0)}$. In order to find the elements of the pseudo-diagonal tensor $\mathcal{Z}$, we use the power constraints from (5.48). Substituting $\mathcal{P}$ from (5.60) into (5.48), we get:

$$\sum_{\underline{i}_r} (\mathcal{Z} *_N \mathcal{P}^{(0)} *_N (\mathcal{Z} *_N \mathcal{P}^{(0)})^H)_{\underline{i},\underline{i}} = P_{\underline{i}_c}, \quad \forall \underline{i}_c. \tag{5.61}$$

$$\sum_{\underline{i}_r} (\mathcal{Z} *_N \mathcal{P}^{(0)} *_N \mathcal{P}^{(0)H} *_N \mathcal{Z}^H)_{\underline{i},\underline{i}} = P_{\underline{i}_c}, \quad \forall \underline{i}_c. \tag{5.62}$$

Note that $\mathcal{Z}$ is a pseudo-diagonal tensor with only real elements, so $\mathcal{Z}^H = \mathcal{Z}$, and also $\mathcal{Z}_{\underline{i},\underline{j}}$ is non-zero only when $\underline{i} = \underline{j}$. Thus we get :

$$(\mathcal{Z} *_N \mathcal{P}^{(0)} *_N \mathcal{P}^{(0)H} *_N \mathcal{Z}^H)_{\underline{i},\underline{i}} = (\mathcal{Z} *_N \mathcal{P}^{(0)} *_N \mathcal{P}^{(0)H} *_N \mathcal{Z})_{\underline{i},\underline{i}} \tag{5.63}$$

$$= \sum_{\underline{j},\underline{j}'} \mathcal{Z}_{\underline{i},\underline{j}} \cdot (\mathcal{P}^{(0)} *_N \mathcal{P}^{(0)H})_{\underline{j},\underline{j}'} \cdot \mathcal{Z}_{\underline{j}',\underline{i}} \tag{5.64}$$

$$= \mathcal{Z}^2_{\underline{i},\underline{i}} \cdot (\mathcal{P}^{(0)} *_N \mathcal{P}^{(0)H})_{\underline{i},\underline{i}} \tag{5.65}$$

where the last equality follows from the pseudo-diagonality of $\mathcal{Z}$. On substituting (5.65) into (5.62) we get

$$\sum_{\underline{i}_r} \mathcal{Z}^2_{\underline{i},\underline{i}} \cdot (\mathcal{P}^{(0)} *_N \mathcal{P}^{(0)H})_{\underline{i},\underline{i}} = P_{\underline{i}_c}, \quad \forall \underline{i}_c. \tag{5.66}$$

Using (5.59) and (5.66), we get

$$z^2_{\underline{i}_c} \sum_{\underline{i}_r} (\mathcal{P}^{(0)} *_N \mathcal{P}^{(0)H})_{\underline{i},\underline{i}} = P_{\underline{i}_c}, \quad \forall \underline{i}_c, \tag{5.67}$$

$$z_{\underline{i}_c} = \sqrt{\frac{P_{\underline{i}_c}}{\sum_{\underline{i}_r} (\mathcal{P}^{(0)} *_N \mathcal{P}^{(0)H})_{\underline{i},\underline{i}}}}, \quad \forall \underline{i}_c, \tag{5.68}$$

which gives us all the elements of the pseudo-diagonal tensor $\mathcal{Z}$. To better understand this result, consider the case of sum power constraint for which $\underline{i}_c$ is empty, $P_{\underline{i}_c} = P$ represents the total power budget, and $\underline{i}_r = (i_1, \ldots, i_N)$. Since there is a single power constraint, we have a single Lagrange multiplier, and thus $\mathcal{Z}$ is a scaled identity tensor $z \cdot \mathcal{I}_N$. Thus we get :

$$z = \sqrt{\frac{P}{\sum_{\underline{i}}(\mathcal{P}^{(0)} *_N \mathcal{P}^{(0)H})_{\underline{i},\underline{i}}}} \tag{5.69}$$

$$= \sqrt{\frac{P}{\mathrm{tr}(\mathcal{P}^{(0)} *_N \mathcal{P}^{(0)H})}} \tag{5.70}$$

$$= \sqrt{\frac{P}{||\mathcal{P}^{(0)}||^2}} \tag{5.71}$$

i.e. in case of sum-power constraint since the tensor $\mathcal{Z}$ is a scaled identity tensor $z \cdot \mathcal{I}_N$, the solution to (5.47)-(5.48) is proportional to $\mathcal{P}^{(0)}$ upto a scaling factor z, chosen to satisfy the power constraint. This result when applied to a MIMO matrix channel under sum power constraint is consistent with the same solution for the MIMO case as presented in [107].

5.3.2 Calculation of the MMSE tensor

The MMSE tensor has a closed form expression in case of Gaussian input. For the numerical computation of the MMSE tensor for discrete inputs, the usual approach is to use Monte-Carlo simulations [106, 107]. If the input constellation size is Θ, then we can have a total of $K = \Theta^I$ possible input tensors $\boldsymbol{\mathcal{X}} \in \mathbb{C}^{I_1 \times \ldots \times I_N}$ where $I = \prod_{n=1}^{N} I_n$. We denote the realizations of input tensor as $\mathcal{X}^{(k)}$ for $k = 1, \ldots, K$. Finding the MMSE tensor entails averaging over all such input tensors. For a fixed tensor channel, corresponding to a given input tensor $\mathcal{X}^{(k)}$, we generate L noise tensors which are drawn from circularly symmetric complex Gaussian distribution, and compute the corresponding output tensor using (5.7) and denote it as $\mathcal{Y}^{(k,l)}$ for $l = 1, \ldots, L$. With large L, the MMSE tensor can be approximated using Monte Carlo runs as :

$$\mathcal{Q}_{\mathcal{E}} = \mathbb{E}[(\boldsymbol{\mathcal{X}} - \mathbb{E}[\boldsymbol{\mathcal{X}}|\boldsymbol{\mathcal{Y}}]) \circ (\boldsymbol{\mathcal{X}} - \mathbb{E}[\boldsymbol{\mathcal{X}}|\boldsymbol{\mathcal{Y}}])^*] \tag{5.72}$$

$$\approx \frac{1}{K}\sum_{k=1}^{K}\frac{1}{L}\sum_{l=1}^{L}\left(\mathcal{X}^{(k)} - \mathbb{E}[\boldsymbol{\mathcal{X}}|\boldsymbol{\mathcal{Y}} = \mathcal{Y}^{(k,l)}]\right) \circ \left(\mathcal{X}^{(k)} - \mathbb{E}[\boldsymbol{\mathcal{X}}|\boldsymbol{\mathcal{Y}} = \mathcal{Y}^{(k,l)}]\right)^*. \tag{5.73}$$

If the input distribution conditioned on the output is denoted by the function $p_{\boldsymbol{\mathcal{X}}|\boldsymbol{\mathcal{Y}}}(\mathcal{X}|\mathcal{Y})$, the conditional expectation $\mathbb{E}[\boldsymbol{\mathcal{X}}|\boldsymbol{\mathcal{Y}} = \mathcal{Y}]$ in (5.73) can be computed as :

$$\mathbb{E}[\boldsymbol{\mathcal{X}}|\boldsymbol{\mathcal{Y}} = \mathcal{Y}] = \sum_{\mathcal{X}} \mathcal{X} \cdot p_{\boldsymbol{\mathcal{X}}|\boldsymbol{\mathcal{Y}}}(\mathcal{X}|\mathcal{Y}) = \sum_{\mathcal{X}} \mathcal{X} \cdot \frac{p_{\boldsymbol{\mathcal{Y}}|\boldsymbol{\mathcal{X}}}(\mathcal{Y}|\mathcal{X}) p_{\boldsymbol{\mathcal{X}}}(\mathcal{X})}{p_{\boldsymbol{\mathcal{Y}}}(\mathcal{Y})} \tag{5.74}$$

$$= \frac{\sum_{\mathcal{X}} \mathcal{X} \cdot p_{\boldsymbol{\mathcal{Y}}|\boldsymbol{\mathcal{X}}}(\mathcal{Y}|\mathcal{X}) p_{\boldsymbol{\mathcal{X}}}(\mathcal{X})}{\sum_{\mathcal{X}'} \cdot p_{\boldsymbol{\mathcal{Y}}|\boldsymbol{\mathcal{X}}}(\mathcal{Y}|\mathcal{X}') p_{\boldsymbol{\mathcal{X}}}(\mathcal{X}')}. \tag{5.75}$$

The function $p_{\boldsymbol{\mathcal{X}}}(\cdot)$ denotes the input probability distribution and $p_{\boldsymbol{\mathcal{Y}}|\boldsymbol{\mathcal{X}}}(\cdot)$ denotes the output distribution conditioned on the input, which can be written as :

$$p_{\boldsymbol{\mathcal{Y}}|\boldsymbol{\mathcal{X}}}(\mathcal{Y}|\mathcal{X}) = \frac{1}{\pi^{J_1 \cdot J_2 \cdots J_M}} \exp\big(-||\mathcal{Y} - \mathcal{H} *_N \mathcal{P} *_N \mathcal{X}||^2\big). \tag{5.76}$$

In order to compute the MMSE tensor for a given channel $\mathcal{H}$ and precoder $\mathcal{P}$, it can be seen that the complexity increases exponentially with the input tensor size. First we consider the complexity of computing (5.76). Let $I = I_1 \cdot I_2 \cdots I_N$ and $J = J_1 \cdot J_2 \cdots J_M$. Based on the complexity of computing Einstein product as discussed in section 3.2.3, calculating $\mathcal{H} *_N \mathcal{P} *_N \mathcal{X}$ requires $\mathcal{O}(I^2 \cdot J)$ operations. Further, calculating $\mathcal{Y} - \mathcal{H} *_N \mathcal{P} *_N \mathcal{X}$ is a subtraction of tensors with J elements, and further calculating the norm square requires J multiplications and $J-1$ additions. Finding $(1/\pi^J)\exp(\cdot)$ can be seen as a single operation. Thus calculating the conditional pdf using (5.76) requires $\mathcal{O}(I^2 \cdot J)$ operations. The conditional pdf is substituted into (5.75) to find the conditional expectation for a given $\mathcal{Y}$, which requires summing over all the possible input tensors $\mathcal{X}$. Thus calculating (5.75) requires $\mathcal{O}(K \cdot I^2 \cdot J)$ operations where $K = \Theta^I$ denotes the possible input tensors with I elements and constellation size Θ. Further this conditional expectation is substituted into (5.73) which computes a double summation, where within each summation a tensor subtraction and an outer product is calculated. The subtraction is between I elements, and the outer product between tensors of same size (in this case $I_1 \times \ldots \times I_N$) requires I^2 multiplications. Thus complexity of subtraction and outer product combined is $\mathcal{O}(I^2)$. Subsequently, total number of operations required to compute the expression inside summations in (5.73) is $\mathcal{O}(K \cdot I^2 \cdot J \cdot I^2) = \mathcal{O}(K \cdot I^4 \cdot J)$. Further this summation takes place over all L values of l, and K values of k, thus the overall complexity of calculating the MMSE tensor for a fixed channel and precoder is $\mathcal{O}(K^2 \cdot L \cdot I^4 \cdot J)$. Since $K = \Theta^I = \Theta^{I_1 \cdot I_2 \cdots I_N}$, the complexity

$\mathcal{O}(K^2 \cdot L \cdot I^4 \cdot J)$ can also be written as $\mathcal{O}(\Theta^{2 \cdot I_1 \cdot I_2 \cdots I_N} \cdot L \cdot (I_1 \cdot I_2 \cdots I_N)^4 J_1 \cdot J_2 \cdots J_M)$. Thus the complexity of computing the MMSE tensor using Monte Carlo methods increases exponentially with the number of input elements denoted by $I = I_1 \cdot I_2 \cdots I_N$.

5.3.3 Calculation of Mutual Information for a given precoder

In order to assess the performance for a given precoder, we need to evaluate the mutual information. Since closed form expression for mutual information are not generally available for discrete inputs, we use Monte-Carlo method to calculate the mutual information for the given precoder. In the system model (5.7), since noise tensor $\boldsymbol{\mathcal{N}}$ is independent of the input tensor $\boldsymbol{\mathcal{X}}$, the mutual information is given as

$$\mathcal{I}(\boldsymbol{\mathcal{X}};\boldsymbol{\mathcal{Y}}) = \mathcal{H}(\boldsymbol{\mathcal{Y}}) - \mathcal{H}(\boldsymbol{\mathcal{Y}}|\boldsymbol{\mathcal{X}}) = \mathcal{H}(\boldsymbol{\mathcal{Y}}) - \mathcal{H}(\boldsymbol{\mathcal{N}}) \tag{5.77}$$

where the noise tensor is zero mean circularly symmetric Gaussian with identity covariance $\mathcal{I}_M$. Thus $\mathcal{H}(\boldsymbol{\mathcal{N}})$ is given by (using (3.1)) :

$$\mathcal{H}(\boldsymbol{\mathcal{N}}) = \log\Big((e\pi)^{J_1 \ldots J_M} \det(\mathcal{I}_M)\Big) = \log((e\pi)^{J_1 \cdots J_M}). \tag{5.78}$$

The output entropy can be calculated using Monte Carlo simulations to approximate the mutual information. For the case when input contains equiprobable elements, the mutual information can be calculated as established in the following lemma :

Lemma 7. *In the system model* (5.7)*, assuming input tensor elements are drawn from a set of discrete constellation points which are equiprobable, and given a precoder tensor* $\mathcal{P}$*, the mutual information can be numerically evaluated using :*

$$\mathcal{I}(\boldsymbol{\mathcal{X}};\boldsymbol{\mathcal{Y}}) = I_1 \cdots I_N \cdot \log\Theta - J_1 \cdots J_M \cdot \log(e) - \frac{1}{\Theta^{I_1 \cdots I_N}} \cdot \sum_{m=1}^{\Theta^{I_1 \cdots I_N}} \mathbb{E}_{\boldsymbol{\mathcal{N}}}\Big[\log \sum_{k=1}^{\Theta^{I_1 \cdots I_N}} \exp\big(-||\mathcal{H} *_N \mathcal{P} *_N (\mathcal{X}^{(m)} - \mathcal{X}^{(k)}) + \boldsymbol{\mathcal{N}}||^2\big)\Big] \tag{5.79}$$

where the expectation over noise can be carried out using Monte Carlo runs.

Proof. When input consists of equiprobable i.i.d symbols, the output distribution can be specified as :

$$p_{\boldsymbol{\mathcal{Y}}}(\mathcal{Y}) = \mathbb{E}_{\boldsymbol{\mathcal{X}}}[p_{\boldsymbol{\mathcal{Y}}|\boldsymbol{\mathcal{X}}}(\mathcal{Y}|\boldsymbol{\mathcal{X}})] = \frac{1}{\Theta^{I_1 \cdot I_2 \cdots I_N}} \sum_{k=1}^{\Theta^{I_1 \cdot I_2 \cdots I_N}} p_{\boldsymbol{\mathcal{Y}}|\boldsymbol{\mathcal{X}}}(\mathcal{Y}|\mathcal{X}^{(k)}) \tag{5.80}$$

Using (5.80), the output entropy can be calculated as :

$$\mathcal{H}(\boldsymbol{\mathcal{Y}}) = -\mathbb{E}_{\boldsymbol{\mathcal{Y}}}\Big[\log p_{\boldsymbol{\mathcal{Y}}}(\boldsymbol{\mathcal{Y}})\Big] = -\mathbb{E}_{\boldsymbol{\mathcal{Y}}}\Big[\log \frac{1}{\Theta^{I_1 \cdot I_2 \cdots I_N}} \sum_{k=1}^{\Theta^{I_1 \cdot I_2 \cdots I_N}} p_{\boldsymbol{\mathcal{Y}}|\boldsymbol{\mathcal{X}}}(\boldsymbol{\mathcal{Y}}|\mathcal{X}^{(k)})\Big] \tag{5.81}$$

$$= \log \Theta^{I_1 \cdots I_N} - \mathbb{E}_{\boldsymbol{\mathcal{Y}}}\Big[\log \sum_{k=1}^{\Theta^{I_1 \cdot I_2 \cdots I_N}} p_{\boldsymbol{\mathcal{Y}}|\boldsymbol{\mathcal{X}}}(\boldsymbol{\mathcal{Y}}|\mathcal{X}^{(k)})\Big]. \tag{5.82}$$

The conditional distribution $p_{\boldsymbol{\mathcal{Y}}|\boldsymbol{\mathcal{X}}}(\mathcal{Y}|\mathcal{X}^{(k)})$ can be specified using (5.76) and thus (5.82) becomes :

$$\mathcal{H}(\boldsymbol{\mathcal{Y}}) = \log \Theta^{I_1 \cdots I_N} - \mathbb{E}_{\boldsymbol{\mathcal{Y}}}\Big[\log \sum_{k=1}^{\Theta^{I_1 \cdots I_N}} \frac{\exp\big(-||\boldsymbol{\mathcal{Y}} - \mathcal{H} *_N \mathcal{P} *_N \mathcal{X}^{(k)}||^2\big)}{\pi^{J_1 \cdot J_2 \cdots J_M}}\Big] \tag{5.83}$$

$$= \log \Theta^{I_1 \cdots I_N} - \mathbb{E}_{\boldsymbol{\mathcal{X}},\boldsymbol{\mathcal{N}}}\Big[\log \sum_{k=1}^{\Theta^{I_1 \cdots I_N}} \frac{\exp\big(-||\mathcal{H} *_N \mathcal{P} *_N \boldsymbol{\mathcal{X}} + \boldsymbol{\mathcal{N}} - \mathcal{H} *_N \mathcal{P} *_N \mathcal{X}^{(k)}||^2\big)}{\pi^{J_1 \cdot J_2 \cdots J_M}}\Big] \tag{5.84}$$

$$= \log \Theta^{I_1 \cdots I_N} + \log(\pi^{J_1 \cdots J_M}) - \mathbb{E}_{\boldsymbol{\mathcal{X}},\boldsymbol{\mathcal{N}}}\Big[\log \sum_{k=1}^{\Theta^{I_1 \cdots I_N}} \exp\big(-||\mathcal{H} *_N \mathcal{P} *_N (\boldsymbol{\mathcal{X}} - \mathcal{X}^{(k)}) + \boldsymbol{\mathcal{N}}||^2\big)\Big]. \tag{5.85}$$

Since we assume that $\boldsymbol{\mathcal{X}}$ is independent of $\boldsymbol{\mathcal{N}}$, we get :

$$\mathcal{H}(\boldsymbol{\mathcal{Y}}) = \log \Theta^{I_1 \cdots I_N} + \log(\pi^{J_1 \cdots J_M}) - \mathbb{E}_{\boldsymbol{\mathcal{X}}}\Big[\mathbb{E}_{\boldsymbol{\mathcal{N}}}\Big[\log \sum_{k=1}^{\Theta^{I_1 \cdots I_N}} \exp\big(-||\mathcal{H} *_N \mathcal{P} *_N (\boldsymbol{\mathcal{X}} - \mathcal{X}^{(k)}) + \boldsymbol{\mathcal{N}}||^2\big)\Big]\Big] \tag{5.86}$$

$$= \log \Theta^{I_1 \cdots I_N} + \log(\pi^{J_1 \cdots J_M}) - \frac{1}{\Theta^{I_1 \cdots I_N}} \cdot \sum_{m=1}^{\Theta^{I_1 \cdots I_N}} \mathbb{E}_{\boldsymbol{\mathcal{N}}}\Big[\log \sum_{k=1}^{\Theta^{I_1 \cdots I_N}} \exp\big(-||\mathcal{H} *_N \mathcal{P} *_N (\mathcal{X}^{(m)} - \mathcal{X}^{(k)}) + \boldsymbol{\mathcal{N}}||^2\big)\Big]. \tag{5.87}$$

On substituting (5.87) and (5.78) into (5.77) gives us

$$\mathcal{I}(\boldsymbol{\mathcal{X}};\boldsymbol{\mathcal{Y}}) = \log\Theta^{I_1\cdots I_N} + \log(\pi^{J_1\cdots J_M}) - \frac{1}{\Theta^{I_1\cdots I_N}}\cdot \sum_{m=1}^{\Theta^{I_1\cdots I_N}} \mathbb{E}_{\boldsymbol{\mathcal{N}}}\Big[\log \sum_{k=1}^{\Theta^{I_1\cdots I_N}} \exp\big(-||\mathcal{H} *_N \mathcal{P} *_N (\mathcal{X}^{(m)} - \mathcal{X}^{(k)}) + \boldsymbol{\mathcal{N}}||^2\big)\Big] - \log((e\pi)^{J_1\cdots J_M}) \tag{5.88}$$

$$= I_1\cdots I_N \cdot \log\Theta - J_1\cdots J_M\cdot\log(e) - \frac{1}{\Theta^{I_1\cdots I_N}}\cdot \sum_{m=1}^{\Theta^{I_1\cdots I_N}} \mathbb{E}_{\boldsymbol{\mathcal{N}}}\Big[\log \sum_{k=1}^{\Theta^{I_1\cdots I_N}} \exp\big(-||\mathcal{H} *_N \mathcal{P} *_N (\mathcal{X}^{(m)} - \mathcal{X}^{(k)}) + \boldsymbol{\mathcal{N}}||^2\big)\Big]. \tag{5.89}$$

□

The complexity of calculating the mutual information for a given channel and precoder is also exponential in the size of the input tensor. Let $I = I_1\cdots I_N$ and $J = J_1\cdots J_M$. For a given noise realization, calculating the $\exp(\cdot)$ expressions inside the summation in (5.79) requires $\mathcal{O}(I^2\cdot J)$ operations. If we are averaging over L noise realizations, then it is clear from the summation over k and m in (5.79) that the overall complexity of finding the mutual information would be given as $\mathcal{O}(I^2\cdot J\cdot\Theta^I\cdot L\cdot\Theta^I)$ which is same as $\mathcal{O}(I^2\cdot J\cdot\Theta^{2\cdot I}\cdot L)$.

In the presence of high transmit power, the mutual information from (5.79) converges to a fixed value as shown in the following lemma.

Lemma 8. *As the transmit power increases, i.e. when $P\to\infty$, the mutual information from* (5.79), $\mathcal{I}(\boldsymbol{\mathcal{X}};\boldsymbol{\mathcal{Y}}) \to I_1\cdots I_N\log(\Theta)$.

Proof. Consider the expression inside $\exp(\cdot)$ in (5.79). As the transmit power becomes very large, i.e. $P\to\infty$, we can write :

$$||\mathcal{H} *_N \mathcal{P} *_N (\mathcal{X}^{(m)} - \mathcal{X}^{(k)}) + \boldsymbol{\mathcal{N}}||^2 \to \begin{cases} \infty, & \text{if } m\neq k \\ ||\boldsymbol{\mathcal{N}}||^2, & \text{if } m = k \end{cases} \tag{5.90}$$

Note that when $m=k$, for any given noise realization $\boldsymbol{\mathcal{N}} = \mathcal{N}$, the function $||\mathcal{H} *_N \mathcal{P} *_N (\mathcal{X}^{(m)} - \mathcal{X}^{(k)}) + \mathcal{N}||^2$ is same as $||\mathcal{N}||^2$. Also for $m\neq k$ when $P\to\infty$, then there is one realization of $\boldsymbol{\mathcal{N}}$ that diverges from the limit in (5.90) (when $\mathcal{N} = -\mathcal{H} *_N \mathcal{P} *_N (\mathcal{X}^{(m)} - \mathcal{X}^{(k)})$).

But the probability that $\boldsymbol{\mathcal{N}}$ takes this specific realization is 0. Hence the limit in (5.90) converges almost surely. Thus as $P \to \infty$, we have :

$$\sum_{k=1}^{\Theta^{I_1 \cdots I_N}} \exp\big(- ||\mathcal{H} *_N \mathcal{P} *_N (\mathcal{X}^{(m)} - \mathcal{X}^{(k)}) + \boldsymbol{\mathcal{N}}||^2\big) \to \exp(-||\boldsymbol{\mathcal{N}}||^2) \tag{5.91}$$

Since the noise tensor contains i.i.d. circularly symmetric zero mean unit variance elements, the expectation term in (5.79) can be written as (after substituting (5.91) in (5.79)) :

$$\mathbb{E}_{\boldsymbol{\mathcal{N}}}[\log \exp(-||\boldsymbol{\mathcal{N}}||^2)] \to \log(e) \sum_{j_1,\ldots,j_M} -\mathbb{E}[|\boldsymbol{\mathcal{N}}_{j_1,\ldots,j_M}|^2] = -\log(e) \cdot J_1 \cdot J_2 \cdots J_M. \tag{5.92}$$

Thus (5.79) becomes :

$$\begin{aligned}
\mathcal{I}(\boldsymbol{\mathcal{X}};\boldsymbol{\mathcal{Y}}) &\to I_1 \cdots I_N \cdot \log\Theta - J_1 \cdots J_M \cdot \log(e) - \frac{1}{\Theta^{I_1 \cdots I_N}} \cdot \sum_{m=1}^{\Theta^{I_1 \cdots I_N}} - \log(e) \cdot J_1 \cdot J_2 \cdots J_M \\
&= I_1 \cdots I_N \cdot \log\Theta - J_1 \cdots J_M \cdot \log(e) + \frac{1}{\Theta^{I_1 \cdots I_N}} \Theta^{I_1 \cdots I_N} \log(e) \cdot J_1 \cdot J_2 \cdots J_M \\
&= I_1 \cdots I_N \cdot \log\Theta.
\end{aligned} \tag{5.93}$$

□

5.4 Numerical Examples

Before we present numerical examples, first we summarize the steps required to find the optimal input precoder that maximizes the mutual information when the input is drawn from discrete constellations :

Step 1. Set $n = 0$ and initialize $\mathcal{P}^{(0)}$ to a choice which satisfies the power constraints, such as a precoder doing uniform power allocation.

Step 2. For a given precoder, calculate the MMSE tensor $\mathcal{Q}_{\boldsymbol{\mathcal{E}}}^{(n)}$ using (5.73). Note that calculating the MMSE tensor using (5.73) requires averaging over all the possible input tensors and several noise realizations. Corresponding to each possible input tensor $\mathcal{X}^{(k)}$, we generate output tensors $\mathcal{Y}^{(l)}$ for $l = 1, \ldots, L$ where L denotes the number of noise realizations. For any given input and output pair, the conditional pdf is calculated using (5.76) for the given precoder and channel, and further (5.75) is used to find the conditional expectation which is substituted in (5.73) to find the MMSE tensor.

Step 3. Based on the MMSE tensor calculated in step 2, update the precoder using (5.46).
Step 4. Set $n = n + 1$ and go to step 2 and repeat until a desired convergence criteria is met.
Step 5. Calculate the mutual information for the updated precoder using (5.76). Note that mutual information can be calculated after each precoder update for checking the convergence. However, mutual information is not needed for updating the precoder itself, and convergence criterion for the iterative equation can also be set using the mean square error as we will illustrate through examples in this section.

For the numerical results, the channel elements are generated using circularly symmetric complex Gaussian distribution with zero mean and unit variance unless otherwise stated. Also the input symbols are drawn from equiprobable distribution unless otherwise specified. For the gradient ascent, we use a step size of $\nu = 0.01$. The total transmit power constraint is denoted by P and the noise tensor contains i.i.d. circularly symmetric complex Gaussian entries with zero mean and variance $\sigma^2 = 1$. All the results are presented by averaging over S channel realizations. For a given channel and precoder the calculation of the MMSE tensor and Mutual Information is carried out using L noise realizations. The values of S and L are kept as 100, unless otherwise stated.

5.4.1 Capacity for selected input constellations

In this section, we numerically analyze the capacity of order-4 tensor channels when the input is constrained to be drawn from a fixed discrete constellation. Most of the work in literature which presents the MIMO matrix channel capacity with discrete inputs such as [106, 109], make use of the BPSK constellation to illustrate the performance. One of the reasons why a simple constellation such as BPSK is chosen is because the simulation of the MMSE or the mutual information via Monte Carlo methods becomes computationally very expensive with increasing size of the constellation. For the tensor case, we showed in section 5.3.2 and 5.3.3, that the complexity of computing MMSE from (5.73) and mutual information from (5.79) increases significantly with constellation size and number of input elements. Hence in this work, to illustrate the main concepts we start with a simple constellation such as BPSK as used in [106, 109] which has 1 bit per symbol, and thus $\Theta = 2$. However, to check the performance of the multi-linear precoders, we also run simulations using some higher constellations such as Quadrature Phase Shift Keying (QPSK) which has 2 bits per symbol and thus $\Theta = 4$. In addition, we also test two different constellations which have 3 bits per symbol and thus $\Theta = 8$, namely 8-Phase Shift Keying (PSK) and 8-Amplitude Phase Shift Keying (APSK).

The difference between the 8-PSK and 8-APSK constellation is that in 8-PSK all the 8 constellation points lie on the same circle whereas in 8-APSK the symbols are arranged on concentric circles of different radius with a constant phase offset. The constellation for 8-PSK is shown in Figure 5.1 where all the points lie on unit circle. Whereas, the constellation for 8-APSK, as shown in Figure 5.2, has phase shifted points lying on two circles of different radius. The presented 8-APSK can also be called 4-4 8APSK since it has 4 points on the inner circle and 4 on the outer. In this way, APSK combines both amplitude and phase shift keying. The advantage of a combination of amplitude and phase shift keying is that the minimum euclidean distance between the constellation points is increased for the same average symbol energy. However at the same time 8-APSK leads to higher peak to average power ratio than 8-PSK.

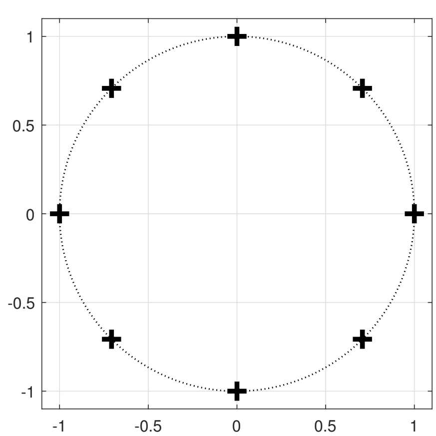


Fig. 5.1: 8 PSK Constellation.

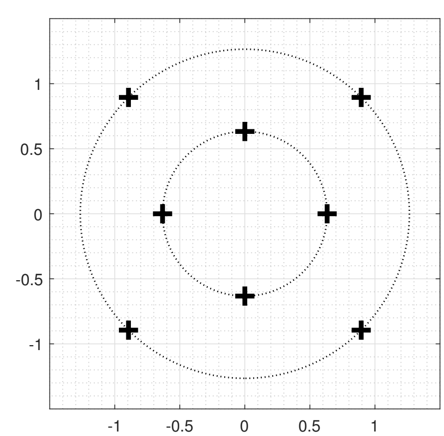


Fig. 5.2: 8 APSK Constellation.

We first consider the example of BPSK constellation. Consider an order-4 tensor channel of size $2 \times 2 \times 2 \times 2$ corresponding to an order-2 input of size 2×2, and order-2 output of size 2×2. Under different power budgets P, we iteratively update the input precoder using (5.46) and plot the mutual information computed through (5.79) in each iteration in Figure 5.3. As can be seen in Figure 5.3, the mutual information increases with iterations and reaches a saturation after some point. This saturated level can be considered as the maximum mutual information or the capacity under BPSK constrained input for the given power budget and equiprobable input distribution. As P increases the saturated mutual information value also increases.

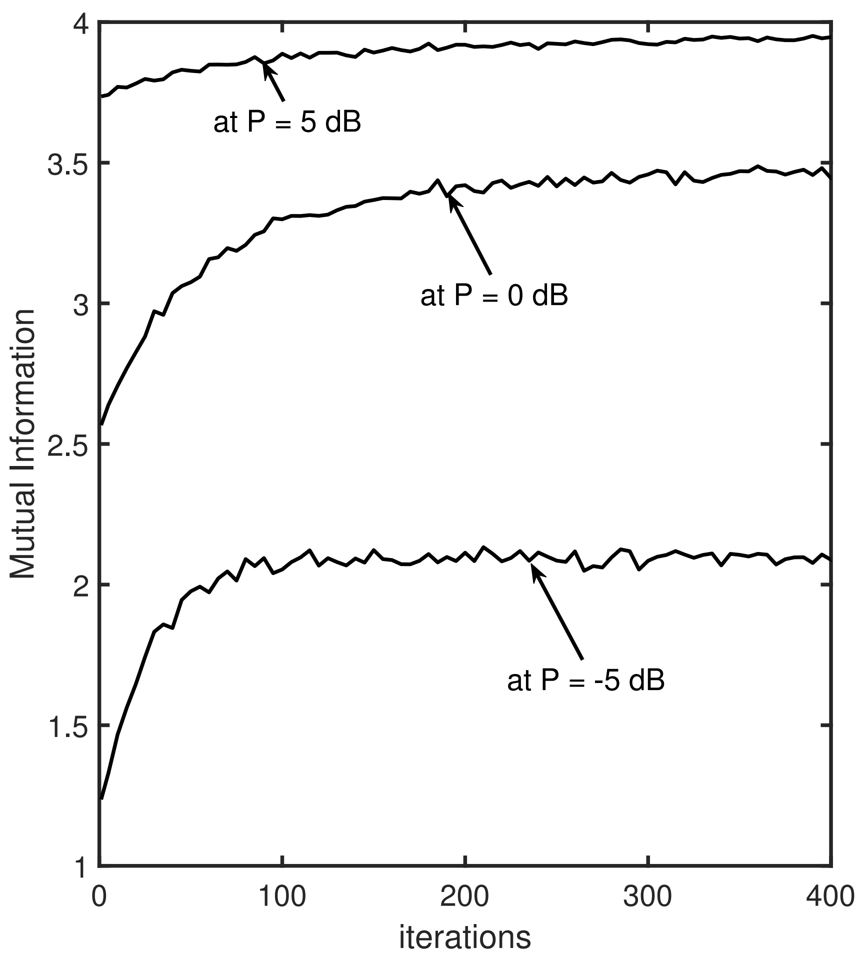


Fig. 5.3: Convergence of Mutual Information (in bits) for order 4 channel with BPSK input.

Note that we present the convergence of mutual information against iterations as the mutual information is an entity of interest for our purpose which can be used as a criteria for terminating the iterations. However, we do not actually need to calculate the mutual information at each iteration. For updating the precoder, we only need an updated MMSE tensor $\mathcal{Q}_{\boldsymbol{\mathcal{E}}}$. Hence, a possible termination criteria for the iterative equation could also be established by observing the MMSE tensor. In Figure 5.4, we present the normalized mean square error against iterations. The normalized mean square error is the ratio of the trace of the the error covariance tensor and the total transmit power, calculated as :

$$\texttt{NMSE} = \frac{1}{P}\,\mathrm{tr}(\mathcal{Q}_{\boldsymbol{\mathcal{E}}}). \tag{5.94}$$

It can be seen in Figure 5.4 that the mean square error reduces as the iteration increases and also reaches a saturation. The saturation level is lower for higher transmit power P. On comparing the convergence of mutual information in Figure 5.3 and of mean square error in Figure 5.4, we see that as the mean squared error decreases, the mutual information increases, and both entities saturate around the same number of iterations. Thus the convergence criteria for the precoder update iterative equation can be set by considering the saturation of either of the two parameters, mutual information or mean square error.

