論文 / 著書情報
Article / Book Information

題目(和文)	
Title(English)	A Study of Algebraic Real Translations of Cayley-Dickson Linear Systems and Signal Processing Applications
著者(和文)	溝口毅彦
Author(English)	Takehiko Mizoguchi
出典(和文)	学位:博士(工学), 学位授与機関:東京工業大学, 報告番号:甲第11256号, 授与年月日:2019年9月20日, 学位の種別:課程博士, 審査員:山田 功,植松 友彦,府川 和彦,山口 雅浩,笠井 健太,湯川 正裕
Citation(English)	Degree:Doctor (Engineering), Conferring organization: Tokyo Institute of Technology, Report number:甲第11256号, Conferred date:2019/9/20, Degree Type:Course doctor, Examiner:,,,,,
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

A Study of Algebraic Real Translations of Cayley-Dickson Linear Systems and Signal Processing Applications

Takehiko Mizoguchi

July 10, 2019
Version: Third Edition

A Study of Algebraic Real Translations of Cayley-Dickson Linear Systems and Signal Processing Applications

Takehiko Mizoguchi

Tokyo Institute of Technology

Department of Communications and Computer Engineering
Graduate School of Science and Engineering

A thesis submitted in partial fulfillment of the requirements for the degree of Doctor of Engineering.

Supervisor Prof. Isao Yamada

July 10, 2019

Takehiko Mizoguchi
A Study of Algebraic Real Translations of Cayley-Dickson Linear Systems and Signal Processing Applications
July 10, 2019
Supervisor: Prof. Isao Yamada

Tokyo Institute of Technology
Graduate School of Science and Engineering
Department of Communications and Computer Engineering
S3-60 O-okayama, Meguro-ku
Tokyo 152-8552, Japan

Abstract

In this thesis, we enhance unified framework for optimization/learning and eigenvalue problems in hypercomplex domain. We first present a systematic algebraic translation of Cayley-Dickson (C-D) hypercomplex valued linear systems into a real vector valued linear model. This translation is designed by using jointly two new isomorphisms between real vector spaces. As an example of many potential algorithms through the proposed algebraic translation, we present $\mathbb{A}_m$-adaptive projected subgradient method ($\mathbb{A}_m$-APSM). We next propose novel definitions of singular value decomposition and low rank approximation in C-D domain. We then derive algorithmic solutions to hypercomplex tensor completion problem and hypercomplex principal component pursuit based on a proximal splitting technique. Numerical examples show the effectiveness of the proposed methods.

Acknowledgement

I would like to express my gratitude to all those who supported me to complete my thesis. First of all, I would like to express my deep sincere gratitude to my supervisor, Professor Isao Yamada. The first contact with him was a counseling interview when I was a 2nd grade undergraduate student about 10 years ago. He was my advisor assigned by the department. He has been letting me know the interests of mathematics, signal processing, optimization and inverse problems since that time. He has constantly advised me many things including other than research regardless of whether I am a laboratory member or not even after I graduated Master course. Moreover, he readily accepted that I return to the laboratory as a part time student. Without his great support and continuous encouragement, I could not complete this thesis.

I am also grateful to Professors with the Department of Information and Communications Engineering, Tokyo Institute of Technology, Professor Tomohiko Uyematsu, Professor Kazuhiko Fukawa, Associate Professor Kenta Kasai and Professor Masahiro Yamaguchi for their fruitful and constructive comments on my research. I have had the support and encouragement of Associate Professor Masahiro Yukawa, Keio University.

My deepest appreciation goes to my supervisor during my Bachelor course, Professor Masashi Sugiyama, RIKEN and the University of Tokyo. It has been a unique chance for me to learn machine learning theory. This experience always helps me to design learning algorithms. I would like to warmly thank all of the people who helped or encouraged me in various ways during my doctoral course: Dr. Masao Yamagishi, Dr. Shunsuke Ono, Dr. Daichi Kitahara, Dr. Hiroki Kuroda, and all colleagues, alumni and secretaries in our laboratory.

I also would like to thank all members of ASDS team in NEC Corporation and Data Science & Systems Security Department in NEC Laboratories America, Inc. for their encouragement.

Of course, I would like to thank to my family: my father Tetsuhiko, my mother Saeko and my grandfather Yasuhiko. My parents Tetsuhiko and Saeko always encourage me and Yasuhiko's financial support lightened the burden of my college fee.

Finally, I would like to thank to music, live shows and rock festivals for promoting my motivation and giving me many inspirations.

Contents

List of Figures

List of Tables

List of Algorithms

1 Introduction

MULTIDIMENSIONAL information arises naturally in many areas of engineering and science since almost all observations have many attributes. Utilizing hypercomplex number system for representing such multidimensional information is one of the most effective ways because we can express multidimensional information not in terms of vectors but in terms of numbers among which we can define the four basic arithmetic operations. Indeed, it has been used in many areas such as computer graphics [CF04] and robotics [Cho92; Yua88] wind forecasting [MG09; Goh+06; Too+10] and noise reduction in acoustic systems [Ben+10]. In the statistical signal processing field, effective utilization of the m-dimensional Cayley-