Next, in addition to the constraint on input signal constellation, we also consider two different types of power constraints : sum power constraint, and per-domain power constraints where one of the two domains has individual power budgets defined by $P_1 = x \cdot P$ and $P_2 = (1-x) \cdot P$ with $x = 0.9$. For both these constraints, we plot the capacity when the input is constrained to be drawn from BPSK constellation. We also plot the capacity with the same power constraints but under Gaussian input signalling for comparison. Figure 5.5 shows the comparison of these capacities and thus illustrates the loss of capacity due to the additional constraints on input constellation. Notice that the use of BPSK constellation limits the capacity, which for high values of P saturates around 4 bits/channel-use as per Lemma 8 and also seen in Figure 5.5. The gap between the capacity with BPSK input and the channel capacity under no input constellation constraint (in which case we have Gaussian signalling) is significantly large at higher transmit powers. Although, at lower P, we observe that capacity is almost same whether we have the additional signalling constraints

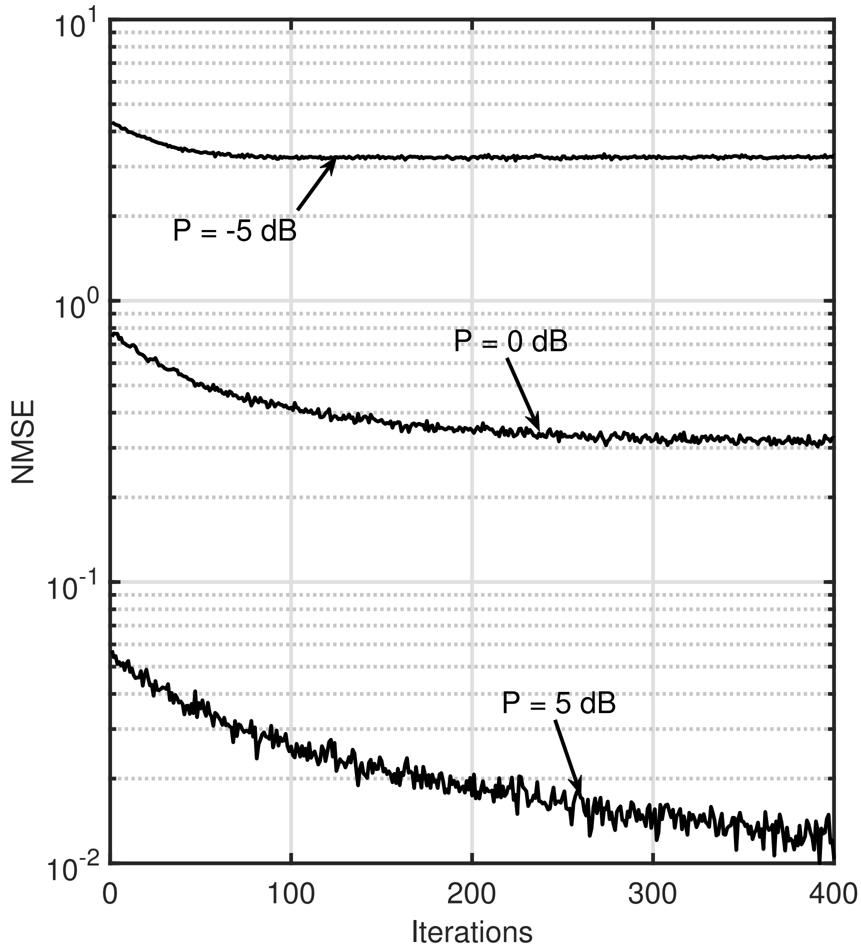


Fig. 5.4: Convergence of Mean Square Error for order 4 channel with BPSK input.

on input or not. Also, with per-domain power constraints the capacity is lower than the capacity achieved with sum power constraint for both Gaussian and BPSK inputs. This is as observed in Chapter-3 where the additional constraints on the input power makes the feasible set smaller and thus lowers the capacity. Figure 5.5 highlights the dominant behaviour of several different constraints. If the total power budget P is low, then any additional constraint on the input constellation does not affect the capacity much. Hence at low SNRs, the limited power budgets are the dominant constraints whereas the signalling constellation constraints are not too relevant. However, at high SNRs, the input signalling constellation constraints severely limits the capacity and thus acts as the dominant constraint. Increasing the power budget P, or having individual domain power constraints, does not change the capacity beyond a point with discrete input signal constellation constraints as the capacity saturates.

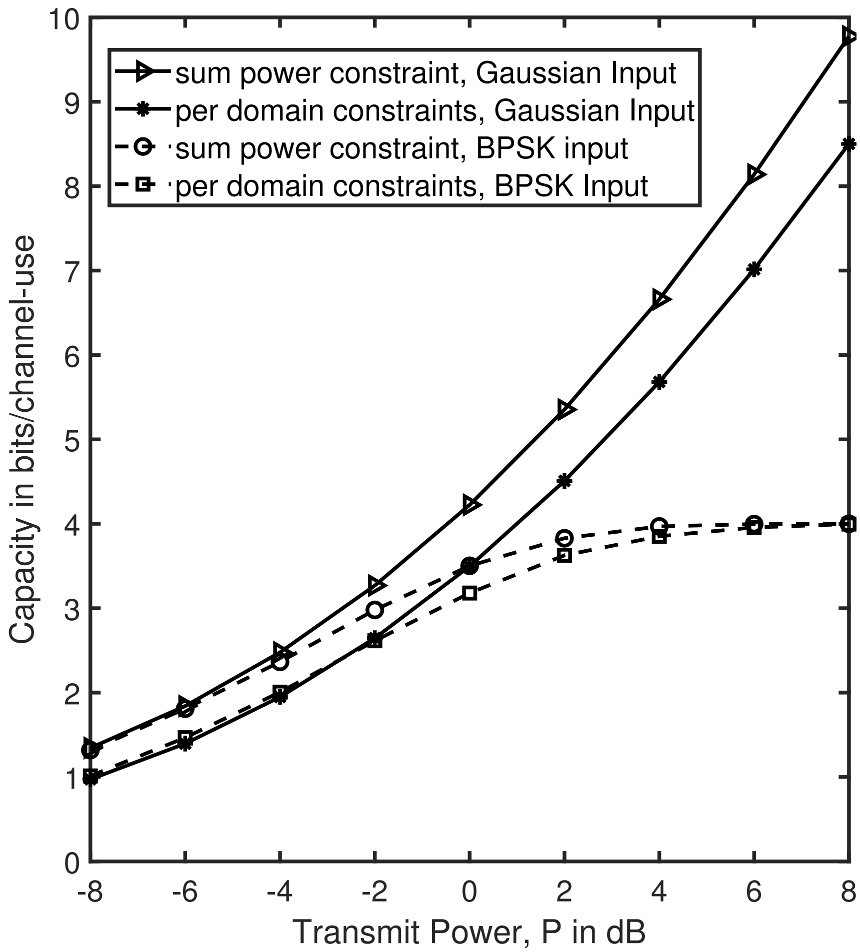


Fig. 5.5: Capacity for order 4 tensor channel with BPSK and Gaussian inputs.

In Figure 5.5, we considered a specific per-domain power constraint corresponding to $x = 0.9$. To analyze the capacity for a larger variety of power constraint, in Figures 5.6 and 5.7 we plot the tensor channel capacity for different values of x for both BPSK and Gaussian inputs. Note that different values of x such that $0 < x < 1$ generate different type of per-domain power constraints. Figure 5.6 shows the capacity behaviour for lower power budgets ($P = 0, -2$ dB), where it can be seen that the capacity for both the Gaussian input and the BPSK input changes with change of per-domain power constraint parameter x. The gap between the curves for Gaussian and BPSK increases as P grows. But changing x does change the capacity for both kind of inputs. On contrasting this with Figure 5.7, which shows the same plot for higher power budgets ($P = 2, 4$ dB), we observe that the capacity with BPSK input does not get much affected by changing the parameter x. In fact, for sufficiently high value of P, the capacity with BPSK input is almost a straight line when plotted against x, whereas with Gaussian input the capacity shows an arch like curve against x. This further establishes that for high transmit powers under discrete signalling constraints, the per-domain power constraints are not too significant as the discrete input constraint is the dominant factor that limits the capacity.

Next we consider the tensor channel capacity when every individual element of the input tensor is under a different power constraint. If the total available power is P, then as used in previous examples $P_1 = x \cdot P$ and $P_2 = (1 - x) \cdot P$. Further, $P_{11} = y \cdot P_1, P_{12} = (1 - y) \cdot P_1, P_{21} = y \cdot P_2$ and $P_{22} = (1 - y) \cdot P_2$. Thus, with $0 < x, y < 1$, $P_{11}, P_{12}, P_{21}, P_{22}$ denote the individual power constraints on all the four elements of the 2×2 input tensor such that $P_{11}+P_{12}+P_{21}+P_{22} = P$. Different choices of x and y lead to different per-element power constraints such that total power budget is P. Figures 5.8, 5.9, 5.10, and 5.11 shows the capacity of the tensor channel for Gaussian and BPSK inputs against different values of x and y, for $P = -2, 0, 2, 4$ dB respectively. In all the four figures it can be seen that capacity with Gaussian input is higher than the BPSK input. However, in Figure 5.8 the curves are fairly close to each other, whereas in Figure 5.11 the gap between the two curves is very wide. With increasing P, the gap between the two curves for Gaussian and BPSK input gradually widens as can be observed in Figure 5.8, 5.9, 5.10, and 5.11. It is also important to note that the variation with x and y for smaller values of P as shown in Figures 5.8 and 5.9 is similar for both Gaussian and BPSK inputs. But for higher values of P, the surface curve corresponding to BPSK input tends to be a flat surface as opposed to the Gaussian input where the curve shows variation with x and y. This further illustrates that as P increases, individual element power constraints do not play a significant role in determining the channel capacity if the input is drawn from discrete constellations.

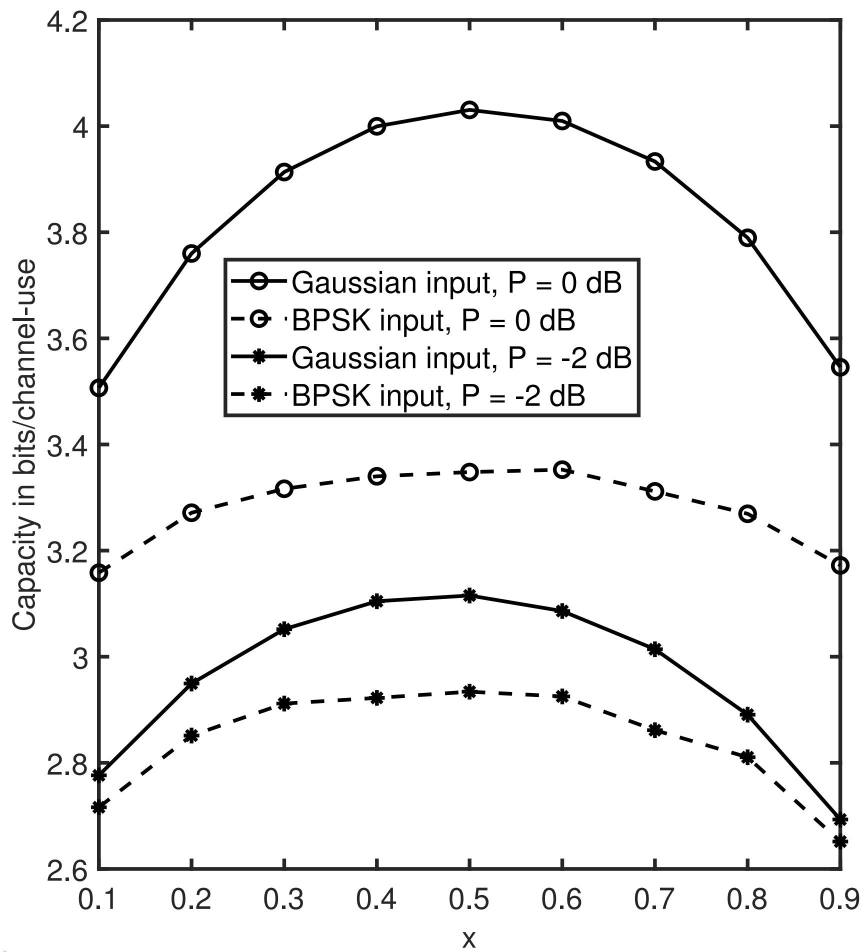


Fig. 5.6: Capacity under per domain power constraints with BPSK and Gaussian inputs.

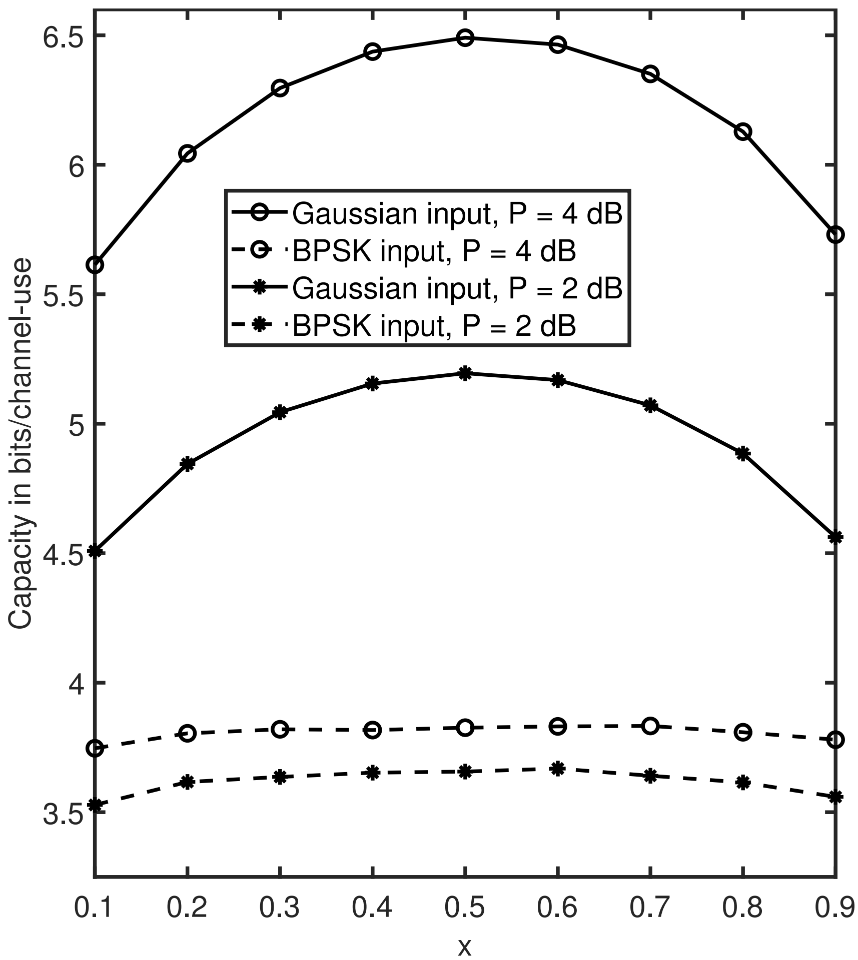


Fig. 5.7: Capacity under per domain power constraints with BPSK and Gaussian inputs.

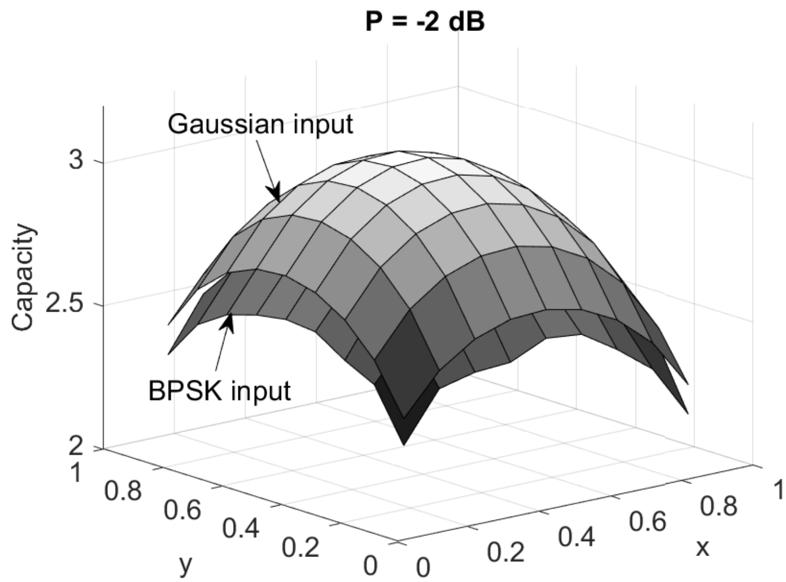


Fig. 5.8: Capacity (in bits/channel-use) for various per element power constraints at $P = -2$ dB with BPSK and Gaussian inputs.

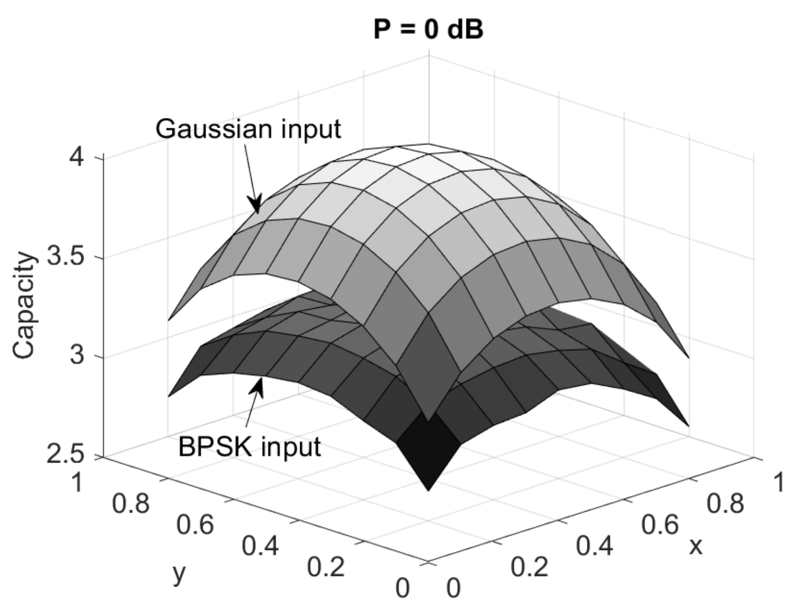


Fig. 5.9: Capacity (in bits/channel-use) for various per element power constraints at $P = 0$ dB with BPSK and Gaussian inputs..

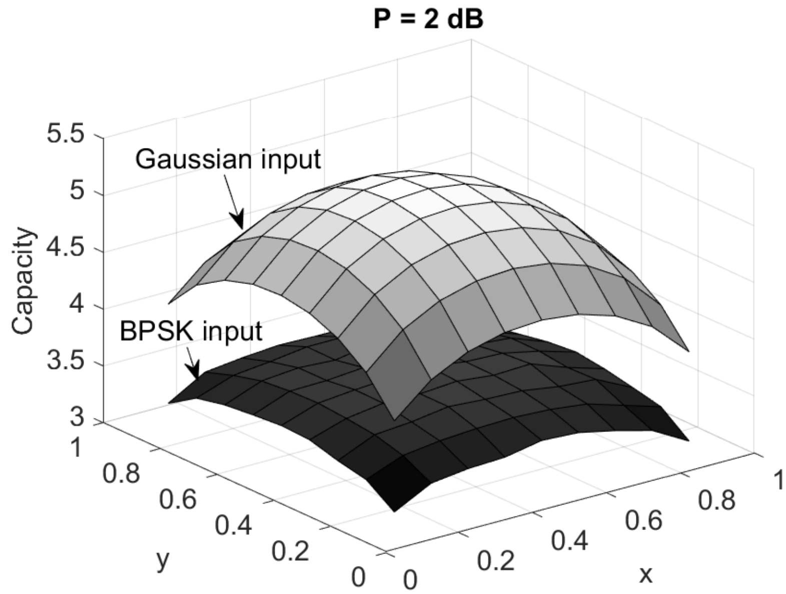


Fig. 5.10: Capacity (in bits/channel-use) for various per element power constraints at $P = 2$ dB with BPSK and Gaussian inputs.

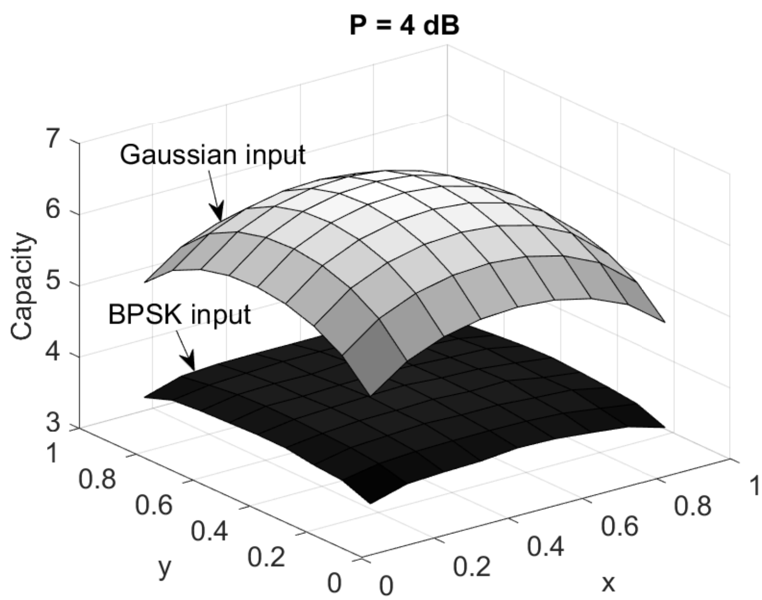


Fig. 5.11: Capacity (in bits/channel-use) for various per element power constraints at $P = 4$ dB with BPSK and Gaussian inputs..

Further to illustrate the behavior of capacity for another constellation, we use QPSK as the input signalling constraint. Figure 5.12 shows the capacity against P for sum power and per-domain power constraint (with $x = 0.9$) when the input is drawn from a QPSK constellation and compares it with Gaussian input signalling. Similar to Figure 5.5, here also it can be observed that the capacity under QPSK input is significantly lower than the Gaussian input and reaches a saturation for large values of P. In this case the value saturates around 8 bits/channel-use as per Lemma 8, however this saturation occurs at a higher value of P as compared to the BPSK case. Also increasing P beyond a certain point does not increase the capacity any more. For lower values of P, the capacity with QPSK input is almost same as the capacity with the Gaussian input. So here also, we observe that the input constellation constraint becomes the dominating factor at higher SNRs, although at low SNR the limited power budget remains the dominant constraint.

Further to illustrate the behaviour of capacity for different constellations of same size, we compare 8-PSK with 8-APSK under sum power constraints. As can be seen in Figure 5.13, the capacity for the discrete constellations is nearly same as compared with Gaussian input for low transmit powers. However for high transmit power the capacity with Gaussian input increases significantly, while the capacity for constellation with 8-PSK with 8-APSK saturates at about 12 bits/channel-use. On comparing the capacity curves for the two discrete constellations 8-PSK with 8-APSK, which have the same size, it can be observed that at low transmit power the curves are indistinguishable. But for moderate values of transmit power, the capacity with 8-PSK is slightly lower than that of 8-APSK. For instance, at 8 dB, the capacity for 8-APSK is approximately 0.5 bits/channel-use lower than the capacity for 8-PSK. This difference can be attributed to the fact that the average distance between neighbouring constellation points in APSK is larger than PSK. This can produce better mean square performance at the receiver, and thus lead to a larger mutual information. However, at high enough transmit power, both the constellations saturate to same level since they have the same size.

Upon comparing Figures 5.5, 5.12, and 5.13, it can be seen that the channel capacity for discrete constellations is always similar to the case with Gaussian input signalling at

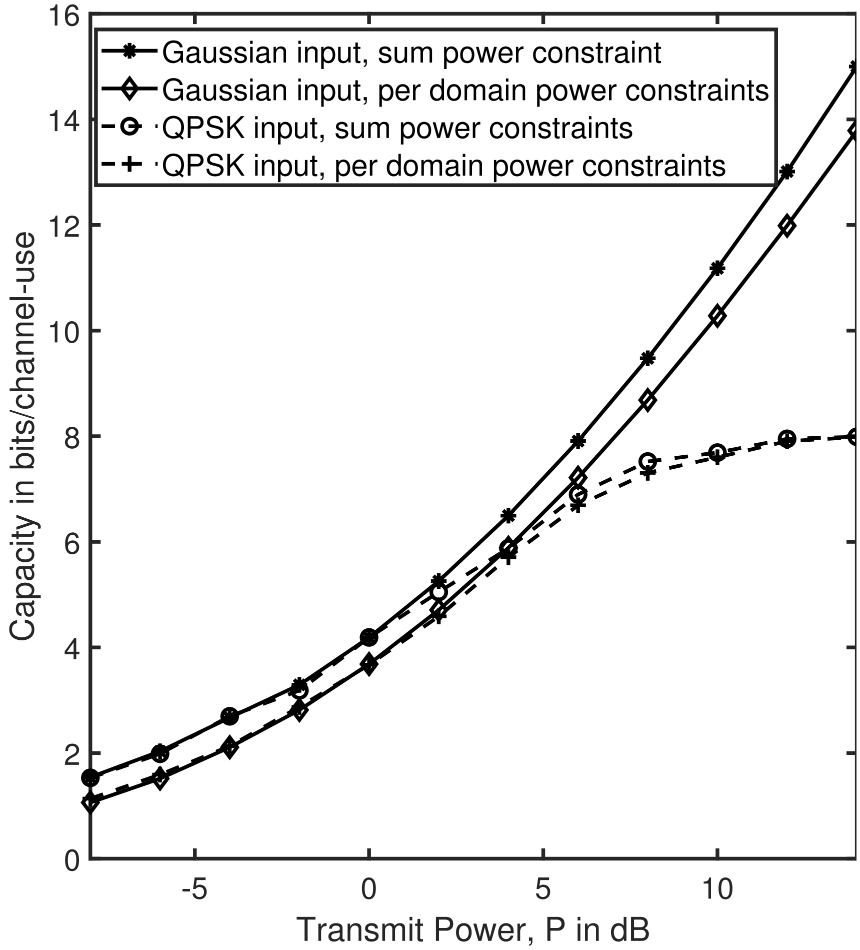


Fig. 5.12: Capacity of order 4 tensor channel with QPSK and Gaussian inputs .

low enough SNR, and starts to significantly deviate only after a moderate value of SNR. For BPSK, the curves started to deviate around -4 dB, for QPSK around 2 dB, and for 8-PSK or 8-APSK around 5 dB. The capacity saturation value also increases on increasing the constellation size. Thus it can be concluded that as the constellation size grows, the performance gap between the discrete constellations and the Gaussian input decreases.

5.4.2 Capacity for different tensor channel order

So far we considered a fixed tensor channel and compared different power constraints and constellations. Now we consider the capacity against various orders of tensor channels

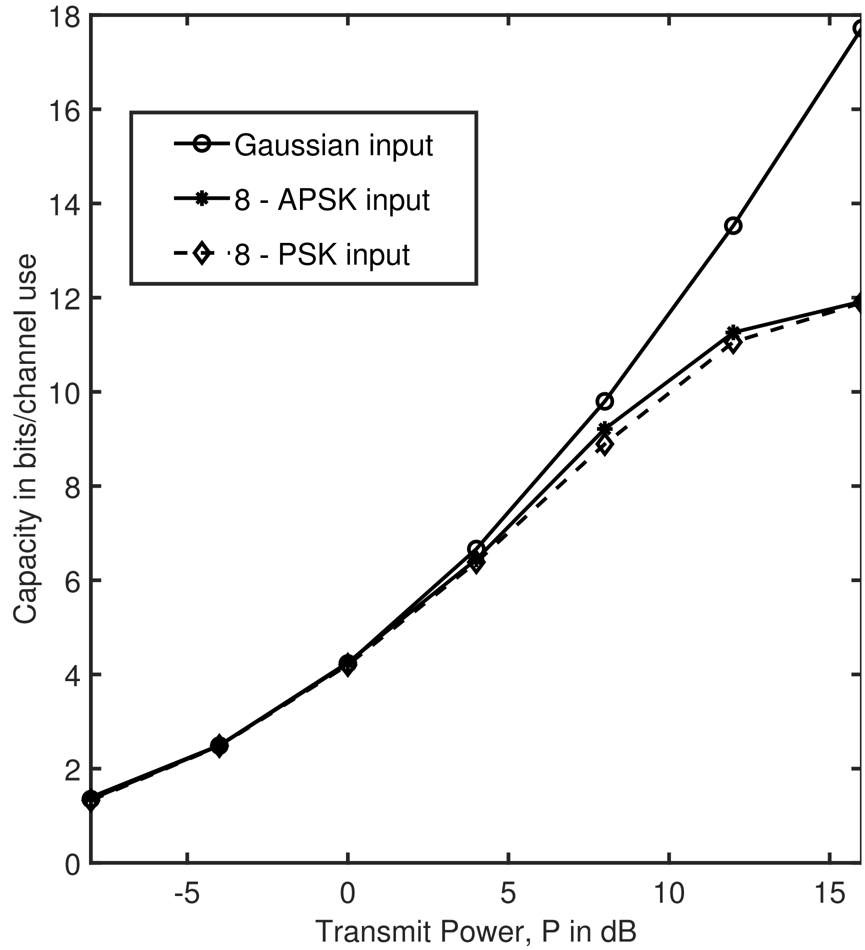


Fig. 5.13: Capacity of order 4 tensor channel with 8 PSK, 8 APSK and Gaussian inputs .

for discrete inputs under sum power constraint. The results are generated using the same procedure as descried in the beginning of section 5.4. Figure 5.14 illustrates the behavior of the mutual information as the precoder is updated iteratively using (5.46) at $P = 0$ dB for BPSK input. The mutual information convergence is plotted for different order of tensor channels. If the channel is even order $2N$, then both input and output are taken to be order N. For odd order $2N + 1$ tensor channels, we take input to be order $N + 1$ and output to be order N. To be specific, corresponding to order $3, 4, 5,$ and 6 channels, the input output pairs are taken as $(2, 1), (2, 2), (3, 2),$ and $(3, 3)$ respectively. The dimensions of the individual channel domains is kept as 2 for all the examples presented in this section. It can be seen in Figure 5.14 that the mutual information increases and saturates after several iterations for any order tensor channel. This saturation level indicates the capacity of the tensor channel. For the same P, the saturation level is higher for higher order channels. It is important to note here that the channel entries are generated as in previous section using circularly symmetric Gaussian with zero mean unit variance, hence the channel is not normalized and provides a power gain at the receiver which increases with increasing channel size.

Further, Figure 5.15 presents the behavior of mutual information as the precoder is iteratively updated for $P = 0$ dB when the tensor channel is normalized to provide unit power gain at the receiver. The channel elements in this case are generated using i.i.d. circularly symmetric complex Gaussian with zero mean and variance $1/(J_1 \cdots J_M)$, where $J_1 \cdots J_M$ is the number of elements in the output tensor. Since the channel elements are generated with lower variance, its individual components are weaker as compared to the non-normalized case and hence the saturation values of the mutual information are lower than the corresponding values in Figure 5.14. Note that Figure 5.14 and 5.15 are presented on different y-scales, thus the curves in Figure 5.15 do not appear as smooth as Figure 5.14. Also, it is to be noted that in Figure 5.15 the saturation value reached with order 3 is larger than that of order 4, and with order 5 is larger than that of order 6. For a normalized channel, as discussed in section 3.3.2, the channel components get weaker as the number of receive elements increases. The order 5 channel corresponds to order 3 input and order 2 output, whereas order 6 channel corresponds to an order 3 input and order 3 output. Thus with lower output order (case of order 5 tensor channel), the channel components are stronger as compared to the order 6 channel because of lower value of $J_1 \cdots J_M$. However, the results in Figures 5.14 and 5.15 are corresponding to a specific input output configuration where for odd order channels, the input is order $N + 1$ and output is order N. A different configuration of the input and output can change these results. We further elaborate on the role of specific input output configuration for a fixed order channel, both normalized and non-normalized, through more examples.

So far all the numerical results presented in the previous section and in this section

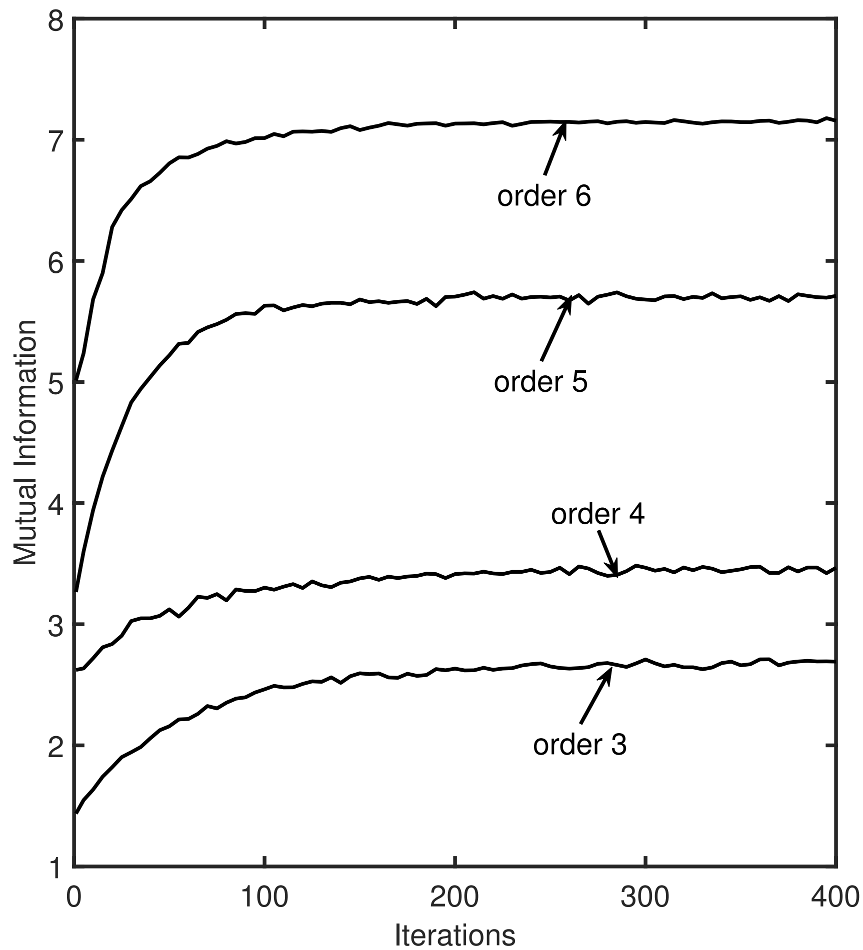


Fig. 5.14: Convergence of Mutual Information (in bits) for different Order channel (non-normalized) with BPSK input at $P = 0$ dB.

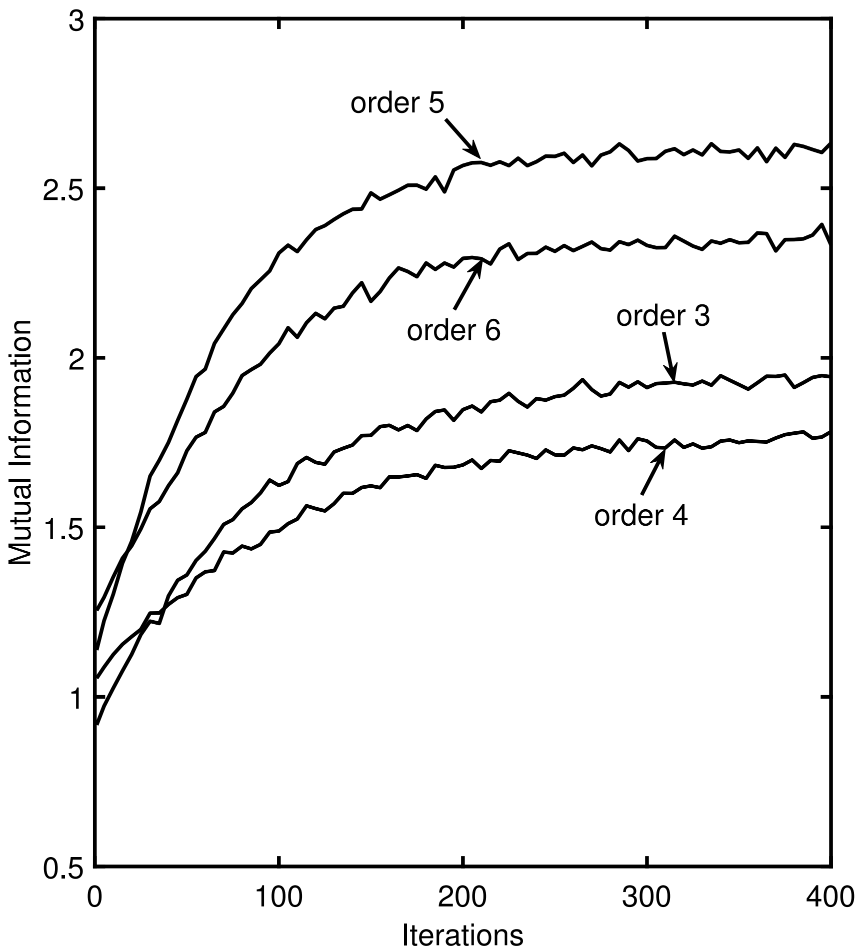


Fig. 5.15: Convergence of Mutual Information (in bits) for different Order Normalized Tensor channels with BPSK input at $P = 0$ dB.

were averaged over 100 channel realizations, where for each channel we used 100 noise realizations to compute the MMSE tensor and the mutual information. The complexity of computing the MMSE tensor and the mutual information increases exponentially with the size of the input tensor as discussed in section 5.3.2 and 5.3.3. Thus to save simulation time, for further examples with higher order inputs we use lower number of random realizations for averaging the results. The parameters used henceforth are listed in Table 5.1. For order 5 and 6 cases, we average over only 50 channel and noise realizations, while for order 7 and 8 cases, we average over only 5 random channel realizations and 25 noise realizations. A primary reason for choosing such lower numbers for higher orders is to save simulation time. However, due to large number of elements within each tensor realization for order 7 and 8 cases, these smaller number of samples still gives satisfactory results. This will be established later through examples, where averaging over smaller number of samples gives us a good approximation of an average taken over larger samples for higher order cases.

Table 5.1: Monte Carlo Simulation Parameters.

Constellation	Channel Order	Channel realizations	Noise realizations
BPSK	≤ 4	100	100
BPSK	$5, 6$	50	50
BPSK	$7, 8$	5	25
QPSK	≤ 4	100	100
QPSK	$5, 6$	15	25

Figure 5.16 illustrates the behaviour of the capacity against increasing order of the tensor channels for different power budgets P when the channel elements are i.i.d. circularly symmetric complex Gaussian with zero mean and unit variance. We consider two different configurations of input and output tensors for odd order $(2N+1)$ tensor channels : case 1) input order $N+1$, output order N, and case 2) output order $N+1$, input order N. Case 1 is represented by dashed lines in the figure and case 2 by solid (or dotted for $P = 5$ dB case) lines. For even order channels, the two cases are the same, thus the points coincide. With input drawn from a discrete constellation of size Θ, the capacity can be specified as:

$$C = \max\{\mathcal{H}(\boldsymbol{\mathcal{X}}) - \mathcal{H}(\boldsymbol{\mathcal{X}}|\boldsymbol{\mathcal{Y}})\} \leq \max\{\mathcal{H}(\boldsymbol{\mathcal{X}})\} = \log(\Theta^{I_1 \cdots I_N}). \tag{5.95}$$

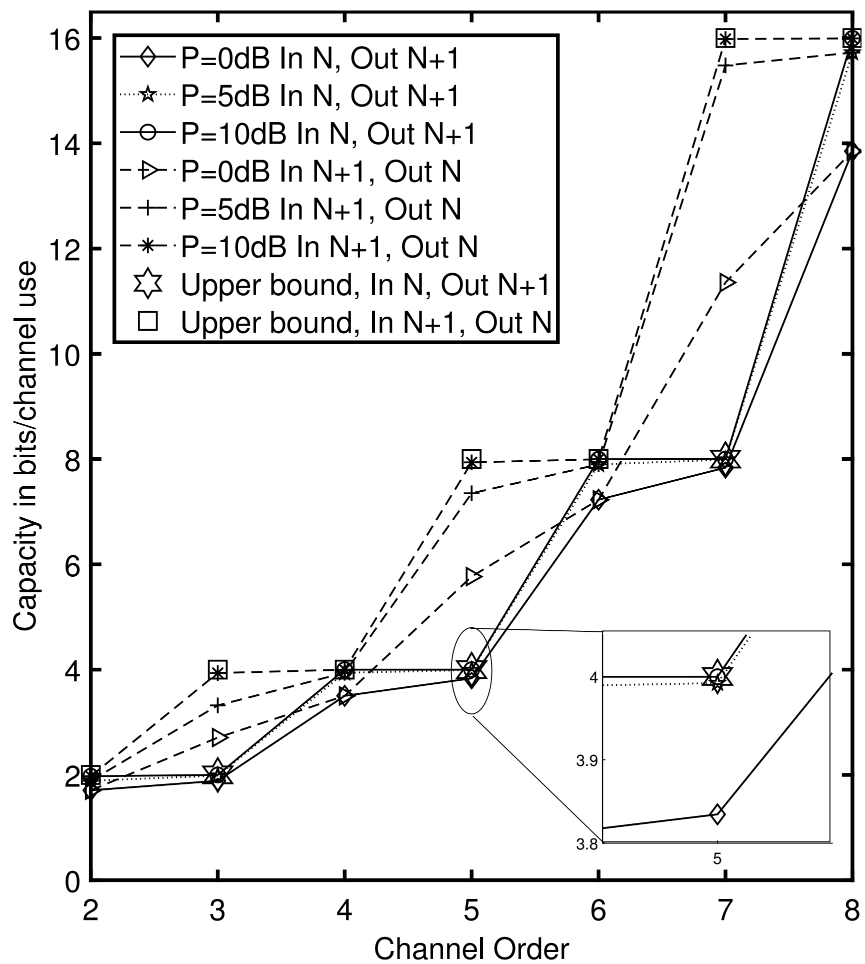


Fig. 5.16: Capacity vs Order of tensor Gaussian channel (non-normalized) for BPSK input.

Thus the capacity of a channel is always upper bounded by the maximum of the input entropy. In Figure 5.16, the upper bound from (5.95) is indicated using a square marker for case 1 and a hexagram marker for case 2. Since for even order channels, the two bounds are exactly same, for clarity of representation, in Figure 5.16 we mark the bound for even order cases only with a square marker. As can be seen in Figure 5.16, the capacity with case 1 is higher than that of case 2 because of larger number of input elements in case 1. Also note that the capacity increases with increase in P for both cases, however for odd order tensor channels, the difference between the capacity with case 1 and case 2 also increases with increase in P. This is because as P increases, the capacity with BPSK input tends to reach a saturation. This saturation depends on the size of the input constellation and the number of input elements as specified by (5.95). Hence a higher input order configuration leads to a higher capacity for a fixed order tensor channel. The bound in (5.95) can be met with sufficiently high transmit power when the input tensor contains equiprobable i.i.d. elements as established in Lemma 8.

Thus for discrete inputs, the capacity is bounded by the size of the input constellation and the number of input elements, and tends to this bound with increasing P. This can be also observed from Figure 5.16, since the square markers and the hexagram markers lie almost on top of the $P = 10$ dB points which shows that the capacity saturation is reached at this power level. On comparing the channel order 4 and 5 for high values of P for case 2 (input order less than output order), we see that the capacity does not increase much and stays almost at 4 bits/channel-use because the order of the input tensor (order 2) did not change in going from channel order 4 to order 5 in this case. However, for case 1 the order of the input tensor goes from 2 to 3 while moving from channel order 4 to 5, which causes a significant increase in the capacity. This increase becomes more substantial as P increases. At sufficiently high P, the capacity almost reaches 8 bits/channel-use whenever the input order is 3, and 4 bits/channel-use whenever the input order is 2. It is important to note that in this example, since the channel is not normalized, the impact of changing the input and output order (case 1 and case 2), does not change the strength of the individual elements of the channel and the difference in capacity arises from the constellation constraint at the input and the size of the input. Also, such a channel provides power gain at the receiver as discussed in section 3.3.2. Thus power levels such as 5 dB and 10 dB are sufficiently high enough to make the capacity saturate. In fact the curves for 5 dB and 10 dB almost overlap for case 2 in Figure 5.16 because of the capacity saturation with increasing power budget. For case 1 also, the curves for 5 dB and 10 dB are closer but they do not overlap as the saturation value of the capacity is higher because of higher input order. For instance, corresponding to order 5 tensor channel, case 1 can lead to 8 bits/channel-use (since input order is 3) with sufficient power. This is double the saturation value for case 2 which is only

4 bits/channel-use (since input order is 2). For case 2, around 5 dB is sufficient to reach the saturated value, whereas for case 1 since the saturation value itself is higher, much higher power is required to reach the saturation. At around 10 dB, the capacity reaches about 8 bits/channel-use for case 1. This implies that with discrete constellations, depending on the input configuration, transmitting signal beyond a certain power would not lead to higher capacity. Thus the available power budget can be adjusted depending on the input configuration. Next we will analyze the channel capacity in case of normalized channels.

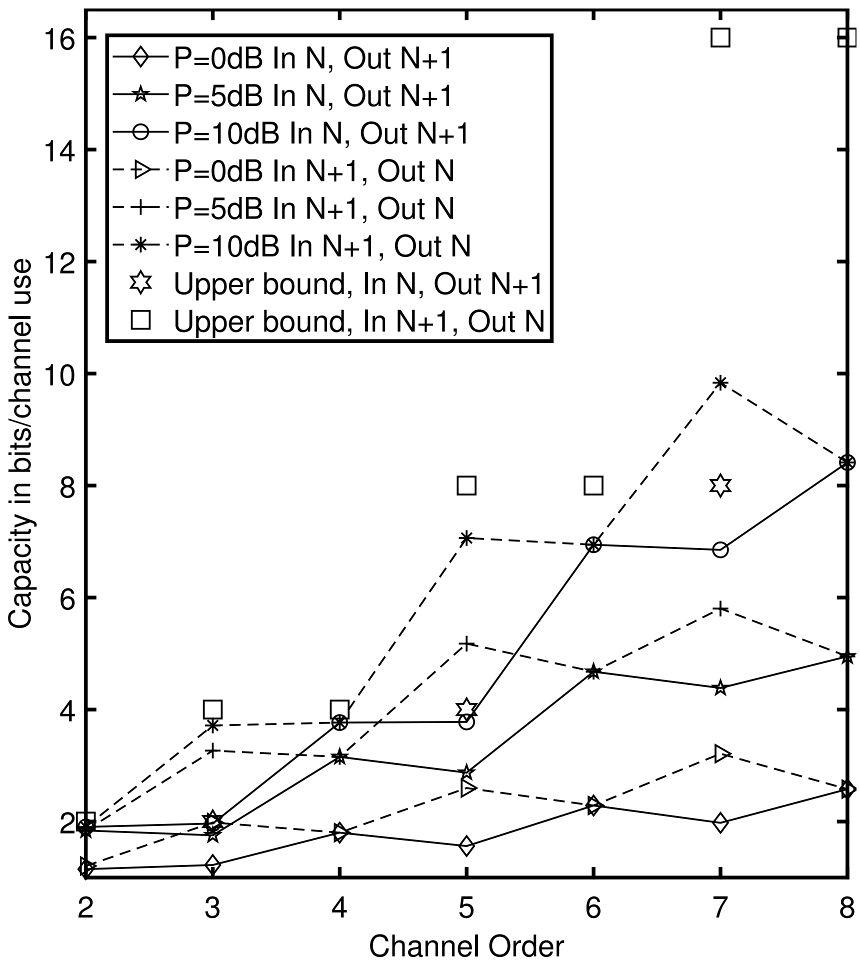


Fig. 5.17: Capacity vs Order of normalized tensor Gaussian channel for BPSK input.

Figure 5.17 presents the capacity against channel order for normalized channels. Similar to the previous example, we use square markers to denote the upper bound for case 1, and hexagram markers to denote the upper bound for case 2. Since for even order channels, the upper bound for case 1 and case 2 are identical, so for clarity of viewing, the upper bound for even order channels is marked only with a square. As discussed in section 3.3.2, for a fixed order normalized tensor channel, a change in the configuration of the input and output will change the strength of the channel components. A difference between the curves in Figures 5.17 and 5.16 is that in Figure 5.17 the capacity does not necessarily increase with channel order. The same observation was made in Figure 5.15 also. For instance, from channel order 5 to order 6, the capacity with case 1 decreases. This is because both order-5 (case 1) and order-6 channels correspond to the same input order 3, but normalization of these channels ensured weaker channel components in order-6 channel case. Further, the gap between any two power levels such as 5 dB and 10 dB is larger for normalized channels as compared to the non-normalized channels. This is so because for non-normalized channel, 5 dB and 10 dB power levels are high enough to reach closer to the saturated capacity values due to the additional power gain provided by the channel. This is evident in Figure 5.16 where the capacity is almost reaching the markers indicating the upper bound on capacity for high P. On the other hand, since normalized channels do not provide any power gain, much higher power is required to reach saturation. Thus the actual capacity is close to the upper bound markers in Figure 5.17 only for lower orders for which the upper bounds are low. The observations regarding the dependence of normalized channel capacity on the input configuration are consistent and similar to the Gaussian case (Figure 3.22) where depending on the specifics of the input and output order, capacity can decrease with increase in channel order due to channel normalization. A comparison between Figure 3.22 and Figure 5.17 shows that the capacity with BPSK input is always lower than the Gaussian case as expected. However, for lower values of P, the difference is small. Also on comparing Figure 3.23 and Figure 5.16 which both represent capacity for non-normalized channels, we see that the capacity increases with channel order for the input output configurations presented here. However, for the Gaussian input with non-normalized channel, the increase is smooth with increasing order, whereas for the BPSK input the increase is in a step fashion. This is because in case of non-normalized channels, input being order $N+1$ or output being order $N+1$ does not change the strength of the channel elements. So the capacity remains same for the two cases with Gaussian input since a Gaussian input does not restrict the capacity based on input size. On the contrary, with discrete inputs, the number of input elements imposes restriction on the capacity due to the discrete nature of input symbols.

It is to be noted that the results presented here for order 7 and 8 tensor channel cases are averaged over much smaller number of channel realizations. However, they still provide a good approximation of the average taken over larger number of samples. To establish this, in Table 5.2 and 5.3 we present the capacity calculated corresponding to 5 different channel realizations for the non-normalized and normalized cases respectively. For each power level, and different channel orders, the tables present the values of the capacity found for 5 specific realizations of random channels. Note that for odd order channels, the values in the table correspond to case 1 (input order $N+1$, output order N), since case 1 has higher complexity due to larger input size. It can be observed from the tables that the values of C obtained for different channel realizations are pretty close to each other in case of order 7 and 8 channels. To analyze the dispersion of the data points, we use the relative standard deviation and relative range as measures. For a set of 5 values of $C = \{C_1, C_2, C_3, C_4, C_5\}$ with average denoted by $C_{\texttt{avg}} = \frac{1}{5}\sum_{i=1}^{5} C_i$, the dispersion measures are defined as :

$$\text{Relative Standard Deviation} = \frac{\sqrt{\frac{1}{4}\sum_{i=1}^{5}(C_i - C_{\texttt{avg}})^2}}{C_{\texttt{avg}}} \tag{5.96}$$

$$\text{Relative Range} = \frac{\max\{C\} - \min\{C\}}{C_{\texttt{avg}}}. \tag{5.97}$$

Based on both the dispersion measures as indicated in the tables, it can be seen that for lower orders, such as order 3 and 4, the dispersion is much higher than for order 7 and 8 cases. This justifies our choice of Monte Carlo simulation parameters as listed in Table 5.1. Also, the relative deviation and range are much smaller for higher values of P, such as $P = 10$ dB especially in non-normalized tensor channels. This is because with channel also providing power gain, $P = 10$ dB is a sufficiently high transmit power for the capacity to saturate. Thus irrespective of the channel realization, the capacity bound from (5.95) is met and there is very little deviation in results from one realization to another. In general, for any power level, the individual data points are very similar for order 7 and 8 channel cases. Note that for order 7 and 8 tensor channels (when the input order is 4), iteratively updating the MMSE tensor and calculating the mutual information for a single channel realization itself can take upto a few weeks of simulation time. However since the number of elements in the tensor channel in such cases is very large (2^7 and 2^8 for order 7 and 8 respectively), due to channel hardening the variation across the results for different random realizations is small enough such that a general trend could be observed with averaging over only a smaller number of realizations.

Table 5.2: Capacity (C) for 5 realizations of tensor channels (non-normalized) with BPSK input.

P in dB	Channel Order	C in bits/channel-use	Relative Standard Deviation	Relative Range
0	3	3.8348, 2.1952, 2.1613, 1.8876, 2.0012	0.3323	0.8060
	4	3.4687, 3.7832, 3.4949, 2.8202, 2.8518	0.1301	0.2933
	5	5.5579, 5.2433, 5.6026, 5.7611, 6.1464	0.0582	0.1595
	6	7.2294, 6.8690, 6.7397, 7.3395, 7.2531	0.0373	0.0846
	7	11.2651, 11.4990, 11.3013, 10.8301, 11.1206	0.0222	0.0597
	8	13.8875, 13.8122, 14.1447, 14.3431, 13.9982	0.0151	0.0378
5	3	3.8262, 3.1292, 3.8835, 3.5796, 3.7686	0.0842	0.2074
	4	3.7407, 3.7007, 3.7160, 3.8265, 3.9890	0.0314	0.0760
	5	7.7475, 7.2652, 7.8428, 7.5489, 7.5900	0.0291	0.0760
	6	7.8748, 7.8539, 7.9775, 7.9043, 7.8227	0.0075	0.0196
	7	15.577, 15.3818, 15.5640, 15.3990, 15.6023	0.0068	0.0142
	8	15.6988, 15.7488, 15.8954, 15.8823, 15.6890	0.0063	0.0131
10	3	3.8997, 3.9981, 3.9368, 3.9156, 3.6862	0.0305	0.0802
	4	3.9849, 4.0000, 4.0000, 3.9111, 3.9922	0.0095	0.0223
	5	7.8122, 7.9726, 7.9757, 7.9615, 7.9701	0.0089	0.0206
	6	7.9851, 8.0000, 7.8991, 7.9982, 7.9888	0.0053	0.0127
	7	15.9650, 15.9965, 15.9254, 15.8991, 15.8673	0.0032	0.0081
	8	15.9987, 16.0000, 15.9803, 16.0000, 15.9877	0.0005	0.0012

Figures 5.18 and 5.19 present capacity with non-normalized and normalized channels respectively against channel order for QPSK inputs. Note that the simulation points for QPSK are included only till order 6 because generating points for higher order requires a

Table 5.3: Capacity (C) for 5 realizations of normalized tensor channels with BPSK input.

P in dB	Channel Order	C in bits/channel-use	Relative Standard Deviation	Relative Range
0	3	2.0256, 1.9544, 0.9360, 1.0099, 1.3445	0.3531	0.7493
	4	1.1880, 1.4366, 0.7594, 2.4211, 1.1001	0.4562	1.2032
	5	2.0980, 3.7683, 2.3716, 2.4565, 2.2611	0.2592	0.6446
	6	2.4002, 2.6929, 2.1583, 1.8743, 2.4795	0.1355	0.3527
	7	3.1376, 3.2705, 3.2278, 3.0109, 3.2059	0.0320	0.0819
	8	2.6683, 2.5415, 2.6202, 2.4869, 2.5690	0.0313	0.0704
5	3	2.0449, 2.2848, 2.9153, 3.5336, 2.4045	0.2252	0.5646
	4	2.4898, 2.5114, 2.8752, 3.1553, 3.3078	0.1289	0.2852
	5	4.2260, 5.0934, 5.3508, 5.4620, 3.9018	0.1458	0.3246
	6	4.7786, 4.8928, 4.0675, 3.9051, 4.2041	0.1008	0.2260
	7	5.7805, 5.7230, 5.9079, 6.5562, 5.9616	0.0556	0.1391
	8	4.9856, 4.9131, 4.9605, 5.2611, 4.9683	0.0276	0.0694
10	3	3.7695, 2.8141, 3.5459, 3.5820, 2.9086	0.1300	0.2874
	4	3.8896, 3.4402, 3.6065, 3.3335, 3.9488	0.0742	0.1689
	5	6.8043, 7.0262, 6.0578, 6.5212, 6.8863	0.0576	0.1454
	6	6.7376, 6.4160, 6.8139, 6.8052, 7.0859	0.0354	0.0989
	7	9.9630, 9.7091, 10.235, 10.0603, 9.8570	0.0200	0.0528
	8	8.4750, 8.3502, 8.2942, 8.3787, 8.4984	0.0102	0.0243

very long simulation time. However, a general trend could be observed using the results till order 6 and comparing it with the BPSK case. Similar trends as in Figure 5.16 and 5.17 can be observed for QPSK case, except that the QPSK case leads to higher capacity than BPSK for any fixed configuration. The saturation level of capacity with increasing P also changes because the constellation size is bigger. The capacity upper bound is marked using the square and hexagram markers. It can be seen that the actual capacity is close to this bound only for lower order cases in non-normalized channels at high power. Since the capacity saturation is twice as compared to BPSK, much higher power is required to reach the saturation with QPSK. Similar to BPSK, for QPSK also it can be seen that the difference between case 1 and case 2 for odd order channels gets bigger as P increases. The curves for lower P are closer to the BPSK and Gaussian case, and as P increases the QPSK curves fall between the BPSK and the Gaussian case. Especially in case of the

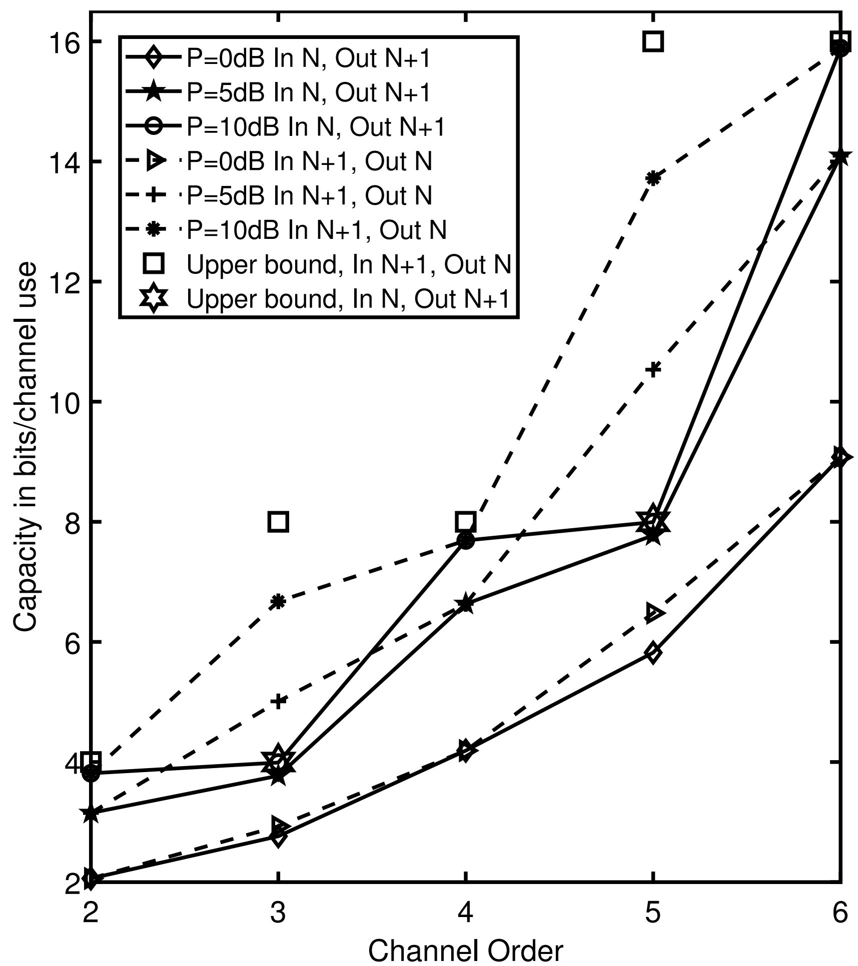


Fig. 5.18: Capacity vs Order of tensor Gaussian channel (non-normalized) for QPSK input.

normalized channels, since there is no power gain provided by the channel also, the curves corresponding to $P = 0$ dB are almost overlapping for BPSK, QPSK, and Gaussian cases. This is in line with our observation in the previous section where we showed that at very low transmit power budgets, the constellation constraints are not the dominant factor in determining the capacity. Furthermore, a comparison between BPSK and QPSK curves shows that at sufficiently high transmit power the capacity can increase almost by a factor of 2 when employing QPSK as compared to BPSK. This gain becomes more exact in cases where the capacity tends to reach its saturation due to high power. For instance, for an

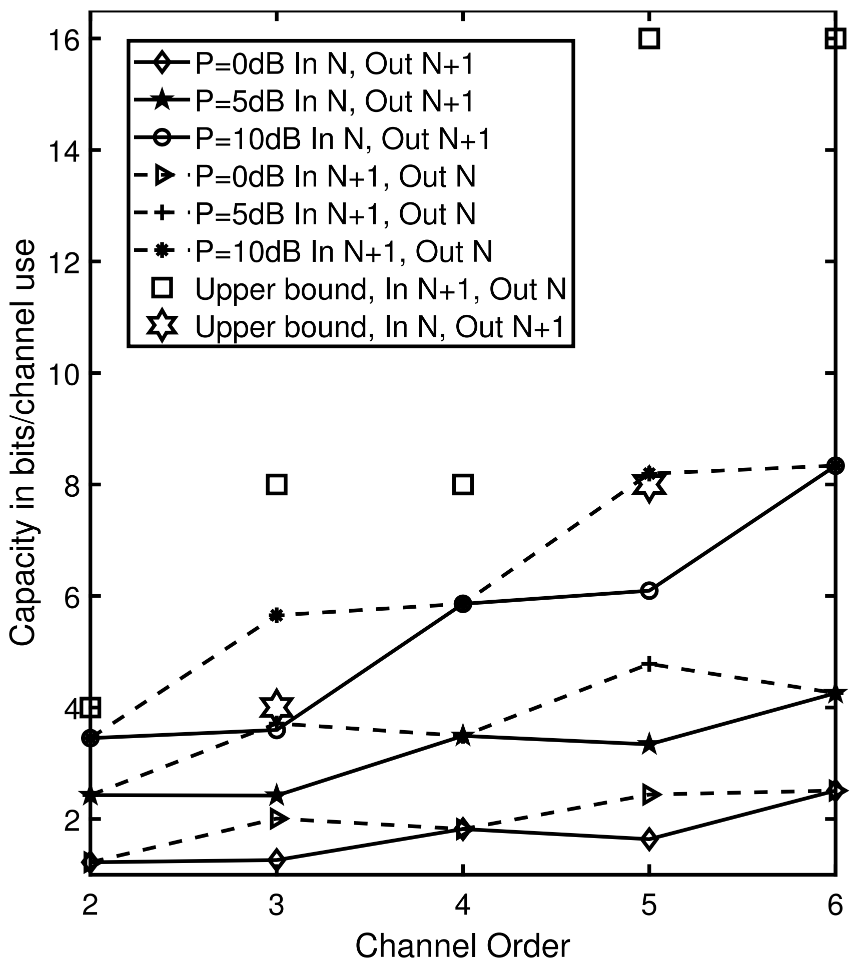


Fig. 5.19: Capacity vs Order of normalized tensor Gaussian channel for QPSK input.

order 4 channel, the capacity at $P = 10$ dB for non-normalized channels with BPSK input is around 4 bits/channel-use while with QPSK is close to 8 bits/channel-use. In case of normalized channels, since the channel is not providing any power gain, a transmit power of 10 dB is not sufficiently high enough for the QPSK input capacity to reach saturation. We can see that in case of normalized channels with QPSK input the capacity values are lower than the bound specified by (5.95). At high power, the gain in capacity by a factor of 2 achieved with QPSK when compared to BPSK arises since the constellation size Θ increases by a factor of 2, and at sufficiently high transmit power, the capacity bound specified by (5.95) is reached as per Lemma 8.

5.4.3 Capacity for different input distributions

In the previous sections, we considered examples where the input consists of independent symbols drawn from an equiprobable distribution as it is a widely used assumption for discrete constellations [105, 106, 108]. In this section, we consider the capacity under BPSK constellation constraint, where the BPSK symbols $\{+1, -1\}$ are not necessarily equally likely. Throughout this section, we assume that the elements in the input tensor, denoted by s, are independently drawn from $\{+1, -1\}$ using a Bernoulli distribution, where $Pr[s = +1] = p$ and $Pr[s = -1] = (1 - p)$. We analyze the capacity behavior with changing parameter p, and how much it deviates for smaller p as compared to $p = 0.5$ which corresponds to the equiprobable distribution. We consider examples of order 4 and order 6 tensor channels, and present results averaged over several channel realizations as listed in Table 5.1. The number of input and output domains is kept the same. Since here the objective is not to compare different order channels, but to compare effect of p for a given order channel, the tensor channels are not normalized and generated using circularly symmetric Gaussian distribution with unit variance.

Figure 5.20 shows the capacity of order 4 tensor channels vs p for three different levels of transmit power budgets P. As can be observed from Figure 5.20, the largest capacity is achieved when $p = 0.5$ for any transmit power. In particular at higher transmit powers, $p = 0.5$ almost reaches 4 bits/channel-use which is the upper bound on the capacity in this case. For other values of p also, it can be seen that at high enough transmit power P, the capacity tends to a value closer to the source entropy. For instance, corresponding to $p = 0.2$, the entropy of s where $Pr[s = +1] = p$ and $Pr[s = -1] = (1 - p)$ can be calculated as $-0.2 \log(0.2) - 0.8 \log(0.8) = 0.7219$ bits, and since the input contains 4 such independent elements, the source entropy can be written as $4 \times 0.7219 = 2.8877$ bits. In Figure 5.20 for $p = 0.2$, and $P = 10$ dB, we see that the capacity almost reaches this source entropy. This phenomenon can be further observed in Figure 5.21 where the order 4 tensor channel capacity is plotted against increasing transmit power for fixed values of p. For any given p, with sufficiently high transmit power, we see that the capacity reaches a saturation, where the saturation level can be specified by the source entropy. The saturation is highest

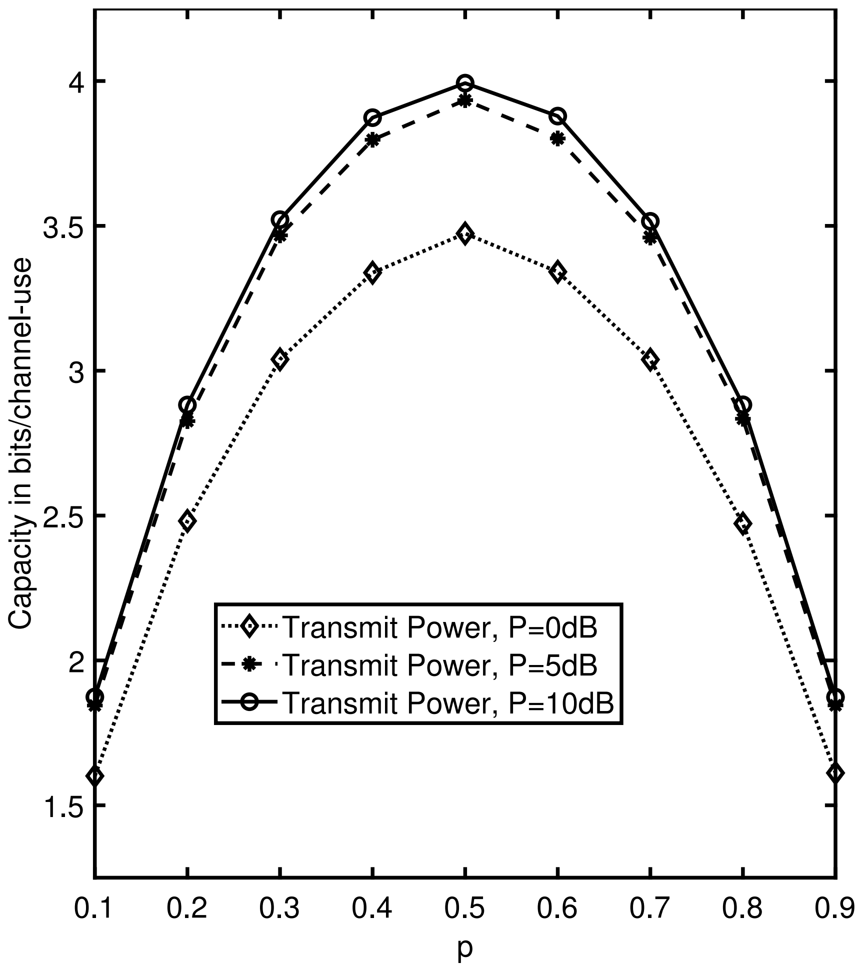


Fig. 5.20: Capacity vs p for order 4 channels with BPSK input for different transmit powers.

when $p = 0.5$, and decreases as p decreases. This suggests that among all the choices on input distribution for BPSK constellation, an equiprobable distribution provides the largest capacity.

Further, Figure 5.22 shows the capacity against p for an order 6 tensor channel with BPSK input. Similar to the order 4 case, here also we observe that capacity is largest when $p = 0.5$ for any transmit power. In this case, the capacity corresponding to $p = 0.5$ reaches a value of almost 8 bits/channel-use at high transmit power. As p deviates from 0.5, the capacity reduces significantly. For very small values of p such as $p = 0.1$, the drop in

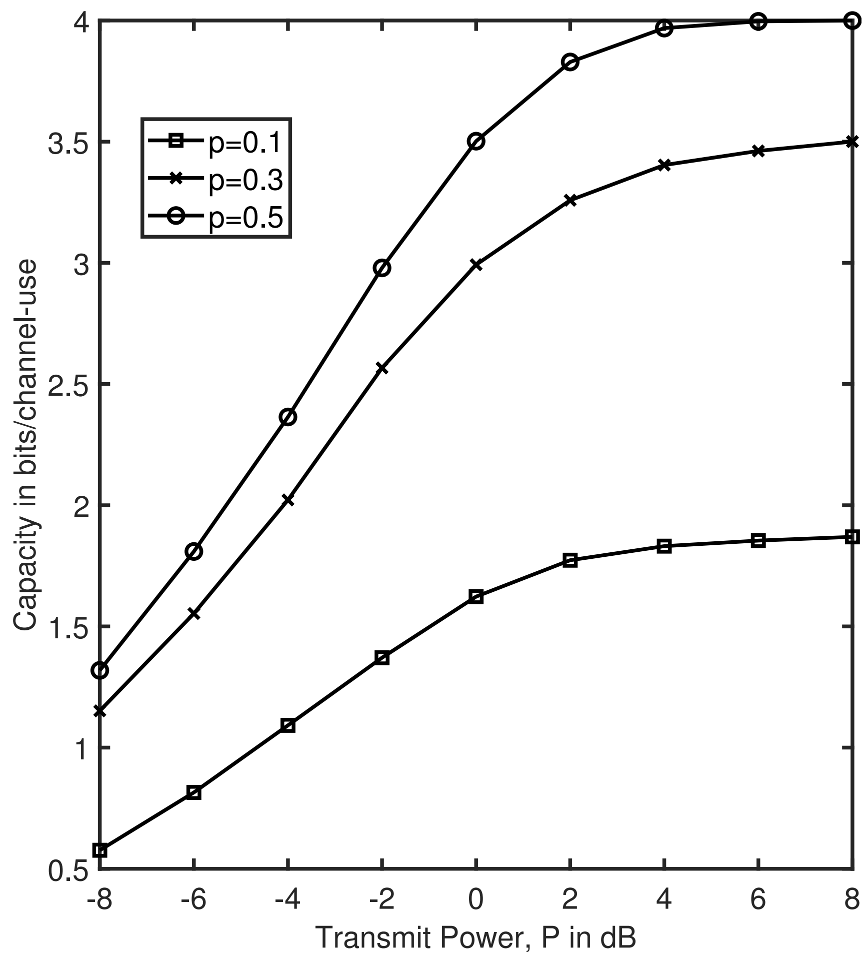


Fig. 5.21: Capacity vs Transmit power for order 4 channels with BPSK input for different p.

capacity as compared to $p = 0.5$ can be more than 50%. Here also, for any given p, at high transmit power the capacity approaches the source entropy. For instance, since the input contains 8 independent elements, at $p = 0.2$, the source entropy is $8 \times 0.7219 = 5.7744$ bits, and the capacity can be seen to be very close to this bound. It should be noted that the conclusion regarding equiprobable distribution providing the largest capacity is specific to the input constellation considered, which is BPSK for the presented example. A natural extension of the work in this chapter would be to consider maximizing the mutual information not for a given input distribution but over all the possible input distributions of the given constellation. This forms one of the future directions of this research.

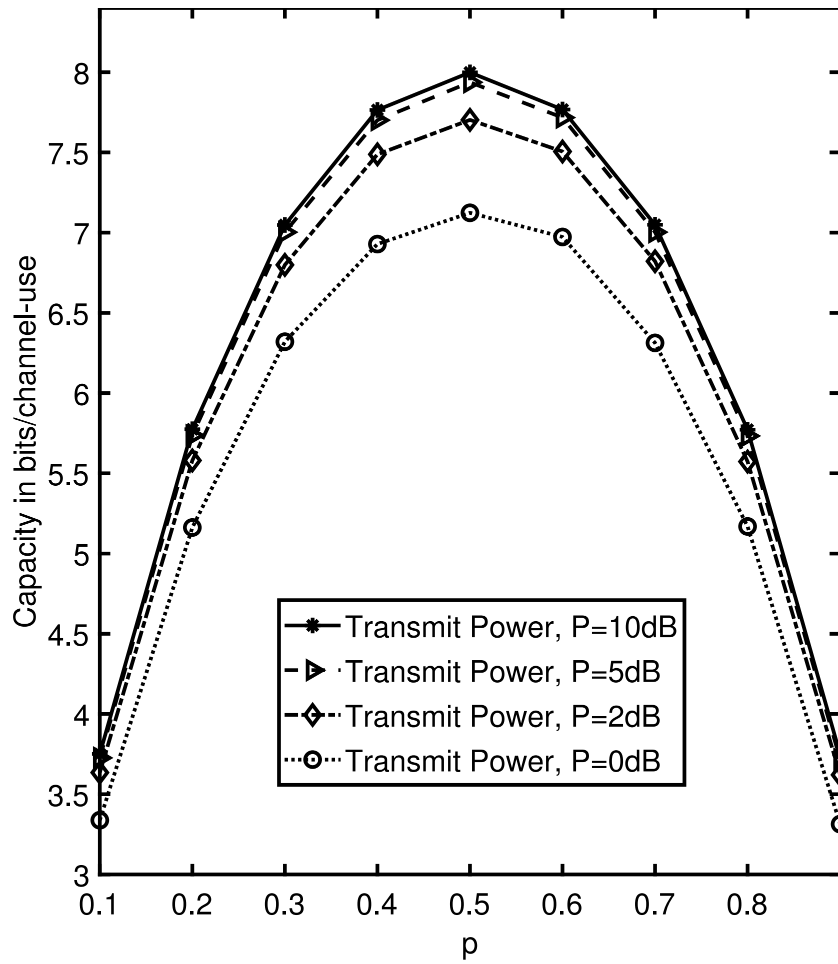


Fig. 5.22: Capacity vs p for order 6 channels with BPSK input for different transmit powers.

5.5 Chapter Summary

This chapter considered the problem of maximizing the mutual information in a multi-domain communication system with discrete inputs. We extended the relation between the derivative of the mutual information and the error covariance associated with the best MMSE estimator from a vector to a tensor setting in the presence of circularly symmetric Gaussian noise. The tensor I-MMSE relation was then used to find a precoder that achieves the tensor channel capacity when the input is drawn from discrete signalling constellations under a family of power constraints. It was shown that the mutual information saturates at high transmit power, where the saturation value is a function of the input tensor size, which increases exponentially with tensor order. Through numerical examples, it was concluded that at lower SNRs, the power constraints play a significant role while the signalling constellation constraint does not affect the capacity much. The capacity with discrete inputs is almost same as the capacity with Gaussian input at low enough SNRs. However, at higher SNRs, the capacity is significantly affected by the signalling constellation constraints, and thus any individual power constraints on input domains or elements does not significantly affect the capacity. Thus, at low SNRs, the power constraints are dominant, while at higher SNRs the signalling constellation constraints are dominant. For BPSK input, we showed through numerical examples that an equiprobable input distribution leads to maximum mutual information.

Chapter 6

Conclusions and Future Work

6.1 Summary and Conclusions

This thesis presented a unified mathematical framework using tensors to represent a multi-domain communication system, and demonstrated the benefits of such a framework through an information theoretic analysis of the tensor channel. A tensor-based system model was presented in Chapter 2 that can be used to model several modern communication systems by leveraging the multi-domain nature of the signals and systems involved.

The tensor framework's ability to retain the domain identifiability (interpretability) of the system model was exploited to represent a family of power constraints and find the Shannon capacity of the tensor channels under such constraints in Chapter 3. It was shown that the multiplexing gain provided by the tensor channel can increase exponentially with the number of domains. However, through several examples, it was demonstrated that the capacity increase with increasing channel order can vary based on channel normalization, and the specific input and output configurations considered for a fixed order tensor channel. Through numerical examples with tensor channels containing Gaussian i.i.d. elements, it was shown that having larger number of input domains provide more robustness in case of failed transmissions. Furthermore, the tensor framework highlights the domain trade-off phenomenon as was shown for Gaussian channels with i.i.d elements, and a MIMO GFDM system. The family of power constraints formulation also allowed to characterize the capacity of multi-user MIMO systems having per user power constraints for any number of users. In case of Gaussian MAC and IC channels, it was shown that using the tensor framework leads to a scheme where users cooperate for transmission and reception. Such user coordination can lead to higher achievable sum rates as compared to the sum rates achieved with independent users, especially as the number of users grows or the power of the interfering links increases.

Chapter 4 considered MMSE estimation for tensor based signals and established the notion of error covariance as a higher order tensor, which was later used in Chapter 5. The proposed framework in Chapter 4 dealt with estimation of complex tensors, proper or improper, using the Einstein product without reshaping the tensors, and thus could be employed for tensors in TT format as well. A comparison with the Tucker estimation was presented which showed that the Tucker approach while providing a low complexity solution leads to sub-optimal MSE results as compared to the proposed estimator. The application of the tensor estimator for a MIMO OFDM system with doubly selective channel was considered. It was shown that the tensor estimation outperforms per sub-carrier estimation for MIMO OFDM by a significant margin when the inter-carrier interference is high.

Further, Chapter 5 extended the well known vector I-MMSE relationship to a tensor setting by exploiting the representation of error covariance as a higher order tensor from Chapter 4. The tensor I-MMSE was used to find a precoder that achieves the tensor channel capacity when the input is drawn from discrete signalling constellations for a family of power constraints. Through numerical examples, it was concluded that at lower SNRs, the power constraints are dominant and the signalling constellation constraints do not have much impact on the capacity. However, at higher SNRs, the capacity is significantly affected by the signalling constellation constraints. The capacity under discrete signalling constellation constraints reaches a saturation with increasing transmit power. The saturation level is a function of the number of input domains and input distribution.

This thesis presented several examples of multi-domain systems where domains such as antenna, frequency, time, and users were considered and the channel was a tensor between order 3 to order 8. Going forward towards beyond 5G and 6G systems, as the nature of the transmission media evolves, identifying and incorporating additional domains in the system model would be a necessity. Systems with multiple domains are inherently complicated to deal with mathematically if one wishes to account for all the inter-domain interferences. The proposed tensor approach provides a convenient mathematical framework to model such complicated systems, derive their information transmission capabilities across all domains, and develop associated joint domain transmit and receive signal processing methods.

6.2 Directions for Future Research

In all the work presented in this thesis, it was assumed that the tensor channel is deterministic, and is thus known at both transmitter and receiver. One logical succession of this work is to analyze the cases when the channel is random, and the channel state information is either unknown or only partially known to the transmitter and receiver. In particular, it would be interesting to see how capacity of tensor channels behaves in correlated channels. A tensor channel has multiple modes, so the correlation can exist across several modes which may not always be separable as assumed in the Kronecker correlation model. Thus developing correlation model for tensor channels and analyzing their effect on channel capacity is a future work to consider.

Further, the proposed system model in this thesis is generic where we presented examples of systems such as MIMO OFDM and GFDM that fit the proposed model. However, it would be worthwhile to explore other systems where transmission spans across multiple domains that fit in the proposed framework. In particular, integration of future communication technologies such as IRS, mmWave, etc., into the tensor framework is of relevance. Also, tensor structures can be leveraged for improving both space and time complexities of several algorithms. Thus, one interesting future direction of the proposed research would be to mould and adapt the algorithms and techniques developed in this thesis to be implemented on parallel programming platforms. In particular, such computational benefits can be very useful in simulations related to Chapter 5 where the complexity of Monte Carlo simulations to find the MMSE tensor and mutual information increases exponentially with tensor size. The exponential dependency of the size of a tensor on its order is often referred as a curse of dimensionality, and several tensor decomposition techniques can be employed to work around it. An investigation of such techniques in the context of specific tensor channels forms another future direction of this research.

Lastly, while this work was focussed on the physical Layer, the tensor-based framework can be extended to include higher layers as well, providing a convenient basis for communication systems cross-layer design and signal processing. In particular the use of a tensor framework for multi-domain resource allocation can be of considerable interest.

Appendix A

KKT conditions

A.1 KKT conditions for Tensors

Consider scalar-valued functions of a tensor $\mathcal{Q}$ denoted by $f_i(\mathcal{Q})$. For an optimization problem of the form,

$$\begin{aligned} \min_{\mathcal{Q}} \quad & f_0(\mathcal{Q}) \\ s.t. \quad & f_i(\mathcal{Q}) \leq 0, \quad i = 1, \ldots, I \end{aligned} \tag{A.1}$$

the Lagrangian functional is defined as :

$$\mathcal{L}(\mathcal{Q}, \{\lambda_i\}) = f_0(\mathcal{Q}) + \sum_{i=1}^{I} \lambda_i f_i(\mathcal{Q}) \tag{A.2}$$

where λ_i is the Lagrange multiplier associated with the i^{th} inequality constraint. The inequality constraints in (A.1) along with the following conditions are known as the KKT conditions [195]

$$\nabla_{\mathcal{Q}} f_0(\mathcal{Q}) + \sum_{i=1}^{I} \lambda_i \nabla_{\mathcal{Q}} f_i(\mathcal{Q}) = 0_{\mathcal{T}} \quad \text{(i.e., } \nabla_{\mathcal{Q}} \mathcal{L} = 0_{\mathcal{T}}) \tag{A.3}$$

$$\lambda_i \geq 0 \quad i = 1, \ldots, I \tag{A.4}$$

$$\lambda_i f_i(\mathcal{Q}) = 0 \quad i = 1, \ldots, I. \tag{A.5}$$

For a convex optimization problem, the set of KKT conditions are both necessary and sufficient for the optimal solution [195, 150]. This is true irrespective of $\mathcal{Q}$ being a tensor or matrix or vector, as the functions f_0 and f_i's are scalar-valued and can be seen as functions of the individual components of $\mathcal{Q}$. For the complex case, the KKT conditions remains the same when we take derivative with respect to $\mathcal{Q}$ or $\mathcal{Q}^*$ [151]. In optimization theory jargon, (A.3) is referred as the stationarity condition and (A.5) is referred as the complementary slackness condition.

Now let us consider a more specific optimization problem where the variable is constrained to be a Hermitian positive semi-definite tensor, i.e., min $f_0(\mathcal{Q})$ such that $\mathcal{Q} \succeq 0$ where $\mathcal{Q} \in \mathbb{C}^{I_1\times\ldots\times I_N\times I_1\times\ldots\times I_N}$. The positive semi-definite constraint $\mathcal{Q} \succeq 0$, can be seen as a set of linear scalar constraints since all the eigenvalues of $\mathcal{Q}$ are non-negative. So $\mathcal{Q} \succeq 0$ is equivalent to the $I_1 I_2 \cdots I_N$ linear constraints $d_{i_1,\ldots,i_N} \geq 0$ or $-d_{i_1,\ldots,i_N} \leq 0$ for $i_1 = 1,\ldots,I_1,\ldots,i_N = 1,\ldots,I_N$ where $d_{i_1,\ldots,i_N}$ represents an eigenvalue of $\mathcal{Q}$. If the Lagrangian multiplier corresponding to each inequality constraint $-d_{i_1,\ldots,i_N} \leq 0$ is $\lambda_{i_1,\ldots,i_N}$ then the Lagrangian can be written as (take f_i to be $-d_{i_1,\ldots,i_N}$ in (A.2)),

$$\mathcal{L}(\mathcal{Q},\{\lambda_{i_1,\ldots,i_N}\}) = f_0(\mathcal{Q}) - \sum_{i_1,\ldots,i_N} \lambda_{i_1,\ldots,i_N} d_{i_1,\ldots,i_N}. \tag{A.6}$$

All $d_{i_1,\ldots,i_N}$ are the pseudo-diagonal entries of $\mathcal{D}$ from the tensor eigenvalue decomposition of $\mathcal{Q} = \mathcal{U} *_N \mathcal{D} *_N \mathcal{U}^H$. If we construct another Hermitian positive semi-definite tensor $\mathcal{M}$ of same size as $\mathcal{Q}$ with same unitary tensor as $\mathcal{U}$ and the pseudo-diagonal tensor whose elements are $\lambda_{i_1,\ldots,i_N}$, then we can see that

$$\sum_{i_1,\ldots,i_N} \lambda_{i_1,\ldots,i_N} d_{i_1,\ldots,i_N} = \mathrm{tr}(\mathcal{M} *_N \mathcal{Q}). \tag{A.7}$$

Thus for any Hermitian positive semi-definite constrained variable $\mathcal{Q}$, the Lagrangian from (A.6) can be written as

$$\mathcal{L} = f_0(\mathcal{Q}) - \mathrm{tr}(\mathcal{M} *_N \mathcal{Q}). \tag{A.8}$$

The complementary slackness condition corresponding to each of the linear constraint on the eigenvalues can be written as

$$\lambda_{i_1,\ldots,i_N} d_{i_1,\ldots,i_N} = 0, \quad \forall i_1,\ldots,i_N. \tag{A.9}$$

Since $\lambda_{i_1,\ldots,i_N}$ and $d_{i_1,\ldots,i_N}$ are non-negative, (A.9) can be substituted into (A.7) to write the complementary slackness as $\mathrm{tr}(\mathcal{M} *_N \mathcal{Q}) = 0$. This is similar to the approach taken for semi-definite programming for the matrix case [150]. Note that since $\mathcal{M}, \mathcal{Q} \succeq 0$, so $\mathrm{tr}(\mathcal{M} *_N \mathcal{Q}) = 0$ is equivalent to $\mathcal{M} *_N \mathcal{Q} = 0_{\mathcal{T}}$, as proven in the following lemma :

Lemma 9. *For Hermitian positive semi-definite tensors $\mathcal{Q}, \mathcal{M} \in \mathbb{C}^{I_1 \times \ldots \times I_N \times I_1 \times \ldots \times I_N}$, $\mathrm{tr}(\mathcal{M} *_N \mathcal{Q}) = 0$, is equivalent to $\mathcal{M} *_N \mathcal{Q} = 0_{\mathcal{T}}$.*

Proof. It is straightforward to see that if $\mathcal{M} *_N \mathcal{Q} = 0_{\mathcal{T}}$, then $\mathrm{tr}(\mathcal{M} *_N \mathcal{Q}) = \mathrm{tr}(0_{\mathcal{T}}) = 0$. Now we will prove in the other direction. Since $\mathcal{M}, \mathcal{Q}$ are Hermitian positive semi-definite tensors, we can write $\mathcal{M} = \mathcal{M}^{1/2H} *_N \mathcal{M}^{1/2}$, and $\mathcal{Q} = \mathcal{Q}^{1/2} *_N \mathcal{Q}^{1/2H}$. Thus we have :

$$\mathrm{tr}(\mathcal{M} *_N \mathcal{Q}) = \mathrm{tr}(\mathcal{M}^{1/2H} *_N \mathcal{M}^{1/2} *_N \mathcal{Q}^{1/2} *_N \mathcal{Q}^{1/2H}) = \mathrm{tr}(\mathcal{Q}^{1/2H} *_N \mathcal{M}^{1/2H} *_N \mathcal{M}^{1/2} *_N \mathcal{Q}^{1/2}). \tag{A.10}$$

Let $\mathcal{C} = \mathcal{M}^{1/2} *_N \mathcal{Q}^{1/2}$, then we have

$$\mathrm{tr}(\mathcal{M} *_N \mathcal{Q}) = \mathrm{tr}(\mathcal{C}^H *_N \mathcal{C}) = \sum_{i_1,\ldots,i_N,i'_1,\ldots,i'_N} |\mathcal{C}_{i_1,\ldots,i_N,i'_1,\ldots,i'_N}|^2. \tag{A.11}$$

Hence $\mathrm{tr}(\mathcal{M} *_N \mathcal{Q}) = 0$, implies every element $\mathcal{C}_{i_1,\ldots,i_N,i'_1,\ldots,i_N} = 0$. Thus

$$\mathrm{tr}(\mathcal{M} *_N \mathcal{Q}) = 0 \Rightarrow \mathcal{C} = 0_{\mathcal{T}} \Rightarrow \mathcal{M}^{1/2} *_N \mathcal{Q}^{1/2} = 0_{\mathcal{T}} \tag{A.12}$$

$$\Rightarrow \mathcal{M}^{1/2H} *_N \mathcal{M}^{1/2} *_N \mathcal{Q}^{1/2} *_N \mathcal{Q}^{1/2H} = 0_{\mathcal{T}} \tag{A.13}$$

$$\Rightarrow \mathcal{M} *_N \mathcal{Q} = 0_{\mathcal{T}}. \tag{A.14}$$

which proves the lemma. □

A.2 Solving the equations derived from KKT conditions for the optimal covariance tensor

In this Appendix, we present the solution to the equations (3.28) and (3.30) which are obtained through KKT conditions for finding the optimal transmit covariance tensor $\mathcal{Q}$. The results presented here are a generalization of Theorem 1 from [151] to a tensor setting. In this Appendix, $\mathcal{H}$ represents the channel tensor, $\mathcal{M}$ represents the tensor containing Lagrange multipliers corresponding to the semi-definite constraint on the covariance, and $\mathcal{B}$ represents the tensor whose pseudo-diagonal entries are the Lagrange multipliers corresponding to the other constraints on transmit covariance (such as power). For the sum power constraint, $\mathcal{B}$ is an identity tensor and the Lagrange multiplier is a scalar $\mu > 0$. So for $\mathcal{M} \succeq 0, \mathcal{B} \succ 0$, our objective is to find $\mathcal{Q} \succeq 0$, that satisfies (3.28) and (3.30). We re-write (3.28) and (3.30) in a more general form as (A.15) and (A.16) respectively:

$$\mathcal{H}^H *_M (\mathcal{H} *_N \mathcal{Q} *_N \mathcal{H}^H + \mathcal{I}_M)^{-1} *_M \mathcal{H} = \mu\mathcal{B} - \mathcal{M} \tag{A.15}$$

$$\mathcal{Q}^{1/2} *_N \mathcal{M} *_N \mathcal{Q}^{1/2} = 0_{\mathcal{T}} \tag{A.16}$$

where (A.15) is same as (3.28) for $\mu = 1$. We take the Einstein product of $\mathcal{Q}^{1/2}$ across N modes with (A.15) from both left and right side to write it as :

$$\begin{aligned}\mathcal{Q}^{1/2} *_N \mathcal{H}^H *_M (\mathcal{H} *_N \mathcal{Q} *_N \mathcal{H}^H + \mathcal{I}_M)^{-1} *_M \mathcal{H} *_N \mathcal{Q}^{1/2} \\ = \mu\mathcal{Q}^{1/2} *_N \mathcal{B}^{1/2} *_N \mathcal{B}^{1/2} *_N \mathcal{Q}^{1/2} - \underbrace{\mathcal{Q}^{1/2} *_N \mathcal{M} *_N \mathcal{Q}^{1/2}}_{=0_{\mathcal{T}} \text{ (from (A.16))}}.\end{aligned} \tag{A.17}$$

We define a tensor $\mathcal{A}$ as :

$$\mathcal{A} \triangleq \mathcal{B}^{1/2} *_N \mathcal{Q}^{1/2} \Rightarrow \mathcal{Q}^{1/2} = \mathcal{B}^{-1/2} *_N \mathcal{A}. \tag{A.18}$$

Since $\mathcal{Q}$ is Hermitian, we have $\mathcal{Q}^{1/2} = (\mathcal{Q}^{1/2})^H = \mathcal{A}^H *_N \mathcal{B}^{-1/2}$, which gives

$$\mathcal{Q} = \mathcal{B}^{-1/2} *_N \mathcal{A} *_N \mathcal{A}^H *_N \mathcal{B}^{-1/2}. \tag{A.19}$$

On substituting $\mathcal{Q}$ from (A.19) into (A.15), we get :

$$\mathcal{H}^H *_M \left(\mathcal{H} *_N \mathcal{B}^{-1/2} *_N \mathcal{A} *_N \mathcal{A}^H *_N \mathcal{B}^{-1/2} *_N \mathcal{H}^H + \mathcal{I}_M\right)^{-1} *_M \mathcal{H} = \mu\mathcal{B} - \mathcal{M}. \tag{A.20}$$

On taking the Einstein product of $\mathcal{B}^{-1/2}$ over N modes with (A.20) from both left and right side gives us

$$\begin{aligned}\underbrace{\mathcal{B}^{-1/2} *_N \mathcal{H}^H}_{\mathcal{K}^H} *_M \Big(\underbrace{\mathcal{H} *_N \mathcal{B}^{-1/2}}_{\mathcal{K}} *_N \mathcal{A} *_N \mathcal{A}^H *_N \underbrace{\mathcal{B}^{-1/2} *_N \mathcal{H}^H}_{\mathcal{K}^H} + \mathcal{I}_M\Big)^{-1} *_M \underbrace{\mathcal{H} *_N \mathcal{B}^{-1/2}}_{\mathcal{K}} \\ = \mu\mathcal{I}_N - \mathcal{B}^{-1/2} *_N \mathcal{M} *_N \mathcal{B}^{-1/2}\end{aligned} \tag{A.21}$$

where we define tensor $\mathcal{K}$ as $\mathcal{K} \triangleq \mathcal{H} *_N \mathcal{B}^{-1/2}$.

Proposition A.2.1. Let the tensor SVD of $\mathcal{A}$ and $\mathcal{K}$ be written as :

$$\mathcal{A} = \mathcal{U}_{\mathcal{A}} *_N \mathcal{D}_{\mathcal{A}} *_N \mathcal{V}_{\mathcal{A}}^H, \tag{A.22}$$

$$\mathcal{K} = \mathcal{U}_{\mathcal{K}} *_M \mathcal{D}_{\mathcal{K}} *_N \mathcal{V}_{\mathcal{K}}^H, \tag{A.23}$$

then $\mathcal{U}_{\mathcal{A}} = \mathcal{V}_{\mathcal{K}}$.

Proof. Substituting (A.18) and (A.19) into (A.17), we get

$$\underbrace{\mathcal{A}^H *_N \mathcal{B}^{-1/2} *_N \mathcal{H}^H}_{\mathcal{P}^H} *_M (\underbrace{\mathcal{H} *_N \mathcal{B}^{-1/2} *_N \mathcal{A}}_{\mathcal{P}} *_N \underbrace{\mathcal{A}^H *_N \mathcal{B}^{-1/2} *_N \mathcal{H}^H}_{\mathcal{P}^H} + \mathcal{I}_M)^{-1}$$
$$*_M \underbrace{\mathcal{H} *_N \mathcal{B}^{-1/2} *_N \mathcal{A}}_{\mathcal{P}} = \mu \mathcal{A}^H *_N \mathcal{A}. \quad \text{(A.24)}$$

We define tensor $\mathcal{P}$ and its tensor SVD as :

$$\mathcal{P} \triangleq \mathcal{H} *_N \mathcal{B}^{-1/2} *_N \mathcal{A} = \mathcal{U}_{\mathcal{P}} *_M \mathcal{D}_{\mathcal{P}} *_N \mathcal{V}_{\mathcal{P}}^H \quad \text{(A.25)}$$

On substituting (A.22) and (A.25) into (A.24), we get

$$\mathcal{V}_{\mathcal{P}} *_N \mathcal{D}_{\mathcal{P}}^H *_M \mathcal{U}_{\mathcal{P}}^H *_M (\mathcal{U}_{\mathcal{P}} *_M (\mathcal{D}_{\mathcal{P}} *_N \mathcal{D}_{\mathcal{P}}^H + \mathcal{I}_M) *_M \mathcal{U}_{\mathcal{P}}^H)^{-1} *_M \mathcal{U}_{\mathcal{P}} *_M \mathcal{D}_{\mathcal{P}} *_N \mathcal{V}_{\mathcal{P}}^H$$
$$= \mu(\mathcal{V}_{\mathcal{A}} *_N \mathcal{D}_{\mathcal{A}}^H *_N \mathcal{D}_{\mathcal{A}} *_N \mathcal{V}_{\mathcal{A}}^H) \quad \text{(A.26)}$$

$$\Rightarrow \mathcal{V}_{\mathcal{P}} *_N \overbrace{\mathcal{D}_{\mathcal{P}}^H *_M (\mathcal{D}_{\mathcal{P}} *_N \mathcal{D}_{\mathcal{P}}^H + \mathcal{I}_M)^{-1} *_M \mathcal{D}_{\mathcal{P}}}^{\text{pseudo-diagonal}} *_N \mathcal{V}_{\mathcal{P}}^H = (\mathcal{V}_{\mathcal{A}} *_N \overbrace{\mu \mathcal{D}_{\mathcal{A}}^H *_N \mathcal{D}_{\mathcal{A}}}^{\text{pseudo-diagonal}} *_N \mathcal{V}_{\mathcal{A}}^H). \quad \text{(A.27)}$$

Since $\mathcal{V}_{\mathcal{P}}$ and $\mathcal{V}_{\mathcal{A}}$ are unitary tensors, and the middle quantities on both sides of (A.27) are pseudo-diagonal, both the right and left side represent the tensor EVD of two equal tensors. From the uniqueness of tensor EVD, (A.27) implies

$$\mathcal{V}_{\mathcal{P}} = \mathcal{V}_{\mathcal{A}}. \quad \text{(A.28)}$$

Also,

$$\mathcal{P}^H *_M \mathcal{P} = \mathcal{A}^H *_N \underbrace{\mathcal{B}^{-1/2} *_N \mathcal{H}^H}_{\mathcal{K}^H} *_M \underbrace{\mathcal{H} *_N \mathcal{B}^{-1/2}}_{\mathcal{K}} *_N \mathcal{A} \quad \text{(A.29)}$$

$$= \mathcal{V}_{\mathcal{A}} *_N \underbrace{\mathcal{D}_{\mathcal{A}}^H *_N \mathcal{U}_{\mathcal{A}}^H *_N \mathcal{V}_{\mathcal{K}} *_N \mathcal{D}_{\mathcal{K}}^H *_M \mathcal{D}_{\mathcal{K}} *_N \mathcal{V}_{\mathcal{K}}^H *_N \mathcal{U}_{\mathcal{A}} *_N \mathcal{D}_{\mathcal{A}}}_{\text{middle term}} *_N \mathcal{V}_{\mathcal{A}}^H. \quad \text{(A.30)}$$

Since $\mathcal{V}_{\mathcal{A}} = \mathcal{V}_{\mathcal{P}}$ (from (A.28)) we see that (A.30) represents the tensor EVD of $\mathcal{P}^H *_M \mathcal{P}$. Hence the middle term in (A.30) is pseudo-diagonal. So $\mathcal{U}_{\mathcal{A}}^H *_N \mathcal{V}_{\mathcal{K}}$ must also be a pseudo-diagonal tensor. Let $\mathcal{S} \triangleq \mathcal{U}_{\mathcal{A}}^H *_N \mathcal{V}_{\mathcal{K}}$, then since $\mathcal{U}_{\mathcal{A}}$ and $\mathcal{V}_{\mathcal{K}}$ are unitary, we get

$$\mathcal{S}^H *_N \mathcal{S} = (\mathcal{U}_{\mathcal{A}}^H *_N \mathcal{V}_{\mathcal{K}})^H *_N (\mathcal{U}_{\mathcal{A}}^H *_N \mathcal{V}_{\mathcal{K}}) = (\mathcal{V}_{\mathcal{K}}^H *_N \underbrace{\mathcal{U}_{\mathcal{A}}) *_N (\mathcal{U}_{\mathcal{A}}^H}_{\mathcal{I}_N} *_N \mathcal{V}_{\mathcal{K}}) = \mathcal{I}_N. \quad \text{(A.31)}$$

Since $\mathcal{S}$ is pseudo-diagonal, (A.31) implies $\mathcal{S} = \mathcal{I}_N \Rightarrow \mathcal{U}_{\mathcal{A}} = \mathcal{V}_{\mathcal{K}}$, proving the proposition. □

From the tensor SVD of $\mathcal{A}$ and $\mathcal{K}$, and Proposition A.2.1, the left-hand side of (A.21) can be written as :

$$\begin{aligned}\mathcal{V}_{\mathcal{K}} *_N \mathcal{D}_{\mathcal{K}}^H *_M \mathcal{U}_{\mathcal{K}}^H *_M (\mathcal{U}_{\mathcal{K}} *_M \mathcal{D}_{\mathcal{K}} *_N \mathcal{D}_{\mathcal{A}} *_N \mathcal{D}_{\mathcal{A}}^H *_N \mathcal{D}_{\mathcal{K}}^H *_M \mathcal{U}_{\mathcal{K}}^H + \mathcal{I}_M)^{-1} *_M \mathcal{U}_{\mathcal{K}} *_M \mathcal{D}_{\mathcal{K}} *_N \mathcal{V}_{\mathcal{K}}^H \\ = \mathcal{V}_{\mathcal{K}} *_N \underbrace{\mathcal{D}_{\mathcal{K}}^H *_M (\mathcal{D}_{\mathcal{K}} *_N \mathcal{D}_{\mathcal{A}} *_N \mathcal{D}_{\mathcal{A}}^H *_N \mathcal{D}_{\mathcal{K}}^H + \mathcal{I}_M)^{-1} *_M \mathcal{D}_{\mathcal{K}}}_{\text{pseudo-diagonal}} *_N \mathcal{V}_{\mathcal{K}}^H.\end{aligned} \tag{A.32}$$

From the EVD, $\mathcal{B}^{-1/2} *_N \mathcal{M} *_N \mathcal{B}^{-1/2} = \mathcal{U} *_N \mathcal{D}_{\mathcal{BM}} *_N \mathcal{U}^H$, the right-hand side in (A.21) becomes

$$\mu \mathcal{I}_N - \mathcal{B}^{-1/2} *_N \mathcal{M} *_N \mathcal{B}^{-1/2} = \mathcal{U} *_N (\mu \mathcal{I}_N - \mathcal{D}_{\mathcal{BM}}) *_N \mathcal{U}^H. \tag{A.33}$$

Equations (A.32) and (A.33) represent the tensor EVD of the left and right hand side of (A.21), hence from uniqueness of tensor EVD we get $\mathcal{U} = \mathcal{V}_{\mathcal{K}}$ and :

$$\mathcal{D}_{\mathcal{K}}^H *_M (\mathcal{D}_{\mathcal{K}} *_N \mathcal{D}_{\mathcal{A}} *_N \mathcal{D}_{\mathcal{A}}^H *_N \mathcal{D}_{\mathcal{K}}^H + \mathcal{I}_M)^{-1} *_M \mathcal{D}_{\mathcal{K}} = \mu \mathcal{I}_N - \mathcal{D}_{\mathcal{BM}}. \tag{A.34}$$

Let the pseudo-diagonal elements of $\mathcal{D}_{\mathcal{A}}$, $\mathcal{D}_{\mathcal{K}}$ and $\mathcal{D}_{\mathcal{BM}}$ be $a_{i_1,\ldots,i_N}$, $k_{i_1,\ldots,i_N}$ and $m_{i_1,\ldots,i_N}$ respectively. Pseudo-diagonal elements of $\mathcal{D}_{\mathcal{A}}^H$ and $\mathcal{D}_{\mathcal{K}}^H$ will also be $a_{i_1,\ldots,i_N}$ and $k_{i_1,\ldots,i_N}$ respectively as these are real values. Since both sides of (A.34) are pseudo-diagonal, hence (A.34) can be written component-wise as :

$$\frac{k_{i_1,\ldots,i_N}^2}{1 + a_{i_1,\ldots,i_N}^2 k_{i_1,\ldots,i_N}^2} = \mu - m_{i_1,\ldots,i_N} \tag{A.35}$$

$$\Rightarrow a_{i_1,\ldots,i_N}^2 = \frac{1}{\mu - m_{i_1,\ldots,i_N}} - \frac{1}{k_{i_1,\ldots,i_N}^2}. \tag{A.36}$$

Note that (A.36) must always be non-negative since it represents the eigenvalues of a positive semi-definite tensor. To ensure that (A.36) is always non-negative, we will now show that $m_{i_1,\ldots,i_N} = 0$ for any non-zero $a_{i_1,\ldots,i_N}^2$. To see this, substitute (A.18) into (A.16) which gives $\mathcal{A}^H *_N \mathcal{B}^{-1/2} *_N \mathcal{M} *_N \mathcal{B}^{-1/2} *_N \mathcal{A} = 0_{\mathcal{T}}$. Using the tensor SVD of $\mathcal{A}$ and tensor EVD of $\mathcal{B}^{-1/2} *_N \mathcal{M} *_N \mathcal{B}^{-1/2}$ we can write :

$$\mathcal{V}_{\mathcal{A}} *_N \mathcal{D}_{\mathcal{A}}^H *_N \mathcal{D}_{\mathcal{BM}} *_N \mathcal{D}_{\mathcal{A}} *_N \mathcal{V}_{\mathcal{A}}^H = 0_{\mathcal{T}} \quad (\text{as } \mathcal{U} = \mathcal{V}_{\mathcal{K}} = \mathcal{U}_{\mathcal{A}}). \tag{A.37}$$

This implies that $\mathcal{D}_{\mathcal{A}}^H *_N \mathcal{D}_{\mathcal{BM}} *_N \mathcal{D}_{\mathcal{A}} = 0_{\mathcal{T}}$ which can be written element-wise as $a_{i_1,\ldots,i_N}^2 \cdot m_{i_1,\ldots,i_N} = 0$. Since $\mathcal{B} \succ 0$ and $\mathcal{M} \succeq 0$, we know $m_{i_1,\ldots,i_N} \geq 0$. So $a_{i_1,\ldots,i_N}^2 = 0$ when

$m_{i_1,\ldots,i_N} > 0$, otherwise it is given by (A.36) with $m_{i_1,\ldots,i_N} = 0$. Together it can be written as

$$a^2_{i_1,\ldots,i_N} = \left(\frac{1}{\mu} - \frac{1}{k^2_{i_1,\ldots,i_N}}\right)^+ \tag{A.38}$$

where $(z)^+ = \max\{0, z\}$. From (A.38) and Proposition A.2.1 we get

$$\mathcal{A} *_N \mathcal{A}^H = \mathcal{U}_{\mathcal{A}} *_N \left(\mu^{-1}\mathcal{I}_N - \bar{\mathcal{D}}^{-1}\right)^+ *_N \mathcal{U}^H_{\mathcal{A}} \tag{A.39}$$

$$= \mathcal{V}_{\mathcal{K}} *_N \left(\mu^{-1}\mathcal{I}_N - \bar{\mathcal{D}}^{-1}\right)^+ *_N \mathcal{V}^H_{\mathcal{K}} \tag{A.40}$$

where $\mathcal{V}_{\mathcal{K}}$ and $\bar{\mathcal{D}}$ are obtained from tensor EVD of $\mathcal{K}^H *_M \mathcal{K} = \mathcal{V}_{\mathcal{K}} *_N \bar{\mathcal{D}} *_N \mathcal{V}^H_{\mathcal{K}}$. Based on the tensor SVD of $\mathcal{K}$ in Proposition A.2.1, we have $\bar{\mathcal{D}} = \mathcal{D}^H_{\mathcal{K}} *_M \mathcal{D}_{\mathcal{K}}$. Substituting (A.40) into (A.19), we can conclude that

$$\mathcal{Q} = \mathcal{B}^{-1/2} *_N \mathcal{V}_{\mathcal{K}} *_N \left(\mu^{-1}\mathcal{I}_N - \bar{\mathcal{D}}^{-1}\right)^+ *_N \mathcal{V}^H_{\mathcal{K}} *_N \mathcal{B}^{-1/2}. \tag{A.41}$$

Appendix B

Proof of Theorems and some Miscellaneous Results

B.1 Proof of Theorem 1, Tensor SVD

For tensors $\mathcal{A} \in \mathbb{C}^{I_1\times\ldots\times I_N\times J_1\times\ldots\times J_M}$ and $\mathcal{B} \in \mathbb{C}^{J_1\times\ldots\times J_M\times K_1\times\ldots\times K_P}$, from (2.8), we get :

$$\mathcal{A} *_M \mathcal{B} = f^{-1}_{I_1,\ldots,I_N|K_1,\ldots,K_P}[f_{I_1,\ldots,I_N|J_1,\ldots,J_M}(\mathcal{A}) \cdot f_{J_1,\ldots,J_M|K_1,\ldots,K_P}(\mathcal{B})] \tag{B.1}$$

where $f^{-1}_{I_1,\ldots,I_N|K_1,\ldots,K_P}$ is the inverse matrix transform defined in (2.1). If $\mathrm{A} \in \mathbb{C}^{I_1I_2\cdots I_N\times J_1J_2\cdots J_M}$ and $\mathrm{B} \in \mathbb{C}^{J_1J_2\cdots J_M\times K_1K_2\cdots K_P}$ are transformed matrices from $\mathcal{A}$ and $\mathcal{B}$ respectively, then substituting $f_{I_1,\ldots,I_N|J_1,\ldots,J_M}(\mathcal{A}) = \mathrm{A}$ and $f_{J_1,\ldots,J_M|K_1,\ldots,K_P}(\mathcal{B}) = \mathrm{B}$ in (B.1) gives us

$$f^{-1}_{I_1,\ldots,I_N|K_1,\ldots,K_P}(\mathrm{A}\cdot\mathrm{B}) = \mathcal{A} *_M \mathcal{B} = f^{-1}_{I_1,\ldots,I_N|J_1,\ldots,J_M}(\mathrm{A}) *_M f^{-1}_{J_1,\ldots,J_M|K_1,\ldots,K_P}(\mathrm{B}). \tag{B.2}$$

Hence if $\mathrm{A} = \mathrm{U}\cdot\mathrm{D}\cdot\mathrm{V}^H$ (obtained from matrix SVD), then based on (B.2), for an order $N+M$ tensor $\mathcal{A} \in \mathbb{C}^{I_1\times\ldots\times I_N\times J_1\times\ldots\times J_M}$, we have :

$$\begin{aligned}\mathcal{A} &= f^{-1}_{I_1,\ldots,I_N|J_1,\ldots,J_M}(\mathrm{A}) = f^{-1}_{I_1,\ldots,I_N|J_1,\ldots,J_M}(\mathrm{U}\cdot\mathrm{D}\cdot\mathrm{V}^H)\\ &= f^{-1}_{I_1,\ldots,I_N|I_1,\ldots,I_N}(\mathrm{U}) *_N f^{-1}_{I_1,\ldots,I_N|J_1,\ldots,J_M}(\mathrm{D}) *_M f^{-1}_{J_1,\ldots,J_M|J_1,\ldots,J_M}(\mathrm{V}^H) = \mathcal{U} *_N \mathcal{D} *_M \mathcal{V}^H\end{aligned} \tag{B.3}$$

B.2 Concavity of log det

Lemma 10. *For a Hermitian positive semi-definite tensor* $\mathcal{Q} \in \mathbb{C}^{I_1 \times \ldots \times I_N \times I_1 \times \ldots \times I_N}$, $\log[\det(\mathcal{Q})]$ *is a concave function of* $\mathcal{Q}$.

Proof. The concavity of log det function for a matrix argument is proven in [197]. Let $\boldsymbol{\mathcal{X}}_1$ and $\boldsymbol{\mathcal{X}}_2$ be order N zero mean complex normal distributed tensors of size $I_1 \times \ldots \times I_N$, with covariance tensors $\mathcal{Q}_1$ and $\mathcal{Q}_2$ respectively. Let random variable $\boldsymbol{\theta}$ have distribution $P(\theta = 1) = \lambda, P(\theta = 2) = 1 - \lambda, 0 \leq \lambda \leq 1$. Let $\boldsymbol{\theta}, \boldsymbol{\mathcal{X}}_1$ and $\boldsymbol{\mathcal{X}}_2$ be independent and let $\boldsymbol{\mathcal{Z}}$ be defined as follows :

$$\boldsymbol{\mathcal{Z}} = \begin{cases} \boldsymbol{\mathcal{X}}_1, & \text{if } \boldsymbol{\theta} = 1, \\ \boldsymbol{\mathcal{X}}_2, & \text{if } \boldsymbol{\theta} = 2. \end{cases} \tag{B.4}$$

Covariance of $\boldsymbol{\mathcal{Z}}$ is given as $\lambda\mathcal{Q}_1 + (1-\lambda)\mathcal{Q}_2$. Note that $\boldsymbol{\mathcal{Z}}$ is Gaussian only if $\boldsymbol{\theta}$ is known, otherwise not. Now from Lemma 4, we know that :

$$\mathcal{H}(\boldsymbol{\mathcal{Z}}) \leq \log[(e\pi)^{I_1 \cdots I_N} \det(\lambda\mathcal{Q}_1 + (1-\lambda)\mathcal{Q}_2)]. \tag{B.5}$$

Since $\boldsymbol{\mathcal{Z}}$ given $\theta = 1$ is Gaussian with covariance $\mathcal{Q}_1$ and $\boldsymbol{\mathcal{Z}}$ given $\theta = 2$ is Gaussian with covariance $\mathcal{Q}_2$,we get :

$$\mathcal{H}(\boldsymbol{\mathcal{Z}}|\boldsymbol{\theta}) = P(\theta = 1) \cdot \mathcal{H}(\boldsymbol{\mathcal{Z}}|\theta = 1) + P(\theta = 2) \cdot \mathcal{H}(\boldsymbol{\mathcal{Z}}|\theta = 2) \tag{B.6}$$

$$= \lambda \log[(e\pi)^{I_1 \cdots I_N} \det(\mathcal{Q}_1)] + (1-\lambda) \log[(e\pi)^{I_1 \cdots I_N} \det(\mathcal{Q}_2)]. \tag{B.7}$$

We know that $\mathcal{H}(\boldsymbol{\mathcal{Z}}) \geq \mathcal{H}(\boldsymbol{\mathcal{Z}}|\boldsymbol{\theta})$, which implies

$$\log[(e\pi)^{I_1 \cdots I_N} \det(\lambda\mathcal{Q}_1 + (1-\lambda)\mathcal{Q}_2)] \geq \lambda \log[(e\pi)^{I_1 \cdots I_N} \det(\mathcal{Q}_1)] + (1-\lambda) \log[(e\pi)^{I_1 \cdots I_N} \det(\mathcal{Q}_2)]$$

$$\Rightarrow \log\det(\lambda\mathcal{Q}_1 + (1-\lambda)\mathcal{Q}_2) \geq \lambda \log\det(\mathcal{Q}_1) + (1-\lambda) \log\det(\mathcal{Q}_2) \tag{B.8}$$

which completes the proof. □

B.3 Proof of Theorem 3, the Orthogonality principle

$$\mathbb{E}[||\boldsymbol{\mathcal{X}} - h(\boldsymbol{\mathcal{Y}})||^2] = \mathbb{E}[||\underbrace{\boldsymbol{\mathcal{X}} - g(\boldsymbol{\mathcal{Y}})}_{\boldsymbol{\mathcal{E}}} + \underbrace{g(\boldsymbol{\mathcal{Y}}) - h(\boldsymbol{\mathcal{Y}})}_{\bar{h}(\boldsymbol{\mathcal{Y}})}||^2]$$

$$= \mathbb{E}[(\boldsymbol{\mathcal{E}} + \bar{h}(\boldsymbol{\mathcal{Y}})) *_N (\boldsymbol{\mathcal{E}} + \bar{h}(\boldsymbol{\mathcal{Y}}))^*]$$

$$= \mathbb{E}[\boldsymbol{\mathcal{E}} *_N \boldsymbol{\mathcal{E}^*} + \boldsymbol{\mathcal{E}} *_N \bar{h}(\boldsymbol{\mathcal{Y}})^* + \bar{h}(\boldsymbol{\mathcal{Y}}) *_N \boldsymbol{\mathcal{E}}^* + \bar{h}(\boldsymbol{\mathcal{Y}}) *_N \bar{h}(\boldsymbol{\mathcal{Y}})^*]$$

$$= \mathbb{E}[||\boldsymbol{\mathcal{E}}||^2] + \underbrace{\mathbb{E}[\boldsymbol{\mathcal{E}} *_N \bar{h}(\boldsymbol{\mathcal{Y}})^*] + \mathbb{E}[\bar{h}(\boldsymbol{\mathcal{Y}}) *_N \boldsymbol{\mathcal{E}}^*]}_{\text{cross-terms}} + \mathbb{E}[||\bar{h}(\boldsymbol{\mathcal{Y}})||^2].$$

Since $\bar{h}(\boldsymbol{\mathcal{Y}})$ is another function of $\boldsymbol{\mathcal{Y}}$, hence if (4.1) holds, then the cross terms above would be zero, which results into :

$$\mathbb{E}[||\boldsymbol{\mathcal{X}} - h(\boldsymbol{\mathcal{Y}})||^2] = \mathbb{E}[||\boldsymbol{\mathcal{E}}||^2] + \underbrace{\mathbb{E}[||\bar{h}(\boldsymbol{\mathcal{Y}})||^2]}_{\geq 0} \geq \mathbb{E}[||\boldsymbol{\mathcal{E}}||^2].$$

B.4 Derivation of Error Covariance Tensor from (4.32)

$$\begin{aligned} \mathcal{Q}_{WL} &= \mathbb{E}[(\boldsymbol{\mathcal{X}} - \hat{\boldsymbol{\mathcal{X}}}_{WL}) \circ (\boldsymbol{\mathcal{X}} - \hat{\boldsymbol{\mathcal{X}}}_{WL})^*] \\ &= \mathbb{E}[\boldsymbol{\mathcal{X}} \circ \boldsymbol{\mathcal{X}}^*] - \mathbb{E}[\boldsymbol{\mathcal{X}} \circ \hat{\boldsymbol{\mathcal{X}}}^*_{WL}] - \mathbb{E}[\hat{\boldsymbol{\mathcal{X}}}_{WL} \circ \boldsymbol{\mathcal{X}}^*] + \mathbb{E}[\hat{\boldsymbol{\mathcal{X}}}_{WL} \circ \hat{\boldsymbol{\mathcal{X}}}^*_{WL}] \end{aligned} \tag{B.9}$$

where $\hat{\boldsymbol{\mathcal{X}}}_{WL} = \mathcal{A}_1 *_M \boldsymbol{\mathcal{Y}} + \mathcal{A}_2 *_M \boldsymbol{\mathcal{Y}}^*$ (from (4.17)). Since the Einstein product in $\mathcal{A}_1 *_M \boldsymbol{\mathcal{Y}}$ and $\mathcal{A}_2 *_M \boldsymbol{\mathcal{Y}}^*$ are over all the M modes of the tensor $\mathcal{Y}$ (which is an order M tensor), thus the commutativity rule from (2.10) can be used to write $(\mathcal{A}_1 *_M \boldsymbol{\mathcal{Y}})^* = (\boldsymbol{\mathcal{Y}}^* *_M \mathcal{A}_1^H)$ and $(\mathcal{A}_2 *_M \boldsymbol{\mathcal{Y}}^*)^* = (\boldsymbol{\mathcal{Y}} *_M \mathcal{A}_2^H)$. So individual terms in (B.9) can be simplified as :

$$\begin{aligned} \mathbb{E}[\boldsymbol{\mathcal{X}} \circ \hat{\boldsymbol{\mathcal{X}}}^*_{WL}] &= \mathbb{E}[\boldsymbol{\mathcal{X}} \circ (\mathcal{A}_1^* *_M \boldsymbol{\mathcal{Y}}^* + \mathcal{A}_2^* *_M \boldsymbol{\mathcal{Y}})] \\ &= \mathbb{E}[(\boldsymbol{\mathcal{X}} \circ \boldsymbol{\mathcal{Y}}^*) *_M \mathcal{A}_1^H + (\boldsymbol{\mathcal{X}} \circ \boldsymbol{\mathcal{Y}}) *_M \mathcal{A}_2^H] \\ &= \mathcal{C}_{\boldsymbol{\mathcal{X}}\boldsymbol{\mathcal{Y}}} *_M \mathcal{A}_1^H + \tilde{\mathcal{C}}_{\boldsymbol{\mathcal{X}}\boldsymbol{\mathcal{Y}}} *_M \mathcal{A}_2^H, \end{aligned} \tag{B.10}$$

$$\begin{aligned} \mathbb{E}[\hat{\boldsymbol{\mathcal{X}}}_{WL} \circ \boldsymbol{\mathcal{X}}^*] &= \mathbb{E}[(\mathcal{A}_1 *_M \boldsymbol{\mathcal{Y}} + \mathcal{A}_2 *_M \boldsymbol{\mathcal{Y}}^*) \circ \boldsymbol{\mathcal{X}}^*] \\ &= \mathcal{A}_1 *_M \mathbb{E}[\boldsymbol{\mathcal{Y}} \circ \boldsymbol{\mathcal{X}}^*] + \mathcal{A}_2 *_M \mathbb{E}[\boldsymbol{\mathcal{Y}}^* \circ \boldsymbol{\mathcal{X}}^*] \\ &= \mathcal{A}_1 *_M \mathcal{C}^H_{\boldsymbol{\mathcal{X}}\boldsymbol{\mathcal{Y}}} + \mathcal{A}_2 *_M \tilde{\mathcal{C}}^H_{\boldsymbol{\mathcal{X}}\boldsymbol{\mathcal{Y}}}, \end{aligned} \tag{B.11}$$

$$
\begin{aligned}
&\mathbb{E}[\hat{\mathbf{\mathfrak{X}}}_{WL} \circ \hat{\mathbf{\mathfrak{X}}}^*_{WL}] \\
&= \mathbb{E}[(\mathcal{A}_1 *_M \mathbf{\mathfrak{Y}} + \mathcal{A}_2 *_M \mathbf{\mathfrak{Y}}^*) \circ (\mathcal{A}_1^* *_M \mathbf{\mathfrak{Y}}^* + \mathcal{A}_2^* *_M \mathbf{\mathfrak{Y}})] \\
&= \mathbb{E}[(\mathcal{A}_1 *_M \mathbf{\mathfrak{Y}} + \mathcal{A}_2 *_M \mathbf{\mathfrak{Y}}^*) \circ (\mathbf{\mathfrak{Y}}^* *_M \mathcal{A}_1^H + \mathbf{\mathfrak{Y}} *_M \mathcal{A}_2^H)] \\
&= \mathcal{A}_1 *_M \mathcal{C}_{\mathbf{\mathfrak{Y}}} *_M \mathcal{A}_1^H + \mathcal{A}_2 *_M \tilde{\mathcal{C}}^*_{\mathbf{\mathfrak{Y}}} *_M \mathcal{A}_1^H + \mathcal{A}_1 *_M \tilde{\mathcal{C}}_{\mathbf{\mathfrak{Y}}} *_M \mathcal{A}_2^H + \mathcal{A}_2 *_M \mathcal{C}^*_{\mathbf{\mathfrak{Y}}} *_M \mathcal{A}_2^H \\
&= (\mathcal{A}_1 *_M \mathcal{C}_{\mathbf{\mathfrak{Y}}} + \mathcal{A}_2 *_M \tilde{\mathcal{C}}^*_{\mathbf{\mathfrak{Y}}}) *_M \mathcal{A}_1^H + (\mathcal{A}_1 *_M \tilde{\mathcal{C}}_{\mathbf{\mathfrak{Y}}} + \mathcal{A}_2 *_M \mathcal{C}^*_{\mathbf{\mathfrak{Y}}}) *_M \mathcal{A}_2^H \\
&= \mathcal{C}_{\mathbf{\mathfrak{X}}\mathbf{\mathfrak{Y}}} *_M \mathcal{A}_1^H + \tilde{\mathcal{C}}_{\mathbf{\mathfrak{X}}\mathbf{\mathfrak{Y}}} *_M \mathcal{A}_2^H \quad \text{(from (4.25) and (4.26)).}
\end{aligned}
\tag{B.12}
$$

Substituting $\mathbb{E}[\mathbf{\mathfrak{X}} \circ \mathbf{\mathfrak{X}}^*] = \mathcal{C}_{\mathbf{\mathfrak{X}}}$ along with (B.10), (B.11) and (B.12) into (B.9), we get :

$$
\mathcal{Q}_{WL} = \mathcal{C}_{\mathbf{\mathfrak{X}}} - \mathcal{A}_1 *_M \mathcal{C}^H_{\mathbf{\mathfrak{X}}\mathbf{\mathfrak{Y}}} - \mathcal{A}_2 *_M \tilde{\mathcal{C}}^H_{\mathbf{\mathfrak{X}}\mathbf{\mathfrak{Y}}}. \tag{B.13}
$$

B.5 Proof of Theorem 4

To prove Theorem 4, we show that given $\mathbf{\mathfrak{Y}} = \mathcal{Y}$, $\mathbf{\mathfrak{X}}$ is conditionally $\mathcal{CN}(\mathcal{A}_1 *_M \mathcal{Y} + \mathcal{A}_2 *_M \mathcal{Y}^*, \mathcal{Q}_{WL}, \tilde{\mathcal{Q}}_{WL})$ using the characteristic function.

The *characteristic function* of a complex random vector $\underline{\mathbf{x}} \in \mathbb{C}^N$ is defined as $\Phi_{\underline{\mathbf{x}}}(\underline{\omega}) = \mathbb{E}[\exp(i\Re(\underline{\omega}^H \underline{\mathbf{x}}))]$ for $\underline{\omega} \in \mathbb{C}^N$[198]. Using Einstein Product, the characteristic function of a complex random tensor $\mathbf{\mathfrak{X}} \in \mathbb{C}^{I_1 \times \ldots \times I_N}$ is

$$
\Phi_{\mathbf{\mathfrak{X}}}(\mathcal{W}) = \mathbb{E}[\exp(i\Re(\mathcal{W}^* *_N \mathbf{\mathfrak{X}}))] \tag{B.14}
$$

for tensor $\mathcal{W} \in \mathbb{C}^{I_1 \times \ldots \times I_N}$. Notice that $\mathbf{z} = \mathcal{W}^* *_N \mathbf{\mathfrak{X}}$ is a complex scalar random variable. If $\mathbf{\mathfrak{X}}$ is a complex Gaussian tensor with mean $\mathcal{M}$, covariance $\mathcal{C}$ and pseudo-covariance $\tilde{\mathcal{C}}$, then $\mathbf{z}$ will be Gaussian distributed with mean

$$
\mu_{\mathbf{z}} = \mathbb{E}[\mathbf{z}] = \mathcal{W}^* *_N \mathbb{E}[\mathbf{\mathfrak{X}}] = \mathcal{W}^* *_N \mathcal{M}, \tag{B.15}
$$

variance $\sigma_{\mathbf{z}}^2$ and pseudo-variance $\tilde{\sigma}_{\mathbf{z}}^2$ found using properties (2.9) and (2.18)

$$\begin{aligned}
\sigma_{\mathbf{z}}^2 &= \mathbb{E}[(\mathcal{W}^* *_N (\boldsymbol{\mathfrak{X}} - \mathcal{M})) \circ (\mathcal{W}^* *_N (\boldsymbol{\mathfrak{X}} - \mathcal{M}))^*] \\
&= \mathcal{W}^* *_N \mathbb{E}[(\boldsymbol{\mathfrak{X}} - \mathcal{M}) \circ ((\boldsymbol{\mathfrak{X}} - \mathcal{M})^*] *_N \mathcal{W} \\
&= \mathcal{W}^* *_N \mathcal{C} *_N \mathcal{W} \qquad (B.16) \\
\tilde{\sigma}_{\mathbf{z}}^2 &= \mathbb{E}[(\mathcal{W}^* *_N (\boldsymbol{\mathfrak{X}} - \mathcal{M})) \circ (\mathcal{W}^* *_N (\boldsymbol{\mathfrak{X}} - \mathcal{M}))] \\
&= \mathcal{W}^* *_N \tilde{\mathcal{C}} *_N \mathcal{W}^*. \qquad (B.17)
\end{aligned}$$

The characteristic function $\Phi_{\mathbf{z}}(\omega)$ of a Gaussian scalar $\mathbf{z} \sim \mathcal{CN}(\mu_{\mathbf{z}}, \sigma_{\mathbf{z}}^2, \tilde{\sigma}_{\mathbf{z}}^2)$ is given by [199]:

$$\Phi_{\mathbf{z}}(\omega) = \exp\left\{ i\Re(\omega^* \mu_{\mathbf{z}}) - \frac{1}{4}\Big(\omega^* \sigma_{\mathbf{z}}^2 \omega + \Re(\omega^* \tilde{\sigma}_{\mathbf{z}}^2 \omega^*)\Big)\right\}. \qquad (B.18)$$

Now on putting $\mathbf{z} = \mathcal{W}^* *_N \boldsymbol{\mathfrak{X}}$ in (B.14) we get :

$$\Phi_{\boldsymbol{\mathfrak{X}}}(\mathcal{W}) = \mathbb{E}[\exp(i\Re(\omega^* \cdot \mathbf{z}))]\Big|_{\omega=1} = \Phi_{\mathbf{z}}(\omega)\Big|_{\omega=1}. \qquad (B.19)$$

On substituting (B.15), (B.16), (B.17) and $\omega = 1$ in (B.18),

$$\Phi_{\boldsymbol{\mathfrak{X}}}(\mathcal{W}) = \exp\left\{ i\Re(\mathcal{W}^* *_N \mathcal{M}) - \frac{1}{4}\Big(\mathcal{W}^* *_N \mathcal{C} *_N \mathcal{W} + \Re(\mathcal{W}^* *_N \tilde{\mathcal{C}} *_N \mathcal{W}^*)\Big)\right\}. \qquad (B.20)$$

The characteristic function of an improper complex Gaussian vector as given in [199, 200] can be seen as a specific case of (B.20). Further, the characteristic function of $\boldsymbol{\mathfrak{X}}$ given $\boldsymbol{\mathfrak{Y}} = \mathcal{Y}$ can be written as (from (B.14)) :

$$\begin{aligned}
\Phi_{\boldsymbol{\mathfrak{X}}|\boldsymbol{\mathfrak{Y}}}(\mathcal{W}) &= \mathbb{E}\Big[\exp\Big(i\Re(\mathcal{W}^* *_N \boldsymbol{\mathfrak{X}})\Big) \mid \boldsymbol{\mathfrak{Y}} = \mathcal{Y}\Big] \\
&= \mathbb{E}\Big[\exp\Big(i\Re(\mathcal{W}^* *_N (\boldsymbol{\mathfrak{X}} - \hat{\boldsymbol{\mathfrak{X}}}_{WL} + \hat{\boldsymbol{\mathfrak{X}}}_{WL}))\Big) \mid \boldsymbol{\mathfrak{Y}} = \mathcal{Y}\Big] \\
&= \mathbb{E}\Big[\underbrace{\exp(i\Re(\mathcal{W}^* *_N (\boldsymbol{\mathfrak{X}} - \hat{\boldsymbol{\mathfrak{X}}}_{WL})))}_{a}\underbrace{\exp\Big(i\Re(\mathcal{W}^* *_N \hat{\boldsymbol{\mathfrak{X}}}_{WL})\Big)}_{b} \mid \boldsymbol{\mathfrak{Y}} = \mathcal{Y}\Big].
\end{aligned}$$

Term b can be taken out of the expectation as given $\boldsymbol{\mathfrak{Y}} = \mathcal{Y}$, we also know $\mathcal{Y}^*$, so $\hat{\boldsymbol{\mathfrak{X}}}_{WL}$ which is given as $\mathcal{A}_1 *_M \boldsymbol{\mathfrak{Y}} + \mathcal{A}_2 *_M \boldsymbol{\mathfrak{Y}}^*$ becomes deterministic. In term a, the vector $(\boldsymbol{\mathfrak{X}} - \hat{\boldsymbol{\mathfrak{X}}}_{WL})$ is the error tensor $\boldsymbol{\mathcal{E}}$, hence we get :

$$\Phi_{\boldsymbol{\mathfrak{X}}|\boldsymbol{\mathfrak{Y}}}(\mathcal{W}) = \exp\Big(i\Re(\mathcal{W}^* *_N (\mathcal{A}_1 *_M \mathcal{Y} + \mathcal{A}_2 *_M \mathcal{Y}^*))\Big)\mathbb{E}\Big[\exp\Big(i\Re(\mathcal{W}^* *_N \boldsymbol{\mathcal{E}})\Big) \mid \boldsymbol{\mathfrak{Y}} = \mathcal{Y}\Big]. \qquad (B.21)$$

Since the error tensor defined as $\boldsymbol{\mathcal{E}} = \boldsymbol{\mathfrak{X}} - (\mathcal{A}_1 *_M \boldsymbol{\mathfrak{Y}} + \mathcal{A}_2 *_M \boldsymbol{\mathfrak{Y}}^*)$ is orthogonal to $\boldsymbol{\mathfrak{Y}}$, and also $\boldsymbol{\mathfrak{X}}$ and $\boldsymbol{\mathfrak{Y}}$ are jointly complex normal, so $\boldsymbol{\mathcal{E}}$ is independent of $\boldsymbol{\mathfrak{Y}}$. Thus we can drop the conditioning in (B.21), which gives

$$\Phi_{\boldsymbol{\mathcal{X}}|\boldsymbol{\mathcal{Y}}}(\mathcal{W}) \;=\; \exp\Big(i\Re(\mathcal{W}^* *_N (\mathcal{A}_1 *_M \mathcal{Y} + \mathcal{A}_2 *_M \mathcal{Y}^*))\Big)\underbrace{\mathbb{E}\Big[\exp\Big(i\Re(\mathcal{W}^* *_N \boldsymbol{\mathcal{E}})\Big)\Big]}_{\Phi_{\boldsymbol{\mathcal{E}}}(\mathcal{W})}. \quad \text{(B.22)}$$

Since $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{Y}}$ are assumed zero mean and jointly Gaussian, so the error tensor $\boldsymbol{\mathcal{E}}$ will also be Gaussian with zero mean. Hence its characteristic function is given as (from (B.20)):

$$\Phi_{\boldsymbol{\mathcal{E}}}(\mathcal{W}) = \exp\bigg\{-\frac{1}{4}\Big(\mathcal{W}^* *_N \mathcal{C}_{\boldsymbol{\mathcal{E}}} *_N \mathcal{W} + \Re(\mathcal{W}^* *_N \tilde{\mathcal{C}}_{\boldsymbol{\mathcal{E}}} *_N \mathcal{W}^*)\Big)\bigg\}. \quad \text{(B.23)}$$

Substituting (B.23) into (B.22) with the error covariance $\mathcal{C}_{\boldsymbol{\mathcal{E}}} = \mathcal{Q}_{WL}$ and pseudo-covariance $\tilde{\mathcal{C}}_{\boldsymbol{\mathcal{E}}} = \tilde{\mathcal{Q}}_{WL}$, we get :

$$\begin{aligned}\Phi_{\boldsymbol{\mathcal{X}}|\boldsymbol{\mathcal{Y}}}(\mathcal{W}) = \exp\bigg\{& i\Re(\mathcal{W}^* *_N (\mathcal{A}_1 *_M \mathcal{Y} + \mathcal{A}_2 *_M \mathcal{Y}^*)) \\ & - \frac{1}{4}\Big(\mathcal{W}^* *_N \mathcal{Q}_{WL} *_N \mathcal{W} + \Re(\mathcal{W}^* *_N \tilde{\mathcal{Q}}_{WL} *_N \mathcal{W}^*)\Big)\bigg\} \quad \text{(B.24)}\end{aligned}$$

which is the characteristic function of a complex Gaussian tensor with mean $(\mathcal{A}_1 *_M \mathcal{Y} + \mathcal{A}_2 *_M \mathcal{Y}^*)$, covariance $\mathcal{Q}_{WL}$ and pseudo-covariance $\tilde{\mathcal{Q}}_{WL}$ (based on (B.20)). So we have shown that given $\boldsymbol{\mathcal{Y}} = \mathcal{Y}$, $\boldsymbol{\mathcal{X}}$ is conditionally $\mathcal{CN}(\mathcal{A}_1 *_M \mathcal{Y} + \mathcal{A}_2 *_M \mathcal{Y}^*, \mathcal{Q}_{WL}, \tilde{\mathcal{Q}}_{WL})$. Hence the best MMSE estimate which is the conditional mean is same as the widely multi-linear MMSE estimate for jointly complex Gaussian tensors.

B.6 Tensor Eigenvalue Upper bound

Lemma 11. *For a Hermitian tensor $\mathcal{C} \in \mathbb{C}^{I_1\times\ldots\times I_N\times I_1\times\ldots\times I_N}$, we have $\sigma^2_{max} \leq m + s(\prod_{n=1}^N I_n - 1)^{1/2}$, where σ^2_{max} is the largest eigenvalue of $\mathcal{C}$, $m = \mathrm{tr}(\mathcal{C})/(\prod_{n=1}^N I_n)$ and $s^2 = \mathrm{tr}(\mathcal{C} *_N \mathcal{C}^H)/(\prod_{n=1}^N I_n) - m^2$.*

Proof. Let the tensor EVD of $\mathcal{C}$ be given as : $\mathcal{C} = \mathcal{U} *_N \mathcal{D} *_N \mathcal{U}^H$. Since $\mathcal{C}$ is Hermitian, its eigenvalues are real (from Lemma 2), hence we can write $\mathcal{D} \in \mathbb{R}^{I_1\times\ldots\times I_N\times I_1\times\ldots\times I_N}$. Using (2.19), we know that $\mathrm{tr}(\mathcal{C}) = \mathrm{tr}(\mathcal{U} *_N \mathcal{D} *_N \mathcal{U}^H) = \mathrm{tr}(\mathcal{D} *_N \mathcal{U}^H *_N \mathcal{U}^H) = \mathrm{tr}(\mathcal{D})$. Similarly, $\mathrm{tr}(\mathcal{C} *_N \mathcal{C}^H) = \mathrm{tr}(\mathcal{D} *_N \mathcal{D})$. Thus, we can write

$$m = \frac{\mathrm{tr}(\mathcal{C})}{\prod_{n=1}^N I_n} = \frac{\mathrm{tr}(\mathcal{D})}{\prod_{n=1}^N I_n} = \frac{\sum_{i_1,\ldots,i_N} \mathcal{D}_{i_1,\ldots,i_N,i_1,\ldots,i_N}}{\prod_{n=1}^N I_n}, \quad \text{(B.25)}$$

$$s^2 = \frac{\mathrm{tr}(\mathcal{C} *_N \mathcal{C}^H)}{\prod_{n=1}^N I_n} - m^2 = \frac{\mathrm{tr}(\mathcal{D} *_N \mathcal{D})}{\prod_{n=1}^N I_n} - m^2 = \frac{\sum_{i_1,\ldots,i_N} \mathcal{D}^2_{i_1,\ldots,i_N,i_1,\ldots,i_N}}{\prod_{n=1}^N I_n} - m^2. \tag{B.26}$$

Hence the quantities m and s^2 denote the mean and variance of the $(\prod_{n=1}^N I_n)$ real eigenvalues. Thus, from Samuelson's inequality [201] we get that all the eigenvalues (including the largest) satisfy $\mathcal{D}_{i_1,\ldots,i_N,i_1,\ldots,i_N} \leq m + s(\prod_{n=1}^N I_n - 1)^{1/2}$, which proves Lemma 11. □

B.7 Faster implementation of Newton's Iteration

Reducing the complexity using an alternate NM form

The Newton Method recursion from (2.33) is used to iteratively find the inverse of a tensor. Most often the objective of finding the inverse is to find the solution to a system of multilinear equations represented by $\mathcal{A} *_N \mathcal{X} = \mathcal{Y}$, where $\mathcal{A} \in \mathbb{C}^{I_1 \times \ldots \times I_N \times I_1 \times \ldots \times I_N}$, and $\mathcal{X}, \mathcal{Y} \in \mathbb{C}^{I_1 \times \ldots \times I_N}$. If we use Newton's iteration from (2.33) to find $\mathcal{A}^{-1}$, each iteration requires computing Einstein product $(*_N)$ between tensors of order $2N$. Hence the complexity per iteration is cubic in the size of tensor, i.e. $\mathcal{O}((I_1 \cdots I_N)^3)$. However, employing an alternate form of the Newton's iteration can reduce this complexity from cubic to square in the size of the tensor. Using the Einstein product, we can write (2.33) in expanded form linking $\mathcal{B}^{(k)}$ to $\mathcal{B}^{(0)}$ as [202]:

$$\mathcal{B}^{(k)} = \sum_{m=1}^{2^k-1} c_{k,m} (\mathcal{B}^{(0)} *_N \mathcal{A})^m *_N \mathcal{B}^{(0)} \tag{B.27}$$

where $c_{k,m}$ is the coefficient of the mth summation term in (B.27) and $\mathcal{B}^{(k)}$ is the approximation of $\mathcal{A}^{-1}$ at the kth iteration. For an order $2N$ tensor, the notation $(\mathcal{A})^m$ denotes:

$$(\mathcal{A})^m = \underbrace{\mathcal{A} *_N \mathcal{A} *_N \cdots *_N \mathcal{A}}_{m \text{ times}}. \tag{B.28}$$

Equation (B.27) can be seen as another form of the Newton's method. By considering $c_{k,m}$ as coefficients of a polynomial $f_k(z) = c_{k,0} z^0 + c_{k,1} z^1 + \cdots + c_{k,2^k-1} z^{2^k-1}$, we can write $f_{k+1}(z) = 2 f_k(z) - z[f_k(z)]^2$ with $f_0(z) = 1$ [202]. Thus the coefficients $c_{k,m}$ can be found recursively. In fact these coefficients do not depend on the tensor to be inverted, so can be calculated before hand and used in the solution. Since the objective is to find $\mathcal{X} = \mathcal{A}^{-1} *_N \mathcal{Y}$, rather than approximating $\mathcal{A}^{-1}$ and then taking its Einstein product with $\mathcal{Y}$, we can find the approximation of $\mathcal{A}^{-1} *_N \mathcal{Y}$ directly. Take Einstein product with $\mathcal{Y}$ on both sides in (B.27) to get :

$$\mathcal{B}^{(k)} *_N \mathcal{Y} = \sum_{m=1}^{2^k-1} c_{k,m} \underbrace{(\mathcal{B}^{(0)} *_N \mathcal{A})^m *_N \mathcal{B}^{(0)} *_N \mathcal{Y}}_{\tilde{\mathcal{Y}}^{(m)}} \tag{B.29}$$

where the left hand side is the approximation of $\mathcal{X}$ at the kth iteration, and thus we can write :

$$\mathcal{X}^{(k)} = \sum_{m=1}^{2^k-1} c_{k,m} \tilde{\mathcal{Y}}^{(m)}. \tag{B.30}$$

Since $\tilde{\mathcal{Y}}^{(m)} = (\mathcal{B}^{(0)} *_N \mathcal{A})^m *_N \mathcal{B}^{(0)} *_N \mathcal{Y}$, we have :

$$\tilde{\mathcal{Y}}^{(m+1)} = (\mathcal{B}^{(0)} *_N \mathcal{A}) *_N \tilde{\mathcal{Y}}^{(m)}. \tag{B.31}$$

Hence $\tilde{\mathcal{Y}}^{(m)} \in \mathbb{C}^{I_1 \times \ldots \times I_N}$ can be found recursively with $\tilde{\mathcal{Y}}^{(0)} = \mathcal{B}^{(0)} *_N \mathcal{Y}$. The initial value $\mathcal{B}^{(0)}$ can be taken as in the standard Newton's equation from (2.33) where $\mathcal{B}^{(0)} = a\mathcal{A}^H$, with constant a bounded as $0 < a < 2/\sigma_{max}^2$ and σ_{max}^2 is the largest eigenvalue of $\mathcal{A}^H *_N \mathcal{A}$. Using such an approach the Newton's method now requires taking Einstein product over N modes between a tensor of order $2N$ with a tensor of order N in each iteration, thus reducing the complexity to $\mathcal{O}((I_1 \cdots I_N)^2)$ from $\mathcal{O}((I_1 \cdots I_N)^3)$. The complexity still remains exponential in the number of domains. However, using parallel processing of NM, the exponential dependence on the number of domains can be brought to linear dependence as shown next.

Reducing the complexity using parallel processing

Several flops in the Einstein product can be executed simultaneously on a parallel computing platform. To see this, assume tensors $\mathcal{X}$ and $\mathcal{Y}$ of size $I_1 \times \ldots \times I_N \times I_1 \times \ldots \times I_N$ each and $\mathcal{Z} = \mathcal{X} *_N \mathcal{Y}$. Then all the elements of $\mathcal{Z}$ can be computed using multiple processors as shown in Figure B.1.

The white rectangular nodes in Figure B.1 correspond to the individual multiplication operations. All the white nodes need to be added to compute a single element (the gray rectangular nodes) in tensor $\mathcal{Z}$. The addition of white nodes to generate the gray nodes can be done using a binary tree approach as shown in Figure B.1 where all the addition operations at a given level of the tree are performed simultaneously. The figure illustrates this process for a single gray node, but further all the gray nodes can be computed simultaneously using similar binary tree approach if multiple processors are available.

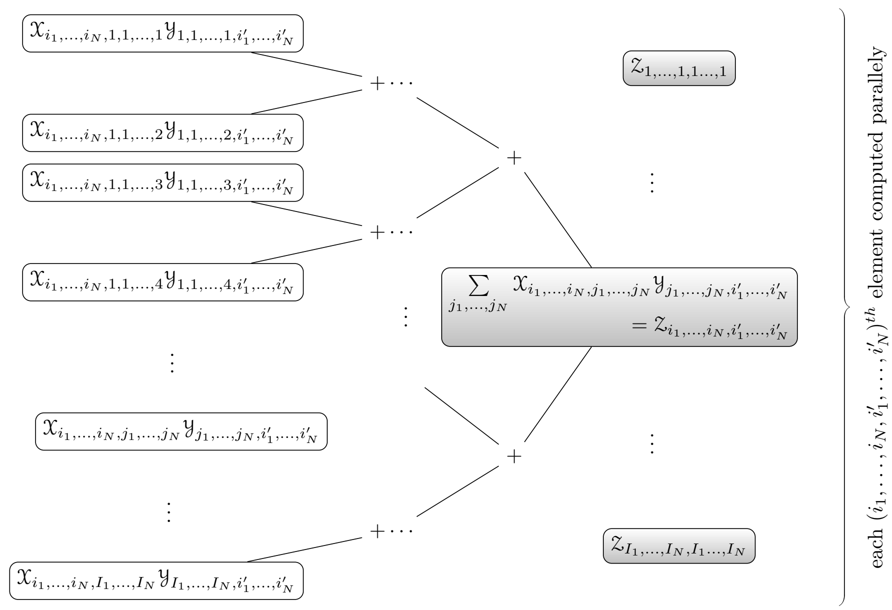


Fig. B.1: Parallel execution of Einstein product.

For a specific gray node, the number of nodes in a binary tree at each level starting from root (gray node) are $2^0, 2^1, 2^2, \ldots, 2^h$ where h is the depth of the tree and 2^h is the number of leaf nodes. We can have similar binary trees for each of the gray nodes. Since in Figure B.1, the number of leaf nodes (white nodes) are $I_1 \cdots I_N$, we get that the depth of the tree corresponding to each gray node is $\lceil \log(I_1 \cdots I_N) \rceil$. We use the ceil operator as $I_1 \cdots I_N$ may not always be a power of 2. Hence we can say that the height of the tree is $\mathcal{O}(\log(I_1 \cdots I_N))$. Since all the gray nodes can be computed simultaneously, all the individual elements of $\mathcal{X} *_N \mathcal{Y}$ can be calculated in $\mathcal{O}(\log(I_1 \cdots I_N))$ parallel time units. Such a parallel processing method can significantly reduce the time complexity of calculating the Einstein product, and subsequently other tensor operations which rely on the Einstein product such as tensor inversion.

Each iteration in NM (2.33), other than the Einstein products, also requires one tensor subtraction which has a time complexity of $\mathcal{O}(1)$ using multiple processors. Hence the time complexity of each Newton iteration using parallel processors is $\mathcal{O}(\log(I_1 \cdots I_N))$. With $I_n = L$ for all n, the complexity is given as $\mathcal{O}(N \log L)$, which is linear in the number of domains.

B.8 Chain Rule for Tensor Derivatives

Let $\mathrm{U} = h(\mathrm{X})$ be a function of matrix X where U is also a matrix. In order the find the derivative of a scalar function $g(\mathrm{U})$ with respect to matrix X, the chain rule is defined as [119, 107] :

$$\frac{\partial g}{\partial \mathrm{X}_{i,j}} = \sum_{k,l} \frac{\partial g}{\partial \mathrm{U}_{k,l}} \cdot \frac{\partial \mathrm{U}_{k,l}}{\partial \mathrm{X}_{i,j}}. \tag{B.32}$$

Similarly, the derivative chain rule for a scalar function $g(h(\mathcal{X}))$, where $\mathcal{X} \in \mathbb{C}^{I_1 \times \ldots \times I_N}$ and $\mathcal{U} = h(\mathcal{X}) \in \mathbb{C}^{J_1 \times \ldots \times J_M}$ are order N and M tensors respectively, can be defined as:

$$\frac{\partial g}{\partial \mathcal{X}_{i_1,\ldots,i_N}} = \sum_{j_1,\ldots,j_M} \frac{\partial g}{\partial \mathcal{U}_{j_1,\ldots,j_M}} \cdot \frac{\partial \mathcal{U}_{j_1,\ldots,j_M}}{\partial \mathcal{X}_{i_1,\ldots,i_N}}. \tag{B.33}$$

We will now present two lemmas based on the chain rule of tensor derivatives which were used for proving Corollary 5.1.

Lemma 12. *For a real valued scalar function $h(\mathcal{D})$ where tensor $\mathcal{D} \in \mathbb{C}^{I_1 \times \ldots \times I_P \times T_1 \times \ldots \times T_Q}$ depends on tensor $\mathcal{B} \in \mathbb{C}^{J_1 \times \ldots \times J_M \times K_1 \times \ldots \times K_N}$ as $\mathcal{D} = \mathcal{A} *_M \mathcal{B} *_N \mathcal{C}$ with tensors $\mathcal{A} \in \mathbb{C}^{I_1 \times \ldots \times I_P \times J_1 \times \ldots \times J_M}$, and $\mathcal{C} \in \mathbb{C}^{K_1 \times \ldots \times K_N \times T_1 \times \ldots \times T_Q}$, we have :*

$$\nabla_{\mathcal{B}} h = \mathcal{A}^H *_P \nabla_{\mathcal{D}} h *_Q \mathcal{C}^H. \tag{B.34}$$

Proof. Since $\mathcal{D} = \mathcal{A} *_M \mathcal{B} *_N \mathcal{C}$, using (2.12) we have $\mathcal{D}^H = \mathcal{C}^H *_N \mathcal{B}^H *_M \mathcal{A}^H$, which can be written component wise as :

$$(\mathcal{D}^H)_{t_1,\ldots,t_Q,i_1,\ldots,i_P} = \sum_{\substack{k_1,\ldots,k_N \\ j_1,\ldots,j_M}} (\mathcal{C}^H)_{t_1,\ldots,t_Q,k_1,\ldots,k_N} (\mathcal{B}^H)_{k_1,\ldots,k_N,j_1,\ldots,j_M} (\mathcal{A}^H)_{j_1,\ldots,j_M,i_1,\ldots,i_P}. \tag{B.35}$$

Note that $(\mathcal{B}^H)_{k_1,\ldots,k_N,j_1,\ldots,j_M}$ is same as $\mathcal{B}^*_{j_1,\ldots,j_M,k_1,\ldots,k_N}$. Thus taking the derivative of (B.35) with respect to a specific component of $\mathcal{B}^*$ can be written as :

$$\frac{\partial(\mathcal{D}^H)_{t_1,\ldots,t_Q,i_1,\ldots,i_P}}{\partial\mathcal{B}^*_{j'_1,\ldots,j'_M,k'_1,\ldots,k'_N}} = (\mathcal{C}^H)_{t_1,\ldots,t_Q,k'_1,\ldots,k'_N}(\mathcal{A}^H)_{j'_1,\ldots,j'_M,i_1,\ldots,i_P}. \tag{B.36}$$

Now using the chain rule from (B.33), we can write :

$$\frac{\partial h}{\partial\mathcal{B}^*_{j'_1,\ldots,j'_M,k'_1,\ldots,k'_N}} = \sum_{\substack{t_1,\ldots,t_Q\\ i_1,\ldots,i_P}} \frac{\partial h}{\partial(\mathcal{D}^H)_{t_1,\ldots,t_Q,i_1,\ldots,i_P}} \cdot \frac{\partial(\mathcal{D}^H)_{t_1,\ldots,t_Q,i_1,\ldots,i_P}}{\partial\mathcal{B}^*_{j'_1,\ldots,j'_M,k'_1,\ldots,k'_N}}. \tag{B.37}$$

Upon substituting (B.36) into (B.37), we get :

$$\begin{aligned}\frac{\partial h}{\partial\mathcal{B}^*_{j'_1,\ldots,j'_M,k'_1,\ldots,k'_N}} &= \sum_{\substack{t_1,\ldots,t_Q\\ i_1,\ldots,i_P}} \frac{\partial h}{\partial(\mathcal{D}^H)_{t_1,\ldots,t_Q,i_1,\ldots,i_P}} \cdot (\mathcal{C}^H)_{t_1,\ldots,t_Q,k'_1,\ldots,k'_N}(\mathcal{A}^H)_{j'_1,\ldots,j'_M,i_1,\ldots,i_P}\\ &= \sum_{\substack{t_1,\ldots,t_Q\\ i_1,\ldots,i_P}} (\mathcal{A}^H)_{j'_1,\ldots,j'_M,i_1,\ldots,i_P} \cdot \frac{\partial h}{\partial\mathcal{D}^*_{i_1,\ldots,i_P,t_1,\ldots,t_Q}} \cdot (\mathcal{C}^H)_{t_1,\ldots,t_Q,k'_1,\ldots,k'_N} \qquad (B.38)\end{aligned}$$

$$\Rightarrow \nabla_{\mathcal{B}} h = \mathcal{A}^H *_P \nabla_{\mathcal{D}} h *_Q \mathcal{C}^H. \tag{B.39}$$

$\square$

Lemma 13. *For a real valued scalar function $h(\mathcal{Q})$, where a Hermitian tensor $\mathcal{Q} \in \mathbb{C}^{I_1\times\ldots\times I_P\times I_1\times\ldots\times I_P}$ depends on tensor $\mathcal{B} \in \mathbb{C}^{J_1\times\ldots\times J_M\times K_1\times\ldots\times K_N}$ as $\mathcal{Q} = \mathcal{A} *_M \mathcal{B} *_N \bar{\mathcal{Q}} *_N \mathcal{B}^H *_M \mathcal{A}^H$ with tensors $\mathcal{A} \in \mathbb{C}^{I_1\times\ldots\times I_P\times J_1\times\ldots\times J_M}$ and a Hermitian tensor $\bar{\mathcal{Q}} \in \mathbb{C}^{K_1\times\ldots\times K_N\times K_1\times\ldots\times K_N}$, we have :*

$$\nabla_{\mathcal{B}} h = \mathcal{A}^H *_P \nabla_{\mathcal{Q}} h *_P \mathcal{A} *_M \mathcal{B} *_N \bar{\mathcal{Q}}^H. \tag{B.40}$$

Proof. The tensor $\mathcal{Q} = \mathcal{A} *_M \mathcal{B} *_N \bar{\mathcal{Q}} *_N \mathcal{B}^H *_M \mathcal{A}^H$ on writing component wise is given by :

$$\begin{aligned}\mathcal{Q}_{i_1,\ldots,i_P,i'_1,\ldots,i'_P} = &\sum_{\substack{j_1,\ldots,j_M,k_1,\ldots,k_N\\ k'_1,\ldots,k'_N,j'_1,\ldots,j'_M}} \mathcal{A}_{i_1,\ldots,i_P,j_1,\ldots,j_M} \cdot \mathcal{B}_{j_1,\ldots,j_M,k_1,\ldots,k_N} \cdot \bar{\mathcal{Q}}_{k_1,\ldots,k_N,k'_1,\ldots,k'_N}\\ &\cdot (\mathcal{B}^H)_{k'_1,\ldots,k'_N,j'_1,\ldots,j'_M} \cdot (\mathcal{A}^H)_{j'_1,\ldots,j'_M,i'_1,\ldots,i'_P}. \qquad (B.41)\end{aligned}$$

Noting that $(\mathcal{B}^H)_{k'_1,\ldots,k'_N,j'_1,\ldots,j'_M}$ is same as $\mathcal{B}^*_{j'_1,\ldots,j'_M,k'_1,\ldots,k'_N}$, we can write the derivative of (B.41) with respect to a specific component of $\mathcal{B}^*$ as :

$$\frac{\partial\mathcal{Q}_{i_1,\ldots,i_P,i'_1,\ldots,i'_P}}{\partial\mathcal{B}^*_{p_1,\ldots,p_M,q_1,\ldots,q_N}} = \sum_{\substack{j_1,\ldots,j_M\\ k_1,\ldots,k_N}} \mathcal{A}_{i_1,\ldots,i_P,j_1,\ldots,j_M}\mathcal{B}_{j_1,\ldots,j_M,k_1,\ldots,k_N}\bar{\mathcal{Q}}_{k_1,\ldots,k_N,q_1,\ldots,q_N}(\mathcal{A}^H)_{p_1,\ldots,p_M,i'_1,\ldots,i'_P}. \tag{B.42}$$

Now using the chain rule from (B.33), we can write :

$$\frac{\partial h}{\partial \mathcal{B}^*_{p_1,\ldots,p_M,q_1,\ldots,q_N}} = \sum_{\substack{i_1,\ldots,i_P\\ i'_1,\ldots,i'_P}} \frac{\partial h}{\partial \mathcal{Q}_{i_1,\ldots,i_P,i'_1,\ldots,i'_P}} \cdot \frac{\partial \mathcal{Q}_{i_1,\ldots,i_P,i'_1,\ldots,i'_P}}{\partial \mathcal{B}^*_{p_1,\ldots,p_M,q_1,\ldots,q_N}}. \tag{B.43}$$

Upon substituting (B.42) into (B.43), we get :

$$\frac{\partial h}{\partial \mathcal{B}^*_{p_1,\ldots,p_M,q_1,\ldots,q_N}} = \sum_{\substack{i_1,\ldots,i_P\\ i'_1,\ldots,i'_P}} \frac{\partial h}{\partial \mathcal{Q}_{i_1,\ldots,i_P,i'_1,\ldots,i'_P}} \cdot \sum_{\substack{j_1,\ldots,j_M\\ k_1,\ldots,k_N}} \mathcal{A}_{i_1,\ldots,i_P,j_1,\ldots,j_M} \cdot \mathcal{B}_{j_1,\ldots,j_M,k_1,\ldots,k_N}$$

$$\cdot \bar{\mathcal{Q}}_{k_1,\ldots,k_N,q_1,\ldots,q_N} \cdot (\mathcal{A}^H)_{p_1,\ldots,p_M,i'_1,\ldots,i'_P}. \tag{B.44}$$

Since $\mathcal{Q}$ is Hermitian, we have $\mathcal{Q}_{i_1,\ldots,i_P,i'_1,\ldots,i'_P} = \mathcal{Q}^*_{i'_1,\ldots,i'_P,i_1,\ldots,i_P}$, we can write (B.44) as :

$$\frac{\partial h}{\partial \mathcal{B}^*_{p_1,\ldots,p_M,q_1,\ldots,q_N}} = \sum_{\substack{i_1,\ldots,i_P,i'_1,\ldots,i'_P\\ j_1,\ldots,j_M,k_1,\ldots,k_N}} (\mathcal{A}^H)_{p_1,\ldots,p_M,i'_1,\ldots,i'_P} \frac{\partial h}{\partial \mathcal{Q}^*_{i'_1,\ldots,i'_P,i_1,\ldots,i_P}}$$

$$\mathcal{A}_{i_1,\ldots,i_P,j_1,\ldots,j_M} \cdot \mathcal{B}_{j_1,\ldots,j_M,k_1,\ldots,k_N} \cdot \bar{\mathcal{Q}}_{k_1,\ldots,k_N,q_1,\ldots,q_N} \tag{B.45}$$

$$\Rightarrow \nabla_{\mathcal{B}} h = \mathcal{A}^H *_P \nabla_{\mathcal{Q}} h *_P \mathcal{A} *_M \mathcal{B} *_N \bar{\mathcal{Q}}^H. \tag{B.46}$$

□

B.9 Proof of Theorem 5, the tensor I-MMSE relationship

In order to prove Theorem 5, we first present a few results which would be required at several intermediate steps in the proof of Theorem 5.

Corollary 5.2. Let $\mathcal{U} = \mathcal{Y} - \mathcal{H} *_N \mathcal{X}$ where $\mathcal{Y} \in \mathbb{C}^{J_1\times\ldots\times J_M}$, $\mathcal{X} \in \mathbb{C}^{I_1\times\ldots\times I_N}$, and $\mathcal{H} \in \mathbb{C}^{J_1\times\ldots J_M\times I_1\times\ldots\times I_N}$, and let $h(\mathcal{U}) = ||\mathcal{U}||^2 = ||\mathcal{Y} - \mathcal{H} *_N \mathcal{X}||^2$, then we have :

$$\nabla_{\mathcal{Y}}(||\mathcal{Y} - \mathcal{H} *_N \mathcal{X}||^2) = (\mathcal{Y} - \mathcal{H} *_N \mathcal{X}), \tag{B.47}$$

$$\nabla_{\mathcal{H}}(||\mathcal{Y} - \mathcal{H} *_N \mathcal{X}||^2) = -(\mathcal{Y} - \mathcal{H} *_N \mathcal{X}) \circ \mathcal{X}^*. \tag{B.48}$$

Proof. Note that $h(\mathcal{U}) = ||\mathcal{U}||^2 = \sum_{j'_1,\ldots,j'_M} \mathcal{U}_{j'_1,\ldots,j'_m} \cdot \mathcal{U}^*_{j'_1,\ldots,j'_M}$. Hence we have

$$[\nabla_{\mathcal{U}} h]_{j_1,\ldots,j_M} = \frac{\partial h}{\partial \mathcal{U}^*_{j_1,\ldots,j_M}} = \frac{\partial}{\partial \mathcal{U}^*_{j_1,\ldots,j_M}} \sum_{j'_1,\ldots,j'_M} \mathcal{U}_{j'_1,\ldots,j'_m} \cdot \mathcal{U}^*_{j'_1,\ldots,j'_M} = \mathcal{U}_{j_1,\ldots,j_M}. \tag{B.49}$$

Thus we get $\nabla_{\mathcal{U}} h = \mathcal{U}$, which directly leads to $\nabla_{\mathcal{Y}}(||\mathcal{Y} - \mathcal{H} *_N \mathcal{X}||^2) = (\mathcal{Y} - \mathcal{H} *_N \mathcal{X})$ and hence proves (B.47). Now to prove (B.48), we invoke Lemma 12 from Appendix B.8. In the statement of Lemma 12, replace $\mathcal{A}$ with a scaled identity tensor $(-1)\cdot\mathcal{I}_M$ of order $2M$, $\mathcal{B}$ with a tensor $\mathcal{H}$ of order $M+N$, $\mathcal{C}$ with a tensor $\mathcal{X}$ of order N, and function h as $||\mathcal{Y}-\mathcal{H}*_N\mathcal{X}||^2$. Then from (B.34) we get :

$$\nabla_{\mathcal{H}} h = (-1)\cdot\mathcal{I}_M *_M \nabla_{\mathcal{U}} h *_0 \mathcal{X}^* = -\mathcal{U} *_0 \mathcal{X}^*. \tag{B.50}$$

From the definition of the outer product in (2.5), and substituting $\mathcal{U}$ and h, we can write :

$$\nabla_{\mathcal{H}}(||\mathcal{Y}-\mathcal{H}*_N\mathcal{X}||^2) = -(\mathcal{Y}-\mathcal{H}*_N\mathcal{X})\circ\mathcal{X}^*. \tag{B.51}$$

□

Also, the following lemma would be used in proving Theorem 5:

Lemma 14. *Let $h(\boldsymbol{\mathcal{X}};\mathcal{A})$ be a real valued scalar function of a random tensor $\boldsymbol{\mathcal{X}}$ and a parameter tensor $\mathcal{A}\in\mathbb{C}^{I_1\times\ldots\times I_N}$ which is differentiable for all $\mathcal{A}$, then if $\left|\frac{\partial}{\partial\mathcal{A}^*_{i_1,\ldots,i_N}}h(\boldsymbol{\mathcal{X}};\mathcal{A})\right|\leq z$ where z is a random scalar with finite mean, then :*

$$\nabla_{\mathcal{A}}\mathbb{E}_{\boldsymbol{\mathcal{X}}}[h(\boldsymbol{\mathcal{X}};\mathcal{A})] = \mathbb{E}_{\boldsymbol{\mathcal{X}}}[\nabla_{\mathcal{A}} h(\boldsymbol{\mathcal{X}};\mathcal{A})], \tag{B.52}$$

i.e. the sequence of expectation and gradient can be interchanged.

Proof. We have

$$\mathbb{E}_{\boldsymbol{\mathcal{X}}}[h(\boldsymbol{\mathcal{X}};\mathcal{A})] = \int h(\mathcal{X};\mathcal{A})p_{\boldsymbol{\mathcal{X}}}(\mathcal{X})\partial\mathcal{X} \tag{B.53}$$

$$\Rightarrow \nabla_{\mathcal{A}}\mathbb{E}_{\boldsymbol{\mathcal{X}}}[h(\boldsymbol{\mathcal{X}};\mathcal{A})] = \frac{\partial}{\partial\mathcal{A}^*}\int h(\mathcal{X};\mathcal{A})p_{\boldsymbol{\mathcal{X}}}(\mathcal{X})\partial\mathcal{X} \tag{B.54}$$

which can be written element wise as

$$\Big[\nabla_{\mathcal{A}}\mathbb{E}_{\boldsymbol{\mathcal{X}}}[h(\boldsymbol{\mathcal{X}};\mathcal{A})]\Big]_{i_1,\ldots,i_N} = \frac{\partial}{\partial\mathcal{A}^*_{i_1,\ldots,i_N}}\int h(\mathcal{X};\mathcal{A})p_{\boldsymbol{\mathcal{X}}}(\mathcal{X})\partial\mathcal{X}. \tag{B.55}$$

Since we assume that $|\partial h/\partial\mathcal{A}^*_{i_1,\ldots,i_N}|$ is bounded by a random scalar with finite mean, we can use the Dominated Convergence Theorem [203], [107, Lemma 2], to bring the derivative inside integral and write :

$$\Big[\nabla_{\mathcal{A}} \mathbb{E}_{\boldsymbol{\mathfrak{X}}}[h(\boldsymbol{\mathfrak{X}};\mathcal{A})]\Big]_{i_1,\dots,i_N} = \int \frac{\partial}{\partial \mathcal{A}^*_{i_1,\dots,i_N}} h(\mathfrak{X};\mathcal{A}) p_{\boldsymbol{\mathfrak{x}}}(\mathfrak{X}) \partial \mathfrak{X} \tag{B.56}$$

$$\Rightarrow \nabla_{\mathcal{A}} \mathbb{E}_{\boldsymbol{\mathfrak{X}}}[h(\boldsymbol{\mathfrak{X}};\mathcal{A})] = \int \frac{\partial}{\partial \mathcal{A}^*} h(\mathfrak{X};\mathcal{A}) p_{\boldsymbol{\mathfrak{x}}}(\mathfrak{X}) \mathtt{d}\mathfrak{X} \tag{B.57}$$

$$= \int \nabla_{\mathcal{A}} h(\mathfrak{X};\mathcal{A}) p_{\boldsymbol{\mathfrak{x}}}(\mathfrak{X}) \mathtt{d}\mathfrak{X} \tag{B.58}$$

$$= \mathbb{E}_{\boldsymbol{\mathfrak{X}}}[\nabla_{\mathcal{A}} h(\boldsymbol{\mathfrak{X}};\mathcal{A})]. \tag{B.59}$$

$\square$

Integration by Parts for scalar function of Tensors : If $h(\mathcal{Y})$ and $g(\mathcal{Y})$ are scalar functions of a tensor $\mathcal{Y} \in \mathbb{C}^{J_1 \times \dots \times J_M}$, then from the product rule of gradient we can write [204]:

$$\nabla_{\mathcal{Y}} \Big(h(\mathcal{Y})g(\mathcal{Y})\Big) = g(\mathcal{Y})\nabla_{\mathcal{Y}} h(\mathcal{Y}) + h(\mathcal{Y})\nabla_{\mathcal{Y}} g(\mathcal{Y}) \tag{B.60}$$

where $\nabla_{\mathcal{Y}} \Big(h(\mathcal{Y})g(\mathcal{Y})\Big)$ is a tensor of same size as $\mathcal{Y}$. This equation can be rearranged as :

$$g(\mathcal{Y})\nabla_{\mathcal{Y}} h(\mathcal{Y}) = \nabla_{\mathcal{Y}} \Big(h(\mathcal{Y})g(\mathcal{Y})\Big) - h(\mathcal{Y})\nabla_{\mathcal{Y}} g(\mathcal{Y}) \tag{B.61}$$

and subsequently a form of integration by parts can be written as :

$$\int g(\mathcal{Y})\nabla_{\mathcal{Y}} h(\mathcal{Y}) \mathtt{d}\mathcal{Y} = \int \nabla_{\mathcal{Y}} \Big(h(\mathcal{Y})g(\mathcal{Y})\Big) \mathtt{d}\mathcal{Y} - \int h(\mathcal{Y})\nabla_{\mathcal{Y}} g(\mathcal{Y}) \mathtt{d}\mathcal{Y}. \tag{B.62}$$

It is important to note here that each term in (B.62) is an integral as defined in (2.30) and is a tensor not a scalar. Thus unlike the scalar case, the first term on the right hand side $\int \nabla_{\mathcal{Y}}(h(\mathcal{Y})g(\mathcal{Y}))\mathtt{d}\mathcal{Y}$ can not simply be written as $h(\mathcal{Y})g(\mathcal{Y})$.

We now present the proof of the tensor I-MMSE relation.

Proof of Theorem 5:

Since the noise tensor $\boldsymbol{\mathcal{N}}$ is circular symmetric Gaussian with zero mean and identity covariance tensor, the conditional pdf of the output is given as :

$$p_{\boldsymbol{\mathfrak{y}}|\boldsymbol{\mathfrak{x}}}(\mathcal{Y}|\mathfrak{X}) = \frac{1}{\pi^{J_1 \cdot J_2 \cdots J_M}} \exp\Big(- (\mathcal{Y} - \mathcal{H} *_N \mathfrak{X})^* *_M (\mathcal{Y} - \mathcal{H} *_N \mathfrak{X})\Big) \tag{B.63}$$

$$= \frac{1}{\pi^{J_1 \cdot J_2 \cdots J_M}} \exp\big(- ||\mathcal{Y} - \mathcal{H} *_N \mathfrak{X}||^2\big) \tag{B.64}$$

and the unconditional output pdf is

$$p_{\mathcal{Y}}(\mathcal{Y}) = \mathbb{E}_{\boldsymbol{\mathcal{X}}}[p_{\mathcal{Y}|\boldsymbol{\mathcal{X}}}(\mathcal{Y}|\boldsymbol{\mathcal{X}})]. \tag{B.65}$$

Also, the noise tensor $\boldsymbol{\mathcal{N}}$ is independent of the input tensor $\boldsymbol{\mathcal{X}}$, thus the mutual information is given by

$$\mathcal{I}(\boldsymbol{\mathcal{X}};\boldsymbol{\mathcal{Y}}) = \mathcal{H}(\boldsymbol{\mathcal{Y}}) - \mathcal{H}(\boldsymbol{\mathcal{Y}}|\boldsymbol{\mathcal{X}}) = \mathcal{H}(\boldsymbol{\mathcal{Y}}) - \mathcal{H}(\boldsymbol{\mathcal{N}}) \tag{B.66}$$

where covariance of the noise tensor is identity $\mathfrak{I}_M$. Using (3.1), $\mathcal{H}(\boldsymbol{\mathcal{N}})$ can be written as :

$$\mathcal{H}(\boldsymbol{\mathcal{N}}) = \log\Big((e\pi)^{J_1\ldots J_M}\det(\mathfrak{I}_M)\Big) = \log((e\pi)^{J_1\cdots J_M}) \tag{B.67}$$

which makes the mutual information :

$$\mathcal{I}(\boldsymbol{\mathcal{X}};\boldsymbol{\mathcal{Y}}) = -\mathbb{E}_{\boldsymbol{\mathcal{Y}}}[\log p_{\mathcal{Y}}(\boldsymbol{\mathcal{Y}})] - (J_1\cdots J_M\log(\pi e)) \tag{B.68}$$

$$= -\int p_{\mathcal{Y}}(\mathcal{Y})\log(p_{\mathcal{Y}}(\mathcal{Y}))\partial\mathcal{Y} - (J_1\cdots J_M\log(\pi e)) \tag{B.69}$$

$$\Rightarrow \frac{\partial}{\partial\mathcal{H}^*}\mathcal{I} = -\frac{\partial}{\partial\mathcal{H}^*}\int p_{\mathcal{Y}}(\mathcal{Y})\log(p_{\mathcal{Y}}(\mathcal{Y}))\partial\mathcal{Y}. \tag{B.70}$$

Taking the derivative inside integral and using the product rule from (B.60), we get :

$$\frac{\partial}{\partial\mathcal{H}^*}\mathcal{I} = -\int\Big(p_{\mathcal{Y}}(\mathcal{Y})\frac{\partial}{\partial\mathcal{H}^*}\log(p_{\mathcal{Y}}(\mathcal{Y})) + \log(p_{\mathcal{Y}}(\mathcal{Y}))\frac{\partial}{\partial\mathcal{H}^*}p_{\mathcal{Y}}(\mathcal{Y})\Big)\mathtt{d}\mathcal{Y} \tag{B.71}$$

$$= -\int\Big(p_{\mathcal{Y}}(\mathcal{Y})\frac{1}{p_{\mathcal{Y}}(\mathcal{Y})}\frac{\partial}{\partial\mathcal{H}^*}p_{\mathcal{Y}}(\mathcal{Y}) + \log(p_{\mathcal{Y}}(\mathcal{Y}))\frac{\partial}{\partial\mathcal{H}^*}p_{\mathcal{Y}}(\mathcal{Y})\Big)\mathtt{d}\mathcal{Y} \tag{B.72}$$

$$= -\int(\log(p_{\mathcal{Y}}(\mathcal{Y})) + 1)\frac{\partial}{\partial\mathcal{H}^*}p_{\mathcal{Y}}(\mathcal{Y})\mathtt{d}\mathcal{Y}. \tag{B.73}$$

On substituting $p_{\mathcal{Y}}(\mathcal{Y})$ from (B.65) into (B.73), we get :

$$\frac{\partial}{\partial\mathcal{H}^*}\mathcal{I} = -\int\Big((\log(p_{\mathcal{Y}}(\mathcal{Y})) + 1)\frac{\partial}{\partial\mathcal{H}^*}\mathbb{E}_{\boldsymbol{\mathcal{X}}}[p_{\mathcal{Y}|\boldsymbol{\mathcal{X}}}(\mathcal{Y}|\boldsymbol{\mathcal{X}})]\Big)\mathtt{d}\mathcal{Y}. \tag{B.74}$$

Based on Lemma 14, we can take the derivative inside expectation and get,

$$\frac{\partial}{\partial\mathcal{H}^*}\mathcal{I} = -\int\Big((\log(p_{\mathcal{Y}}(\mathcal{Y})) + 1)\mathbb{E}_{\boldsymbol{\mathcal{X}}}\Big[\frac{\partial}{\partial\mathcal{H}^*}p_{\mathcal{Y}|\boldsymbol{\mathcal{X}}}(\mathcal{Y}|\boldsymbol{\mathcal{X}})\Big]\Big)\mathtt{d}\mathcal{Y}. \tag{B.75}$$

To see that the conditions for Lemma 14 are satisfied, note that the derivative of the output conditional pdf with respect to the channel can be written as:

$$\frac{\partial}{\partial \mathcal{H}^*} p_{\mathbf{y}|\mathbf{x}}(\mathcal{Y}|\mathcal{X}) = \frac{\partial}{\partial \mathcal{H}^*}\Big(\frac{\exp\big(-||\mathcal{Y}-\mathcal{H}*_N \mathcal{X}||^2\big)}{\pi^{J_1\cdot J_2\cdots J_M}}\Big), \quad \text{(using (B.64))} \tag{B.76}$$

$$= \frac{\exp\big(-||\mathcal{Y}-\mathcal{H}*_N \mathcal{X}||^2\big)}{\pi^{J_1\cdot J_2\cdots J_M}} \frac{\partial}{\partial \mathcal{H}^*}\Big(-||\mathcal{Y}-\mathcal{H}*_N \mathcal{X}||^2\Big) \tag{B.77}$$

$$= -p_{\mathbf{y}|\mathbf{x}}(\mathcal{Y}|\mathcal{X})\frac{\partial}{\partial \mathcal{H}^*}\Big(||\mathcal{Y}-\mathcal{H}*_N \mathcal{X}||^2\Big) \qquad \text{(using (B.64))} \tag{B.78}$$

$$= p_{\mathbf{y}|\mathbf{x}}(\mathcal{Y}|\mathcal{X})(\mathcal{Y}-\mathcal{H}*_N \mathcal{X})\circ \mathcal{X}^* \qquad \text{(using (B.48)).} \tag{B.79}$$

Since $0 \leq p_{\mathbf{y}|\mathbf{x}}(\mathcal{Y}|\mathcal{X}) \leq 1$, from (B.79) we can write :

$$\left|\frac{\partial}{\partial \mathcal{H}^*_{j_1,\dots,j_M,i_1,\dots,i_N}} p_{\mathbf{y}|\mathbf{x}}(\mathcal{Y}|\mathcal{X})\right| \leq |((\mathcal{Y}-\mathcal{H}*_N \mathcal{X})\circ \mathcal{X}^*)_{j_1,\dots,j_M,i_1,\dots,i_N}| \tag{B.80}$$

where for a given channel $\mathcal{H}$, the expectation of the right hand side will be finite because of finite second order moments, hence the conditions of Lemma 14 are satisfied for interchange of gradient and expectation in (B.75).

Similarly the gradient of the conditional pdf with respect to $\mathcal{Y}$ can be written as

$$\frac{\partial}{\partial \mathcal{Y}^*} p_{\mathbf{y}|\mathbf{x}}(\mathcal{Y}|\mathcal{X}) = \frac{\partial}{\partial \mathcal{Y}^*}\Big(\frac{\exp\big(-||\mathcal{Y}-\mathcal{H}*_N \mathcal{X}||^2\big)}{\pi^{J_1\cdot J_2\cdots J_M}}\Big), \quad \text{(using (B.64))} \tag{B.81}$$

$$= \frac{\exp\big(-||\mathcal{Y}-\mathcal{H}*_N \mathcal{X}||^2\big)}{\pi^{J_1\cdot J_2\cdots J_M}} \frac{\partial}{\partial \mathcal{Y}^*}\Big(-||\mathcal{Y}-\mathcal{H}*_N \mathcal{X}||^2\Big) \tag{B.82}$$

$$= -p_{\mathbf{y}|\mathbf{x}}(\mathcal{Y}|\mathcal{X})(\mathcal{Y}-\mathcal{H}*_N \mathcal{X}), \quad \text{(using (B.64) and (B.47))} \tag{B.83}$$

$$\Rightarrow -\nabla_{\mathcal{Y}} p_{\mathbf{y}|\mathbf{x}}(\mathcal{Y}|\mathcal{X}) = p_{\mathbf{y}|\mathbf{x}}(\mathcal{Y}|\mathcal{X})(\mathcal{Y}-\mathcal{H}*_N \mathcal{X}). \tag{B.84}$$

On substituting $p_{\mathbf{y}|\mathbf{x}}(\mathcal{Y}|\mathcal{X})(\mathcal{Y}-\mathcal{H}*_N \mathcal{X})$ from(B.84) into (B.79), we get :

$$\frac{\partial}{\partial \mathcal{H}^*} p_{\mathbf{y}|\mathbf{x}}(\mathcal{Y}|\mathcal{X}) = -\nabla_{\mathcal{Y}}\, p_{\mathbf{y}|\mathbf{x}}(\mathcal{Y}|\mathcal{X})\circ \mathcal{X}^*. \tag{B.85}$$

Further, on substituting (B.85) into (B.75) :

$$\frac{\partial}{\partial \mathcal{H}^*}\mathcal{I} = -\int \Big((\log(p_{\mathbf{y}}(\mathcal{Y}))+1)\mathbb{E}_{\boldsymbol{\mathcal{X}}}\Big[-\nabla_{\mathcal{Y}} p_{\mathbf{y}|\mathbf{x}}(\mathcal{Y}|\boldsymbol{\mathcal{X}})\circ \boldsymbol{\mathcal{X}}^*\Big]\Big)\mathtt{d}\mathcal{Y} \tag{B.86}$$

$$= \mathbb{E}_{\boldsymbol{\mathcal{X}}}\Big[\int \Big((\log(p_{\mathbf{y}}(\mathcal{Y}))+1)\cdot \nabla_{\mathcal{Y}} p_{\mathbf{y}|\mathbf{x}}(\mathcal{Y}|\boldsymbol{\mathcal{X}})\circ \boldsymbol{\mathcal{X}}^*\Big)\mathtt{d}\mathcal{Y}\Big]. \tag{B.87}$$

Since the integral is with respect to $\mathcal{Y}$, we can take $\boldsymbol{\mathcal{X}}^*$ out of the integral to form :

$$\frac{\partial}{\partial \mathcal{H}^*}\mathcal{I} = \mathbb{E}_{\boldsymbol{\mathcal{X}}}\Big[\int \Big((\log(p_{\mathbf{y}}(\mathcal{Y}))+1)\cdot \nabla_{\mathcal{Y}} p_{\mathbf{y}|\mathbf{x}}(\mathcal{Y}|\boldsymbol{\mathcal{X}})\Big)\mathtt{d}\mathcal{Y}\circ \boldsymbol{\mathcal{X}}^*\Big]. \tag{B.88}$$

The integral in (B.88) can be written using (B.62) as:

$$\int \Big((\log(p_{\mathsf{y}}(\mathcal{Y})) + 1) \cdot \nabla_{\mathcal{Y}} p_{\mathsf{y}|\mathsf{x}}(\mathcal{Y}|\boldsymbol{\mathcal{X}}) \Big) \mathtt{d}\mathcal{Y} = \underbrace{\int \nabla_{\mathcal{Y}} \Big((\log(p_{\mathsf{y}}(\mathcal{Y})) + 1) \cdot p_{\mathsf{y}|\mathsf{x}}(\mathcal{Y}|\boldsymbol{\mathcal{X}}) \Big) \mathtt{d}\mathcal{Y}}_{\mathcal{A}} - \underbrace{\int p_{\mathsf{y}|\mathsf{x}}(\mathcal{Y}|\boldsymbol{\mathcal{X}}) \nabla_{\mathcal{Y}} (\log(p_{\mathsf{y}}(\mathcal{Y})) + 1) \mathtt{d}\mathcal{Y}}_{\mathcal{B}}. \quad \text{(B.89)}$$

The term $\mathcal{B}$ in (B.89) can be written as :

$$\mathcal{B} = \int p_{\mathsf{y}|\mathsf{x}}(\mathcal{Y}|\boldsymbol{\mathcal{X}}) \frac{1}{p_{\mathsf{y}}(\mathcal{Y})} \nabla_{\mathcal{Y}} (p_{\mathsf{y}}(\mathcal{Y})) \mathtt{d}\mathcal{Y}. \quad \text{(B.90)}$$

The term $\mathcal{A}$ in (B.89) can be written element wise as (using (2.30)):

$$\mathcal{A}_{j_1,\dots,j_M} = \int \Big(\nabla_{\mathcal{Y}} \Big((\log(p_{\mathsf{y}}(\mathcal{Y})) + 1) \cdot p_{\mathsf{y}|\mathsf{x}}(\mathcal{Y}|\boldsymbol{\mathcal{X}}) \Big) \Big)_{j_1,\dots,j_M} \partial\mathcal{Y} \quad \text{(B.91)}$$

$$= \int \cdots \int \Big[\int \nabla_{\mathcal{Y}} \Big((\log(p_{\mathsf{y}}(\mathcal{Y})) + 1) \cdot p_{\mathsf{y}|\mathsf{x}}(\mathcal{Y}|\boldsymbol{\mathcal{X}}) \Big)_{j_1,\dots,j_M} \partial\mathcal{Y}_{j_1,\dots,j_M} \Big] \underbrace{\partial\mathcal{Y}_{1,\dots,1} \cdots \partial\mathcal{Y}_{J_1,\dots,J_M}}_{\text{all elements except } \partial\mathcal{Y}_{j_1,\dots,j_M}}. \quad \text{(B.92)}$$

Note that the function $(\log(p_{\mathsf{y}}(\mathcal{Y})) + 1) \cdot p_{\mathsf{y}|\mathsf{x}}(\mathcal{Y}|\boldsymbol{\mathcal{X}})$ is a real valued scalar function of tensor $\mathcal{Y}$. Let us denote it using $a(\mathcal{Y}) = (\log(p_{\mathsf{y}}(\mathcal{Y})) + 1) \cdot p_{\mathsf{y}|\mathsf{x}}(\mathcal{Y}|\boldsymbol{\mathcal{X}})$ which can also be seen as a function of the real and imaginary components of the tensor $\mathcal{Y}$. Also, as $|\mathcal{Y}_{j_1,\dots,j_M}| \to \infty$ both $p_{\mathsf{y}}(\mathcal{Y})$ and $p_{\mathsf{y}|\mathsf{x}}(\mathcal{Y}|\mathcal{X})$ tends to 0, hence the function $a(\mathcal{Y}) \to 0$. Hence the integral inside the square brackets in (B.92) when evaluated from $-\infty$ to $+\infty$ for both real and imaginary parts of $\mathcal{Y}_{j_1,\dots,j_M}$ tends to 0, which makes the square bracket term 0 in (B.92) and subsequently $\mathcal{A}_{j_1,\dots,j_M} = 0$. This implies that $\mathcal{A}$ is an all zero tensor, i.e. $\mathcal{A} = 0_{\mathcal{T}}$. Substituting $\mathcal{A} = 0_{\mathcal{T}}$ and $\mathcal{B}$ from (B.90) into (B.89) gives us :

$$\int \Big((\log(p_{\mathsf{y}}(\mathcal{Y})) + 1) \cdot \nabla_{\mathcal{Y}} p_{\mathsf{y}|\mathsf{x}}(\mathcal{Y}|\boldsymbol{\mathcal{X}}) \Big) \mathtt{d}\mathcal{Y} = -\int p_{\mathsf{y}|\mathsf{x}}(\mathcal{Y}|\boldsymbol{\mathcal{X}}) \frac{1}{p_{\mathsf{y}}(\mathcal{Y})} \nabla_{\mathcal{Y}} (p_{\mathsf{y}}(\mathcal{Y})) \mathtt{d}\mathcal{Y}. \quad \text{(B.93)}$$

On substituting (B.93) into (B.88), we get :

$$\frac{\partial}{\partial \mathcal{H}^*} \mathcal{I} = \mathbb{E}_{\boldsymbol{\mathcal{X}}} \Big[\int \frac{-p_{\mathsf{y}|\mathsf{x}}(\mathcal{Y}|\boldsymbol{\mathcal{X}})}{p_{\mathsf{y}}(\mathcal{Y})} \nabla_{\mathcal{Y}} \, p_{\mathsf{y}}(\mathcal{Y}) \mathtt{d}\mathcal{Y} \circ \boldsymbol{\mathcal{X}}^* \Big] \quad \text{(B.94)}$$

$$= -\int \nabla_{\mathcal{Y}} \, p_{\mathsf{y}}(\mathcal{Y}) \circ \Big(\mathbb{E}_{\boldsymbol{\mathcal{X}}} \Big[\frac{p_{\mathsf{y}|\mathsf{x}}(\mathcal{Y}|\boldsymbol{\mathcal{X}})}{p_{\mathsf{y}}(\mathcal{Y})} \boldsymbol{\mathcal{X}}^* \Big] \Big) \mathtt{d}\mathcal{Y}. \quad \text{(B.95)}$$

The expectation term inside the integral in (B.95) can be written as :

$$\mathbb{E}_{\boldsymbol{\mathcal{X}}}\Big[\frac{p_{\boldsymbol{\mathcal{Y}}|\boldsymbol{\mathcal{X}}}(\mathcal{Y}|\boldsymbol{\mathcal{X}})}{p_{\boldsymbol{\mathcal{Y}}}(\mathcal{Y})}\boldsymbol{\mathcal{X}}^*\Big] = \int \mathcal{X}^* \frac{p_{\boldsymbol{\mathcal{Y}}|\boldsymbol{\mathcal{X}}}(\mathcal{Y}|\mathcal{X})}{p_{\boldsymbol{\mathcal{Y}}}(\mathcal{Y})} p_{\boldsymbol{\mathcal{X}}}(\mathcal{X})\mathtt{d}\mathcal{X} = \int \mathcal{X}^* p_{\boldsymbol{\mathcal{X}}|\boldsymbol{\mathcal{Y}}}(\mathcal{X}|\mathcal{Y})\mathtt{d}\mathcal{X} = \mathbb{E}_{\boldsymbol{\mathcal{X}}|\boldsymbol{\mathcal{Y}}}[\boldsymbol{\mathcal{X}}^*|\boldsymbol{\mathcal{Y}} = \mathcal{Y}] \tag{B.96}$$

and the term $\nabla_{\mathcal{Y}}\, p_{\boldsymbol{\mathcal{Y}}}(\mathcal{Y})$ inside the integral in (B.95) can be written using $p_{\boldsymbol{\mathcal{Y}}}(\mathcal{Y})$ from (B.65) as :

$$\nabla_{\mathcal{Y}}\, p_{\boldsymbol{\mathcal{Y}}}(\mathcal{Y}) = \nabla_{\mathcal{Y}} \mathbb{E}_{\boldsymbol{\mathcal{X}}}[p_{\boldsymbol{\mathcal{Y}}|\boldsymbol{\mathcal{X}}}(\mathcal{Y}|\boldsymbol{\mathcal{X}})] = \mathbb{E}_{\boldsymbol{\mathcal{X}}}[\nabla_{\mathcal{Y}} p_{\boldsymbol{\mathcal{Y}}|\boldsymbol{\mathcal{X}}}(\mathcal{Y}|\boldsymbol{\mathcal{X}})]. \tag{B.97}$$

On substituting $\nabla_{\mathcal{Y}} p_{\boldsymbol{\mathcal{Y}}|\boldsymbol{\mathcal{X}}}(\mathcal{Y}|\mathcal{X})$ from (B.84) into (B.97), we get:

$$\begin{aligned}
\nabla_{\mathcal{Y}}\, p_{\boldsymbol{\mathcal{Y}}}(\mathcal{Y}) &= \mathbb{E}_{\boldsymbol{\mathcal{X}}}[-p_{\boldsymbol{\mathcal{Y}}|\boldsymbol{\mathcal{X}}}(\mathcal{Y}|\boldsymbol{\mathcal{X}}) \cdot (\mathcal{Y} - \mathcal{H} *_N \boldsymbol{\mathcal{X}})] && \text{(B.98)}\\
&= \int p_{\boldsymbol{\mathcal{Y}}|\boldsymbol{\mathcal{X}}}(\mathcal{Y}|\mathcal{X}) \cdot (\mathcal{Y} - \mathcal{H} *_N \mathcal{X}) p_{\boldsymbol{\mathcal{X}}}(\mathcal{X})\mathtt{d}\mathcal{X} && \text{(B.99)}\\
&= -\mathbb{E}_{\boldsymbol{\mathcal{X}}|\boldsymbol{\mathcal{Y}}}[p_{\boldsymbol{\mathcal{Y}}}(\mathcal{Y})(\mathcal{Y} - \mathcal{H} *_N \boldsymbol{\mathcal{X}})|\boldsymbol{\mathcal{Y}} = \mathcal{Y}] && \text{(B.100)}\\
&= -p_{\boldsymbol{\mathcal{Y}}}(\mathcal{Y})(\mathcal{Y} - \mathcal{H} *_N \mathbb{E}_{\boldsymbol{\mathcal{X}}|\boldsymbol{\mathcal{Y}}}[\boldsymbol{\mathcal{X}}|\boldsymbol{\mathcal{Y}} = \mathcal{Y}]). && \text{(B.101)}
\end{aligned}$$

Substituting (B.96) and (B.101) into (B.95), we get:

$$\begin{aligned}
\frac{\partial}{\partial \mathcal{H}^*}\mathcal{I} &= \int p_{\boldsymbol{\mathcal{Y}}}(\mathcal{Y})(\mathcal{Y} - \mathcal{H} *_N \mathbb{E}_{\boldsymbol{\mathcal{X}}|\boldsymbol{\mathcal{Y}}}[\boldsymbol{\mathcal{X}}|\boldsymbol{\mathcal{Y}} = \mathcal{Y}]) \circ \mathbb{E}_{\boldsymbol{\mathcal{X}}|\boldsymbol{\mathcal{Y}}}[\boldsymbol{\mathcal{X}}^*|\boldsymbol{\mathcal{Y}} = \mathcal{Y}]\mathtt{d}\mathcal{Y} && \text{(B.102)}\\
&= \mathbb{E}\Big[(\boldsymbol{\mathcal{Y}} - \mathcal{H} *_N \mathbb{E}_{\boldsymbol{\mathcal{X}}|\boldsymbol{\mathcal{Y}}}[\boldsymbol{\mathcal{X}}|\boldsymbol{\mathcal{Y}}]) \circ \mathbb{E}_{\boldsymbol{\mathcal{X}}|\boldsymbol{\mathcal{Y}}}[\boldsymbol{\mathcal{X}}^*|\boldsymbol{\mathcal{Y}}]\Big] && \text{(B.103)}\\
&= \mathbb{E}\Big[\boldsymbol{\mathcal{Y}} \circ \mathbb{E}[\boldsymbol{\mathcal{X}}^*|\boldsymbol{\mathcal{Y}}]\Big] - \mathcal{H} *_N \mathbb{E}\Big[\mathbb{E}[\boldsymbol{\mathcal{X}}|\boldsymbol{\mathcal{Y}}] \circ \mathbb{E}[\boldsymbol{\mathcal{X}}^*|\boldsymbol{\mathcal{Y}}]\Big] && \text{(B.104)}\\
&= \mathbb{E}\Big[\mathbb{E}[\boldsymbol{\mathcal{Y}} \circ \boldsymbol{\mathcal{X}}^*|\boldsymbol{\mathcal{Y}}]\Big] - \mathcal{H} *_N \mathbb{E}\Big[\mathbb{E}[\boldsymbol{\mathcal{X}}|\boldsymbol{\mathcal{Y}}] \circ \mathbb{E}[\boldsymbol{\mathcal{X}}^*|\boldsymbol{\mathcal{Y}}]\Big] && \text{(B.105)}\\
&= \mathbb{E}[\boldsymbol{\mathcal{Y}} \circ \boldsymbol{\mathcal{X}}^*] - \mathcal{H} *_N \mathbb{E}\Big[\mathbb{E}[\boldsymbol{\mathcal{X}}|\boldsymbol{\mathcal{Y}}] \circ \mathbb{E}[\boldsymbol{\mathcal{X}}^*|\boldsymbol{\mathcal{Y}}]\Big] && \text{(B.106)}\\
&= \mathcal{H} *_N \mathbb{E}[\boldsymbol{\mathcal{X}} \circ \boldsymbol{\mathcal{X}}^*] - \mathcal{H} *_N \mathbb{E}\Big[\mathbb{E}[\boldsymbol{\mathcal{X}}|\boldsymbol{\mathcal{Y}}] \circ \mathbb{E}[\boldsymbol{\mathcal{X}}^*|\boldsymbol{\mathcal{Y}}]\Big] && \text{(B.107)}\\
&= \mathcal{H} *_N \mathcal{Q}_{\boldsymbol{\mathcal{E}}} && \text{(B.108)}
\end{aligned}$$

where the last equality follows based on (5.5).

Appendix C

Simulation Programs guide

All the numerical and simulation results presented in this thesis were generated using MATLAB R2019a on 64 bit Linux systems. The software files for all the results have been kept in a folder titled 'Thesis Software' which can be obtained following the university regulations by contacting the author or his supervisor. The folder contains sub-folders corresponding to each chapter of this thesis. Each sub-folder contains the MATLAB script and function files required to generate the figures in this thesis. Also the mat files required for Figure 3.21 can be found in the folder of Chapter 3. This Appendix contains a list of all the MATLAB script files in the software distribution folder. A ReadMe file is also included in each folder which gives the reader all the information required to run the files. Table C.1 lists all the MATLAB m files used along with the corresponding figures that the file can generate.

The bit error rate curves from Chapter 4 take a long time to be generated, as at least a hundred bit errors are collected for each SNR to calculate the error rate. Hence each point on the plots containing BER and MSE in Chapter 4 were obtained by running the associated MATLAB scripts on several computer systems parallely for different SNR settings. Similarly, the Monte Carlo simulations in Chapter 5 are also time consuming and can take several weeks to run especially for higher order channels or higher constellations. Thus, the simulation files associated with Chapter 5 for different SNRs and channel order were run parallely using multiple multi-core computer systems.

All the simulations were mostly performed on five different Linux based systems: one consisting of an Intel I7-5th generation 3.5 GHz CPU with 6 cores and 32 GB memory, one consisting of an Intel I5-4th generation 3.2 GHz CPU with 4 cores and 24 GB memory, one consisting of an AMD Phenom II X6 2.6 GHz CPU with 6 cores and 32 GB memory and two Intel I7-3rd generation 3.2 GHz CPU with 4 cores and 8 GB memory.

Table C.1: MATLAB code files to generate Figures in this thesis.

Chapter	Section	Folder Name	m file to run	Generates Figures
2	2.1.6	Chap2	Main_File1.m	2.4, 2.5
3	3.3.1/3.3.2	Chap3/NumExamples	Main_File2.m	3.1 to 3.24
	3.3.3	Chap3/MIMOGFDM	Main_File3.m	3.25
			Main_File4.m	3.26
	3.4.1	Chap3/MIMO_MAC	Main_File5.m	3.27
			Main_File6.m	3.28
			Main_File7.m	3.29
	3.4.2	Chap3/MIMO_IC	Main_File8.m	3.30
			Main_File9.m	3.31
			Main_File10.m	3.32
			Main_File11.m	3.33, 3.34
4	4.2.1	Chap4/Gaussian	Main_File12.m	4.1
			Main_File13.m	4.2
	4.2.2	Chap4/Tucker	Main_File14.m	4.3
			Main_File15.m	4.4
	4.2.3	Chap4/TTFormat	Main_File16.m	4.5
			Main_File17.m	4.6
	4.2.4	Chap4/MIMOOFDM	Main_File18.m	4.7
			Main_File19.m	4.8, 4.9, 4.10
5	5.4.1/5.4.2	Chap5/Equiprobable	Main_File20.m	5.3, 5.4, 5.14, 5.15
			Main_File21.m	5.5, 5.12
			Main_File22.m	5.6, 5.7,
			Main_File23.m	5.8, 5.9, 5.10, 5.11
			Main_File24.m	5.13
			Main_File25.m	5.16, 5.17 ,5.18, 5.19
	5.4.3	Chap5/Arbitrary_PMF	Main_File26.m	5.20, 5.22
			Main_File27.m	5.21

References

[1] R. Chataut and R. Akl, "Massive MIMO systems for 5G and beyond networks — Overview, recent trends, challenges, and future research direction," *Sensors*, vol. 20, no. 10, p. 2753, 2020.

[2] Y.-N. R. Li, B. Gao, X. Zhang, and K. Huang, "Beam management in millimeter-wave communications for 5G and beyond," *IEEE Access*, vol. 8, pp. 13282–13293, 2020.

[3] M. Aldababsa, M. Toka, S. Gökçeli, G. K. Kurt, and O. Kucur, "A tutorial on nonorthogonal multiple access for 5G and beyond," *Wireless Communications and Mobile Computing*, vol. 2018, 2018.

[4] H. Mathur and T. Deepa, "A survey on advanced multiple access techniques for 5G and beyond wireless communications," *Wireless Personal Communications*, vol. 118, no. 2, pp. 1775–1792, 2021.

[5] Y. Kabalci, "5G mobile communication systems: fundamentals, challenges, and key technologies," in *Smart Grids and Their Communication Systems*, pp. 329–359, Springer, 2019.

[6] H. Tataria, M. Shafi, M. Dohler, and S. Sun, "Six critical challenges for 6G wireless systems: A summary and some solutions," *IEEE Vehicular Technology Magazine*, vol. 17, no. 1, pp. 16–26, 2022.

[7] H. Tataria, M. Shafi, A. F. Molisch, M. Dohler, H. Sjöland, and F. Tufvesson, "6G wireless systems: vision, requirements, challenges, insights, and opportunities," *Proceedings of the IEEE*, vol. 109, no. 7, pp. 1166–1199, 2021.

[8] L. U. Khan, I. Yaqoob, M. Imran, Z. Han, and C. S. Hong, "6G wireless systems: A vision, architectural elements, and future directions," *IEEE Access*, vol. 8, pp. 147029–147044, 2020.

[9] K. David and H. Berndt, "6G vision and requirements: Is there any need for beyond 5G?," *IEEE Vehicular Technology Magazine*, vol. 13, no. 3, pp. 72–80, 2018.

[10] M. Z. Chowdhury, M. Shahjalal, M. Hasan, and Y. M. Jang, "The role of optical wireless communication technologies in 5G/6G and IoT solutions: Prospects, directions, and challenges," *Applied Sciences*, vol. 9, no. 20, p. 4367, 2019.

[11] E. Calvanese Strinati, S. Barbarossa, J. L. Gonzalez-Jimenez, D. Ktenas, N. Cassiau, L. Maret, and C. Dehos, "6G: The next frontier: from holographic messaging to artificial intelligence using subterahertz and visible light communication," *IEEE Vehicular Technology Magazine*, vol. 14, no. 3, pp. 42–50, 2019.

[12] F. Conceição, M. Gomes, V. Silva, R. Dinis, A. Silva, and D. Castanheira, "A survey of candidate waveforms for beyond 5G systems," *Electronics*, vol. 10, no. 1, p. 21, 2021.

[13] L. Hanzo, Y. Akhtman, J. Akhtman, L. Wang, and M. Jiang, *MIMO-OFDM for LTE, WiFi and WiMAX: Coherent versus non-coherent and cooperative turbo transceivers*. John Wiley & Sons, 2010.

[14] "5G NR : Physical channels and modulation (3GPP TS 38.211 version 15.8.0 Release 15) ," tech. rep., European Telecommunications Standards Institute, 2020.

[15] U. Kumar, C. Ibars, A. Bhorkar, and H. Jung, "A waveform for 5G: Guard interval DFT-s-OFDM," in *2015 IEEE Globecom Workshops (GC Wkshps)*, pp. 1–6, Dec 2015.

[16] X. Zhang, L. Chen, J. Qiu, and J. Abdoli, "On the waveform for 5G," *IEEE Communications Magazine*, vol. 54, no. 11, pp. 74–80, 2016.

[17] R. Nissel, S. Schwarz, and M. Rupp, "Filter bank multicarrier modulation schemes for future mobile communications," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 8, pp. 1768–1782, 2017.

[18] N. Michailow, M. Matthé, I. S. Gaspar, A. N. Caldevilla, L. L. Mendes, A. Festag, and G. Fettweis, "Generalized Frequency Division Multiplexing for 5th Generation Cellular Networks," *IEEE Transactions on Communications*, vol. 62, Sept 2014.

[19] K. Cain, V. Vakilian, and R. Abdolee, "Low-complexity universal-filtered multicarrier for beyond 5G wireless systems," in *2018 International Conference on Computing, Networking and Communications (ICNC)*, pp. 254–258, IEEE, 2018.

[20] G. Bochechka, V. Tikhvinskiy, I. Vorozhishchev, A. Aitmagambetov, and B. Nurgozhin, "Comparative analysis of UFMC technology in 5G networks," in *Control and Communications (SIBCON), 2017 International Siberian Conference on*, pp. 1–6, IEEE, 2017.

[21] R. Gerzaguet, N. Bartzoudis, L. G. Baltar, V. Berg, J.-B. Doré, D. Kténas, O. Font-Bach, X. Mestre, M. Payaró, M. Färber, and K. Roth, "The 5G candidate waveform race: a comparison of complexity and performance," *EURASIP Journal on Wireless Communications and Networking*, vol. 2017, p. 13, Jan 2017.

[22] Y. Liu, X. Chen, Z. Zhong, B. Ai, D. Miao, Z. Zhao, J. Sun, Y. Teng, and H. Guan, "Waveform design for 5G networks: Analysis and comparison," *IEEE Access*, vol. 5, pp. 19282–19292, 2017.

[23] J. Gao, O. C. Ozdural, S. H. Ardalan, and H. Liu, "Performance modeling of MIMO OFDM systems via channel analysis," *IEEE Transactions on Wireless Communications*, vol. 5, pp. 2358–2362, September 2006.

[24] A. Stamoulis, S. N. Diggavi, and N. Al-Dhahir, "Intercarrier interference in MIMO OFDM," *IEEE Transactions on Signal Processing*, vol. 50, pp. 2451–2464, Oct 2002.

[25] D. Zhang, L. L. Mendes, M. Matthé, I. S. Gaspar, N. Michailow, and G. P. Fettweis, "Expectation propagation for near-optimum detection of MIMO-GFDM signals," *IEEE Transactions on Wireless Communications*, vol. 15, pp. 1045–1062, Feb 2016.

[26] M. Matthe, I. Gaspar, D. Zhang, and G. Fettweis, "Near-ML detection for MIMO-GFDM," in *2015 IEEE 82nd Vehicular Technology Conference (VTC2015-Fall)*, pp. 1–2, Sept 2015.

[27] M. Caus and A. I. Pérez-Neira, "Multi-stream transmission in MIMO-FBMC systems," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 5041–5045, May 2013.

[28] X. Chen, S. Zhang, and A. Zhang, "On MIMO-UFMC in the presence of phase noise and antenna mutual coupling," *Radio Science*, vol. 52, no. 11, pp. 1386–1394, 2017.

[29] W. Su, Z. Safar, and K. J. R. Liu, "Towards maximum achievable diversity in space, time, and frequency: Performance analysis and code design," *IEEE Transactions on Wireless Communications*, vol. 4, pp. 1847–1857, July 2005.

[30] Z. E. Ankarali, B. Peköz, and H. Arslan, "Flexible radio access beyond 5G: A future projection on waveform, numerology, and frame design principles," *IEEE Access*, vol. 5, pp. 18295–18309, 2017.

[31] L. Dai, B. Wang, Y. Yuan, S. Han, I. Chih-Lin, and Z. Wang, "Non-orthogonal multiple access for 5G: solutions, challenges, opportunities, and future research trends," *IEEE Communications Magazine*, vol. 53, no. 9, pp. 74–81, 2015.

[32] T. G. Kolda and B. W. Bader, “Tensor decompositions and applications,” *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.

[33] P. Comon, “Tensors : A brief introduction,” *IEEE Signal Processing Magazine*, vol. 31, pp. 44–53, May 2014.

[34] L. R. Tucker, “The extension of factor analysis to three-dimensional matrices,” in *Contributions to mathematical psychology.* (H. Gulliksen and N. Frederiksen, eds.), pp. 110–127, New York: Holt, Rinehart and Winston, 1964.

[35] C. J. Appellof and E. R. Davidson, “Strategies for analyzing data from video fluorometric monitoring of liquid chromatographic effluents,” *Analytical Chemistry*, vol. 53, no. 13, pp. 2053–2056, 1981.

[36] R. Bro, “Review on multiway analysis in Chemistry—2000–2005,” *Critical reviews in analytical chemistry*, vol. 36, no. 3-4, pp. 279–293, 2006.

[37] L. Yu, D. Zhang, N. Liu, and W. Zhou, “A multi-view fusion method via tensor learning and gradient descent for image features,” *IEEE Access*, vol. 9, pp. 79389–79399, 2021.

[38] J. J. Guerrero, A. C. Murillo, and C. SagÜÉs, “Localization and matching using the planar trifocal tensor with bearing-only data,” *IEEE Transactions on Robotics*, vol. 24, no. 2, pp. 494–501, 2008.

[39] X. Li, M. K. Ng, and Y. Ye, “MultiComm: Finding community structure in multi-dimensional networks,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, pp. 929–941, April 2014.

[40] E. E. Papalexakis, C. Faloutsos, and N. D. Sidiropoulos, “Tensors for data mining and data fusion: Models, applications, and scalable algorithms,” *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 8, no. 2, pp. 1–44, 2017.

[41] N. D. Sidiropoulos, L. D. Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, “Tensor decomposition for signal processing and machine learning,” *IEEE Transactions on Signal Processing*, vol. 65, pp. 3551–3582, July 2017.

[42] C. Chatzichristos, E. Kofidis, M. Morante, and S. Theodoridis, “Blind fMRI source unmixing via higher-order tensor decompositions,” *Journal of Neuroscience Methods*, vol. 315, pp. 17–47, 2019.

[43] A. Cichocki, D. Mandic, L. De Lathauwer, G. Zhou, Q. Zhao, C. Caiafa, and H. A. PHAN, “Tensor decompositions for signal processing applications: From two-way to multiway component analysis,” *IEEE Signal Processing Magazine*, vol. 32, no. 2, pp. 145–163, 2015.

[44] P. R. Gomes, J. P. C. da Costa, A. L. de Almeida, and R. T. de Sousa, "Tensor-based multiple denoising via successive spatial smoothing, low-rank approximation and reconstruction for R-D sensor array processing," *Digital Signal Processing*, vol. 89, pp. 1–7, 2019.

[45] C. Chen, A. Surana, A. Bloch, and I. Rajapakse, "Multilinear time invariant system theory," in *2019 Proceedings of the Conference on Control and its Applications*, pp. 118–125, 2019.

[46] C. Chen, A. Surana, A. M. Bloch, and I. Rajapakse, "Multilinear control systems theory," *SIAM Journal on Control and Optimization*, vol. 59, no. 1, pp. 749–776, 2021.

[47] B. W. Bader and T. G. Kolda, "Algorithm 862: MATLAB tensor classes for fast algorithm prototyping," *ACM Transactions on Mathematical Software*, vol. 32, pp. 635–653, December 2006.

[48] M. Brazell, N. Li, C. Navasca, and C. Tamon, "Solving Multilinear Systems via Tensor Inversion," *SIAM Journal on Matrix Analysis and Applications*, vol. 34, no. 2, pp. 542–570, 2013.

[49] I. Kisil, G. G. Calvi, B. S. Dees, and D. P. Mandic, "Tensor decompositions and practical applications: A hands-on tutorial," in *Recent Trends in Learning From Data*, pp. 69–97, Springer, 2020.

[50] H. Chen, F. Ahmad, S. Vorobyov, and F. Porikli, "Tensor decompositions in wireless communications and MIMO radar," *IEEE Journal of Selected Topics in Signal Processing*, vol. 15, no. 3, pp. 438–453, 2021.

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

[52] R. A. Harshman, "Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-modal factor analysis," *UCLA Working Papers in Phonetics*, vol. 16, no. 1, p. 84, 1970.

[53] N. D. Sidiropoulos and R. Bro, "On the uniqueness of multilinear decomposition of N-way arrays," *Journal of Chemometrics*, vol. 14, no. 3, pp. 229–239, 2000.

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

[55] L. De Lathauwer and J. Castaing, "Tensor-based techniques for the blind separation of DS-CDMA signals," *Signal Processing*, vol. 87, no. 2, pp. 322–336, 2007.

[56] L. R. Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika*, vol. 31, pp. 279–311, Sep 1966.

[57] A. Cichocki, "Era of big data processing: A new approach via tensor networks and tensor decompositions," *arXiv preprint arXiv:1403.2048*, 2014.

[58] R. Bro, R. A. Harshman, N. D. Sidiropoulos, and M. E. Lundy, "Modeling multi-way data with linearly dependent loadings," *Journal of Chemometrics: A Journal of the Chemometrics Society*, vol. 23, no. 7-8, pp. 324–340, 2009.

[59] R. A. Harshman and M. E. Lundy, "Uniqueness proof for a family of models sharing features of Tucker's three-mode factor analysis and PARAFAC/CANDECOMP," *Psychometrika*, vol. 61, no. 1, pp. 133–154, 1996.

[60] A. L. De Almeida, G. Favier, and L. R. Ximenes, "Space-Time-Frequency (STF) MIMO communication systems with blind receiver based on a generalized PARATUCK2 model," *IEEE Transactions on Signal Processing*, vol. 61, no. 8, pp. 1895–1909, 2013.

[61] G. Favier and A. L. de Almeida, "Overview of constrained PARAFAC models," *EURASIP Journal on Advances in Signal Processing*, vol. 2014, no. 1, p. 142, 2014.

[62] I. V. Oseledets, "Tensor-train decomposition," *SIAM Journal on Scientific Computing*, vol. 33, no. 5, pp. 2295–2317, 2011.

[63] L.-H. Lim, "Singular values and eigenvalues of tensors: A variational approach," in *1st IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing, 2005.*, pp. 129–132, IEEE, 2005.

[64] S. Weiland and F. Van Belzen, "Singular value decompositions and low rank approximations of tensors," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, p. 1171, 2010.

[65] L. D. Lathauwer, B. D. Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM J. Matrix Anal. Appl.*, vol. 21, pp. 1253–1278, Mar. 2000.

[66] L. Qi, "Eigenvalues of a real supersymmetric tensor," *Journal of Symbolic Computation*, vol. 40, pp. 1302–1324, 2005.

[67] L. Qi, H. Chen, and Y. Chen, *Tensor eigenvalues and their applications*, vol. 39. Springer, 2018.

[68] L. Sun, B. Zheng, C. Bu, and Y. Wei, "Moore–Penrose inverse of tensors via Einstein product," *Linear and Multilinear Algebra*, vol. 64, no. 4, pp. 686–698, 2016.

[69] L.-B. Cui, C. Chen, W. Li, and M. K. Ng, "An eigenvalue problem for even order tensors with its applications," *Linear and Multilinear Algebra*, vol. 64, no. 4, pp. 602–621, 2016.

[70] M. lin Liang, B. Zheng, and R. juan Zhao, "Tensor inversion and its application to the tensor equations with Einstein product," *Linear and Multilinear Algebra*, vol. 67, no. 4, pp. 843–870, 2019.

[71] N. D. Sidiropoulos, G. B. Giannakis, and R. Bro, "Blind PARAFAC receivers for DS-CDMA systems," *IEEE Transactions on Signal Processing*, vol. 48, no. 3, pp. 810–823, 2000.

[72] D. Nion and L. De Lathauwer, "Blind receivers based on tensor decompositions. Application in DS-CDMA and over-sampled systems," in *Conference Record of the Forty-First Asilomar Conference on Signals, Systems and Computers, (ACSSC) 2007*, pp. 403–407, IEEE, 2007.

[73] C. A. Fernandes, G. Favier, and J. C. Mota, "PARAFAC-based channel estimation and data recovery in nonlinear MIMO spread spectrum communication systems," *Signal Processing*, vol. 91, no. 2, pp. 311–322, 2011.

[74] G. Favier and A. L. F. de Almeida, "Tensor space-time-frequency coding with semi-blind receivers for MIMO wireless communication systems," *IEEE Transactions on Signal Processing*, vol. 62, pp. 5987–6002, Nov 2014.

[75] M. N. da Costa, G. Favier, and J. M. T. Romano, "Tensor modelling of MIMO communication systems with performance analysis and Kronecker receivers," *Signal Processing*, vol. 145, pp. 304–316, 2018.

[76] F. Roemer and M. Haardt, "Tensor-based channel estimation (TENCE) for two-way relaying with multiple antennas and spatial reuse," in *IEEE International Conference on Acoustics, Speech and Signal Processing, (ICASSP) 2009.*, pp. 3641–3644, IEEE, 2009.

[77] X. Han, A. L. de Almeida, and Z. Yang, "Channel estimation for MIMO multi-relay systems using a tensor approach," *EURASIP Journal on Advances in Signal Processing*, vol. 2014, no. 1, p. 163, 2014.

[78] G. T. de Araújo, A. L. F. de Almeida, and R. Boyer, "Channel estimation for intelligent reflecting surface assisted MIMO systems: A tensor modeling approach," *IEEE Journal of Selected Topics in Signal Processing*, vol. 15, no. 3, pp. 789–802, 2021.

[79] C. Qian, X. Fu, N. D. Sidiropoulos, and Y. Yang, "Tensor-based channel estimation for dual-polarized massive MIMO Systems," *IEEE Transactions on Signal Processing*, vol. 66, no. 24, pp. 6390–6403, 2018.

[80] D. C. Araújo, A. L. De Almeida, J. P. Da Costa, and R. T. de Sousa, "Tensor-based channel estimation for massive MIMO-OFDM systems," *IEEE Access*, vol. 7, pp. 42133–42147, 2019.

[81] Z. Lin, T. Lv, J. A. Zhang, and R. P. Liu, "Tensor-based high-accuracy position estimation for 5G mmWave massive MIMO systems," in *ICC 2020 - IEEE International Conference on Communications (ICC) held virtually between 7-11 June*, pp. 1–6, 2020.

[82] Y. Lin, S. Jin, M. Matthaiou, and X. You, "Tensor-based channel estimation for millimeter wave MIMO-OFDM with dual-wideband effects," *IEEE Transactions on Communications*, vol. 68, no. 7, pp. 4218–4232, 2020.

[83] J. Du, M. Han, L. Jin, Y. Hua, and X. Li, "Semi-blind receivers for multi-user massive MIMO relay systems based on block Tucker2-PARAFAC tensor model," *IEEE Access*, vol. 8, pp. 32170–32186, 2020.

[84] A. Venugopal and H. Leib, "A tensor based framework for multi-domain communication systems," *IEEE Open Journal of the Communications Society*, vol. 1, pp. 606–633, 2020.

[85] Y. Zniyed, R. Boyer, A. L. F. de Almeida, and G. Favier, "Tensor-train modeling for MIMO OFDM tensor coding-and-forwarding relay systems," in *2019 27th European Signal Processing Conference (EUSIPCO)*, pp. 1–5, 2019.

[86] A. L. F. De Almeida, *Tensor modeling and signal processing for wireless communication systems.* Theses, Université de Nice Sophia Antipolis, Nov. 2007.

[87] C. Buiquang, Z. Ye, J. Dai, and Y. A. Sheikh, "CFO robust blind receivers for MIMO-OFDM systems based on PARALIND factorizations," *Digital Signal Processing*, vol. 69, pp. 337–349, 2017.

[88] A. L. F. de Almeida, G. Favier, J. da Costa, and J. C. M. Mota, "Overview of tensor decompositions with applications to communications," *Signals and Images: Advances and results in speech, estimation, compression, recognition, filtering, and processing*, pp. 325–356, 2016.

[89] P. Comon, "Tensors versus matrices usefulness and unexpected properties," in *2009 IEEE/SP 15th Workshop Statistical Signal Process.*, pp. 781–788, IEEE, 2009.

[90] I. E. Telatar, "Capacity of multi-antenna Gaussian channels," *European Transactions on Telecommunications*, vol. 10, pp. 585–595, 1999.

[91] G. J. Foschini and M. J. Gans, "On limits of wireless communications in a fading environment when using multiple antennas," *Wireless personal communications*, vol. 6, no. 3, pp. 311–335, 1998.

[92] E. Visotsky and U. Madhow, "Space-time transmit precoding with imperfect feedback," *IEEE Transactions on Information Theory*, vol. 47, no. 6, pp. 2632–2639, 2001.

[93] S. A. Jafar, S. Vishwanath, and A. Goldsmith, "Channel capacity and beamforming for multiple transmit and receive antennas with covariance feedback," in *ICC 2001. IEEE International Conference on Communications. Conference Record (Cat. No.01CH37240)*, vol. 7, pp. 2266–2270 vol.7, 2001.

[94] E. A. Jorswieck and H. Boche, "Channel capacity and capacity-range of beamforming in MIMO wireless systems under correlated fading with covariance feedback," *IEEE Transactions on Wireless Communications*, vol. 3, pp. 1543–1553, Sept 2004.

[95] L. Hanlen and A. Grant, "Capacity analysis of correlated MIMO channels," *IEEE Transactions on Information Theory*, vol. 58, pp. 6773–6787, Nov 2012.

[96] T. L. Marzetta and B. M. Hochwald, "Capacity of a mobile multiple-antenna communication link in Rayleigh flat fading," *IEEE Transactions on Information Theory*, vol. 45, pp. 139–157, Jan 1999.

[97] S. A. Jafar and A. Goldsmith, "Multiple-antenna capacity in correlated Rayleigh fading with channel covariance information," *IEEE Transactions on Wireless Communications*, vol. 4, pp. 990–997, May 2005.

[98] H. Bolcskei, D. Gesbert, and A. J. Paulraj, "On the capacity of OFDM-based spatial multiplexing systems," *IEEE Transactions on Communications*, vol. 50, pp. 225–234, Feb 2002.

[99] F. Dupuy and P. Loubaton, "On the capacity achieving covariance matrix for frequency selective MIMO channels using the asymptotic approach," *IEEE Transactions on Information Theory*, vol. 57, pp. 5737–5753, Sept 2011.

[100] M. Vu, "MISO capacity with per-antenna power constraint," *IEEE Transactions on Communications*, vol. 59, no. 5, pp. 1268–1274, 2011.

[101] M. Vu, "MIMO capacity with per-antenna power constraint," in *2011 IEEE Global Telecommunications Conference - GLOBECOM 2011*, pp. 1–5, Dec 2011.

[102] C. Xing, Z. Fei, Y. Zhou, and Z. Pan, "Matrix-field water-filling architecture for MIMO transceiver designs with mixed power constraints," in *2015 IEEE 26th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pp. 392–396, Aug 2015.

[103] F. D. Neeser and J. L. Massey, "Proper complex random processes with applications to information theory," *IEEE Transactions on Information Theory*, vol. 39, pp. 1293–1302, Jul 1993.

[104] D. Guo, S. Shamai, and S. Verdu, "Mutual information and minimum mean-square error in Gaussian channels," *IEEE Transactions on Information Theory*, vol. 51, pp. 1261–1282, April 2005.

[105] A. Lozano, A. M. Tulino, and S. Verdu, "Optimum power allocation for parallel Gaussian channels with arbitrary input distributions," *IEEE Transactions on Information Theory*, vol. 52, pp. 3033–3051, July 2006.

[106] F. Perez-Cruz, M. R. D. Rodrigues, and S. Verdu, "MIMO Gaussian channels with arbitrary inputs: Optimal precoding and power allocation," *IEEE Transactions on Information Theory*, vol. 56, pp. 1070–1084, March 2010.

[107] D. P. Palomar and S. Verdu, "Gradient of mutual information in linear vector Gaussian channels," *IEEE Transactions on Information Theory*, vol. 52, pp. 141–154, Jan 2006.

[108] C. Xiao, Y. R. Zheng, and Z. Ding, "Globally optimal linear precoders for finite alphabet signals over complex vector Gaussian channels," *IEEE Transactions on Signal Processing*, vol. 59, no. 7, pp. 3301–3314, 2011.

[109] M. R. Rodrigues, F. Pérez-Cruz, and S. Verdu, "Multiple-input multiple-output Gaussian channels: Optimal covariance for non-Gaussian inputs," in *2008 IEEE Information Theory Workshop*, pp. 445–449, IEEE, 2008.

[110] W. Cao, A. Dytso, M. Fauss, G. Feng, and H. V. Poor, "Robust waterfilling for approximately Gaussian inputs," in *2019 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, 2019.

[111] J. Feng, B. Feng, Y. Wu, L. Shen, and W. Zhang, "MIMO transmission under discrete input signal constraints," in *2021 IEEE/CIC International Conference on Communications in China (ICCC)*, pp. 723–728, 2021.

[112] Y. Wu, C. Xiao, Z. Ding, X. Gao, and S. Jin, "A survey on MIMO transmission with finite input signals: Technical challenges, advances, and future Trends," *Proceedings of the IEEE*, no. 99, pp. 1–55, 2018.

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

[114] Q.-W. Wang and X. Xu, "Iterative algorithms for solving some tensor equations," *Linear and Multilinear Algebra*, vol. 67, no. 7, pp. 1325–1349, 2019.

[115] R. Pan, "Tensor transpose and its properties," *arXiv preprint arXiv:1411.1503*, 2014.

[116] A. Zhang and D. Xia, "Tensor SVD: Statistical and computational limits," *IEEE Transactions on Information Theory*, vol. 64, no. 11, pp. 7311–7338, 2018.

[117] Z. Luo, L. Qi, and P. L. Toint, "Bernstein concentration inequalities for tensors via Einstein products," *arXiv preprint arXiv:1902.03056*, 2019.

[118] S. Hu, Z.-H. Huang, C. Ling, and L. Qi, "On determinants and eigenvalue theory of tensors," *Journal of Symbolic Computation*, vol. 50, pp. 508–531, 2013.

[119] K. B. Petersen and M. S. Pedersen, "The matrix cookbook," nov 2012. Version 20121115.

[120] H. Liu, L. T. Yang, J. Ding, Y. Guo, and S. S. Yau, "Tensor-train-based high-order dominant eigen decomposition for multimodal prediction services," *IEEE Transactions on Engineering Management*, vol. 68, no. 1, pp. 197–211, 2021.

[121] T. Adali, P. J. Schreier, and L. L. Scharf, "Complex-valued signal processing: the proper way to deal with impropriety," *IEEE Transactions on Signal Processing*, vol. 59, pp. 5101–5125, Nov 2011.

[122] P. J. Schreier and L. L. Scharf, *Statistical signal processing of complex-valued data: the theory of improper and noncircular signals.* Cambridge University Press, 2010.

[123] A. van den Bos, "The multivariate complex normal distribution-a generalization," *IEEE Transactions on Information Theory*, vol. 41, no. 2, pp. 537–539, 1995.

[124] H. L. Van Trees, *Optimum array processing: Part IV of detection, estimation, and modulation theory.* John Wiley & Sons, 2004.

[125] D. Messerschmitt, "Stationary points of a real-valued function of a complex variable," *EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2006-93*, 2006.

[126] A. Feiten, S. Hanly, and R. Mathar, "Derivatives of mutual information in Gaussian vector channels with applications," in *2007 IEEE International Symposium on Information Theory*, pp. 2296–2300, IEEE, 2007.

[127] B. Huang and C. Ma, "An iterative algorithm to solve the generalized Sylvester tensor equations," *Linear and Multilinear Algebra*, pp. 1–26, 2018.

[128] X.-F. Duan, C.-Y. Wang, and C.-M. Li, "Newton's method for solving the tensor square root problem," *Applied Mathematics Letters*, vol. 98, pp. 57–62, 2019.

[129] B. Huang and W. Li, "Numerical subspace algorithms for solving the tensor equations involving Einstein product," *Numerical Linear Algebra with Applications*, vol. 28, no. 2, p. e2351, 2021.

[130] R. Behera and D. Mishra, "Further results on generalized inverses of tensors via the Einstein product," *Linear and Multilinear Algebra*, vol. 65, no. 8, pp. 1662–1682, 2017.

[131] K. Panigrahy and D. Mishra, "Extension of Moore–Penrose inverse of tensor via Einstein product," *Linear and Multilinear Algebra*, vol. 0, no. 0, pp. 1–24, 2020.

[132] B. Huang, "Numerical study on Moore-Penrose inverse of tensors via Einstein product," *Numerical Algorithms*, vol. 87, pp. 1767–1797, 2021.

[133] J. K. Sahoo and R. Behera, "Reverse-order law for core inverse of tensors," *Computational and Applied Mathematics*, vol. 39, no. 2, pp. 1–22, 2020.

[134] V. Pan and R. Schreiber, "An improved Newton iteration for the generalized inverse of a matrix, with applications," *SIAM Journal on Scientific and Statistical Computing*, vol. 12, no. 5, pp. 1109–1130, 1991.

[135] A. Ben-Israel, "A note on an iterative method for generalized inversion of matrices," *Mathematics of Computation*, vol. 20, no. 95, pp. 439–440, 1966.

[136] Y. Wang and H. Leib, "Sphere decoding for MIMO systems with Newton iterative matrix inversion," *IEEE Communications Letters*, vol. 17, no. 2, pp. 389–392, 2013.

[137] H. Wolkowicz and G. P. Styan, "Bounds for eigenvalues using traces," *Linear algebra and its applications*, vol. 29, pp. 471–506, 1980.

[138] P. Singh and A. Manure, *Introduction to TensorFlow 2.0*, pp. 1–24. Berkeley, CA: Apress, 2020.

[139] N. Ketkar and J. Moolayil, "Introduction to PyTorch," in *Deep learning with python*, pp. 27–91, Springer, 2021.

[140] F. Hlawatsch and G. Matz, *Wireless communications over rapidly time-varying channels*. Academic press, 2011.

[141] N. Costa and S. Haykin, "A novel wideband MIMO channel model and experimental validation," *IEEE Transactions on Antennas and Propagation*, vol. 56, pp. 550–562, Feb 2008.

[142] Y. J. Zhang and K. B. Letaief, "An efficient resource-allocation scheme for spatial multiuser access in MIMO/OFDM systems," *IEEE Transactions on Communications*, vol. 53, pp. 107–116, Jan 2005.

[143] Y. Tan and Q. Chang, "Multi-user MIMO-OFDM with adaptive resource allocation over frequency selective fading channel," in *2008 4th International Conference on Wireless Communications, Networking and Mobile Computing : Dalian, China*, pp. 1–5, Oct 2008.

[144] A. Tolli and M. Juntti, "Efficient user, bit and power allocation for adaptive multiuser MIMO-OFDM with low signalling overhead," in *2006 IEEE International Conference on Communications : Istanbul, Turkey*, vol. 12, pp. 5360–5365, IEEE, 2006.

[145] H. J. Yang, W.-Y. Shin, B. C. Jung, C. Suh, and A. Paulraj, "Opportunistic downlink interference alignment for multi-cell MIMO networks," *IEEE Transactions on Wireless Communications*, vol. 16, no. 3, pp. 1533–1548, 2017.

[146] D. A. Feryando, T. Suryani, and Endroyono, "Performance analysis of regularized channel inversion precoding in multiuser MIMO-GFDM downlink systems," in *2017 IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob)*, pp. 101–105, IEEE, 2017.

[147] M. Yuzgeccioglu and E. Jorswieck, "Transceiver design for GFDM with index modulation in multi-user networks," in *WSA 2018; 22nd International ITG Workshop on Smart Antennas*, pp. 1–4, VDE, 2018.

[148] C. Leung, S. Huberman, K. Ho-Van, and T. Le-Ngoc, "Vectored DSL: Potential, implementation issues and challenges," *IEEE Communications Surveys Tutorials*, vol. 15, no. 4, pp. 1907–1923, 2013.

[149] S. Serbetli and A. Yener, "MIMO-CDMA systems: signature and beamformer design with various levels of feedback," *IEEE Transactions on Signal Processing*, vol. 54, no. 7, pp. 2758–2772, 2006.

[150] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge university press, 2004.

[151] C. Xing, Y. Jing, S. Wang, J. Wang, and J. An, "A general framework for covariance matrix optimization in MIMO systems," *arXiv preprint arXiv:1711.04449v8*, 2018.

[152] M. Grant and S. Boyd, "CVX: Matlab Software for Disciplined Convex Programming, version 2.1 (software and webpage)." http://cvxr.com/cvx, Mar. 2014.

[153] A. Fang, L. Cui, Z. Zhang, C. Chen, and Z. Sheng, "A parallel computing framework for cloud services," in *2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA)*, pp. 832–835, IEEE, 2020.

[154] A. M. Mahmood, A. Al-Yasiri, and O. Y. Alani, "A new processing approach for reducing computational complexity in cloud-RAN mobile networks," *IEEE Access*, vol. 6, pp. 6927–6946, 2017.

[155] C. Fan, Y. J. Zhang, and X. Yuan, "Dynamic nested clustering for parallel PHY-layer processing in cloud-RANs," *IEEE Transactions on Wireless Communications*, vol. 15, no. 3, pp. 1881–1894, 2015.

[156] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. New York, NY, USA: Cambridge University Press, 2005.

[157] B. M. Hochwald and S. Ten Brink, "Achieving near-capacity on a multiple-antenna channel," *IEEE Transactions on Communications*, vol. 51, no. 3, pp. 389–399, 2003.

[158] A. Burr, "Capacity bounds and estimates for the finite scatterers MIMO wireless channel," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 5, pp. 812–818, 2003.

[159] S. Loyka and G. Levin, "On physically-based normalization of MIMO channel matrices," *IEEE Transactions on Wireless Communications*, vol. 8, no. 3, pp. 1107–1112, 2009.

[160] G. Favier, C. A. R. Fernandes, and A. L. de Almeida, "Nested tucker tensor decomposition with application to MIMO relay systems using tensor space–time coding (TSTC)," *Signal Processing*, vol. 128, pp. 318–331, 2016.

[161] C. F. Caiafa and A. Cichocki, "Block sparse representations of tensors using Kronecker bases," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2709–2712, 2012.

[162] H. Zhang and F. Ding, "On the Kronecker products and their applications," *Journal of Applied Mathematics*, vol. 2013, 2013.

[163] P. D. Hoff, "Separable covariance arrays via the Tucker product, with applications to multivariate relational data," *Bayesian Analysis*, vol. 6, no. 2, pp. 179–196, 2011.

[164] N. Costa and S. Haykin, *Multiple-input multiple-output channel models: Theory and practice*. John Wiley, 2010.

[165] D.-S. Shiu, G. Foschini, M. Gans, and J. Kahn, "Fading correlation and its effect on the capacity of multielement antenna systems," *IEEE Transactions on Communications*, vol. 48, no. 3, pp. 502–513, 2000.

[166] H. He, C.-K. Wen, S. Jin, and G. Y. Li, "A model-driven deep learning network for MIMO detection," in *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 584–588, IEEE, 2018.

[167] G. Yang, H. Zhang, Z. Shi, S. Ma, and H. Wang, "Asymptotic outage analysis of spatially correlated Rayleigh MIMO channels," *IEEE Transactions on Broadcasting*, vol. 67, no. 1, pp. 263–278, 2021.

[168] H. Shin, M. Z. Win, J. H. Lee, and M. Chiani, "On the capacity of doubly correlated MIMO channels," *IEEE Transactions on Wireless Communications*, vol. 5, no. 8, pp. 2253–2265, 2006.

[169] M. Matthé, N. Michailow, I. Gaspar, and G. Fettweis, "Influence of pulse shaping on bit error rate performance and out of band radiation of Generalized Frequency Division Multiplexing," in *2014 IEEE International Conference on Communications Workshops (ICC)*, pp. 43–48, 2014.

[170] A. Goldsmith, S. A. Jafar, N. Jindal, and S. Vishwanath, "Capacity limits of MIMO channels," *IEEE Journal on selected areas in Communications*, vol. 21, no. 5, pp. 684–702, 2003.

[171] Wei Yu, Wonjong Rhee, S. Boyd, and J. M. Cioffi, "Iterative water-filling for Gaussian vector multiple-access channels," *IEEE Transactions on Information Theory*, vol. 50, no. 1, pp. 145–152, 2004.

[172] M. Kiamari and A. S. Avestimehr, "Capacity region of the symmetric injective K-user deterministic interference channel," *IEEE Transactions on Information Theory*, vol. 65, no. 7, pp. 4010–4022, 2019.

[173] X. Shang, B. Chen, G. Kramer, and H. V. Poor, "Capacity regions and sum-rate capacities of vector Gaussian interference channels," *IEEE Transactions on Information Theory*, vol. 56, no. 10, pp. 5030–5044, 2010.

[174] X. Shang, B. Chen, and M. J. Gans, "On the achievable sum rate for MIMO interference channels," *IEEE Transactions on Information Theory*, vol. 52, no. 9, pp. 4313–4320, 2006.

[175] X. Shang, B. Chen, G. Kramer, and H. V. Poor, "On the capacity of MIMO interference channels," in *2008 46th Annual Allerton Conference on Communication, Control, and Computing*, pp. 700–707, IEEE, 2008.

[176] X. Shang and H. V. Poor, "Noisy-interference sum-rate capacity for vector Gaussian interference channels," *IEEE Transactions on Information Theory*, vol. 59, no. 1, pp. 132–153, 2013.

[177] X. Shang and H. V. Poor, "Capacity region of vector Gaussian interference channels with generally strong interference," *IEEE Transactions on Information Theory*, vol. 58, no. 6, pp. 3472–3496, 2012.

[178] C. Geng, N. Naderializadeh, A. S. Avestimehr, and S. A. Jafar, "On the optimality of treating interference as noise," *IEEE Transactions on Information Theory*, vol. 61, no. 4, pp. 1753–1767, 2015.

[179] A. Ghasemi, A. S. Motahari, and A. K. Khandani, "Interference alignment for the K user MIMO interference channel," in *2010 IEEE International Symposium on Information Theory*, pp. 360–364, 2010.

[180] J. Marot, C. Fossati, and S. Bourennane, "About advances in tensor data denoising methods," *EURASIP Journal on Advances in Signal Processing*, vol. 2008, no. 1, p. 235357, 2008.

[181] R. Tong, G. Bao, and Z. Ye, "A higher order subspace algorithm for multichannel speech enhancement," *IEEE Signal Processing Letters*, vol. 22, no. 11, pp. 2004–2008, 2015.

[182] T. Lin and S. Bourennane, "Survey of hyperspectral image denoising methods based on tensor decompositions," *EURASIP journal on Advances in Signal Processing*, vol. 2013, no. 1, pp. 1–11, 2013.

[183] T. Yokota, Q. Zhao, and A. Cichocki, "Smooth PARAFAC decomposition for tensor completion," *IEEE Transactions on Signal Processing*, vol. 64, no. 20, pp. 5423–5436, 2016.

[184] S. Lang, *Introduction to linear algebra.* Springer Science & Business Media, 2012.

[185] B. Picinbono and P. Chevalier, "Widely linear estimation with complex data," *IEEE Transactions on Signal Processing*, vol. 43, pp. 2030–2033, 1995.

[186] D. Muti, S. Bourennane, and J. Marot, "Lower-rank tensor approximation and multiway filtering," *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 3, pp. 1172–1204, 2008.

[187] D. Muti and S. Bourennane, "Multidimensional estimation based on a tensor decomposition," in *IEEE Workshop on Statistical Signal Processing, 2003*, pp. 98–101, IEEE, 2003.

[188] A. Burg, S. Haene, D. Perels, P. Luethi, N. Felber, and W. Fichtner, "Algorithm and VLSI architecture for linear MMSE detection in MIMO-OFDM systems," in *IEEE International Symposium on Circuits and Systems*, IEEE, 2006.

[189] D. N. Liu and M. P. Fitz, "Low complexity affine MMSE detector for iterative detection-decoding MIMO OFDM systems," *IEEE Transactions on Communications*, vol. 56, no. 1, pp. 150–158, 2008.

[190] M. Wu, B. Yin, G. Wang, C. Dick, J. R. Cavallaro, and C. Studer, "Large-scale MIMO detection for 3GPP LTE: Algorithms and FPGA implementations," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 5, pp. 916–929, 2014.

[191] W. Yi and H. Leib, "OFDM symbol detection integrated with channel multipath gains estimation for doubly-selective fading channels," *Physical communication*, vol. 22, pp. 19–31, 2017.

[192] K. A. D. Teo and S. Ohno, "Pilot-aided channel estimation and Viterbi equalization for OFDM over doubly-selective channel," in *IEEE Globecom 2006*, pp. 1–5, Nov 2006.

[193] A. Alimohammad, S. F. Fard, B. F. Cockburn, and C. Schlegel, "An improved SOS-based fading channel emulator," in *2007 IEEE 66th Vehicular Technology Conference*, pp. 931–935, IEEE, 2007.

[194] P. Mukunthan and P. Dananjayan, "PAPR reduction based on a modified PTS with interleaving and pulse shaping method for STBC MIMO-OFDM system," in *2012 Third International Conference on Computing, Communication and Networking Technologies (ICCCNT'12)*, pp. 1–6, IEEE, 2012.

[195] Z.-Q. Luo and W. Yu, "An introduction to convex optimization for communications and signal processing," *IEEE Journal on selected areas in communications*, vol. 24, no. 8, pp. 1426–1438, 2006.

[196] F. Pérez-Cruz, M. R. Rodrigues, and S. Verdú, "Generalized mercury/waterfilling for multiple-input multiple-output channels," in *45th Allerton Conference on Communication, Control, and Computing*, 2007.

[197] T. Cover and A. Thomas, "Determinant inequalities via Information Theory," *SIAM Journal on Matrix Analysis and Applications*, vol. 9, no. 3, pp. 384–392, 1988.

[198] H. H. Andersen, M. Hojbjerre, D. Sorensen, and P. S. Eriksen, *Linear and graphical models: for the multivariate complex normal distribution*, vol. 101. Springer Science & Business Media, 1995.

[199] B. Picinbono, “Second-order complex random vectors and normal distributions,” *IEEE Transactions on Signal Processing*, vol. 44, pp. 2637–2640, Oct 1996.

[200] G. R. Ducharme, P. L. De Micheaux, and B. Marchina, “The complex multinormal distribution, quadratic forms in complex random vectors and an omnibus goodness-of-fit test for the complex normal distribution,” *Annals of the Institute of Statistical Mathematics*, vol. 68, no. 1, pp. 77–104, 2016.

[201] H. Wolkowicz and G. P. Styan, “Extensions of Samuelson’s inequality,” *The American Statistician*, vol. 33, no. 3, pp. 143–144, 1979.

[202] C.-Y. Hsu and W.-R. Wu, “Low-complexity ICI mitigation methods for high-mobility SISO/MIMO-OFDM systems,” *IEEE Transactions on Vehicular Technology*, vol. 58, no. 6, pp. 2755–2768, 2009.

[203] P. Billingsley, *Probablity and Measure, 3rd ed.* John Wiley & Sons, 1995.

[204] A. Guzman, *Derivatives and integrals of multivariable functions.* Springer Science & Business Media, 2003.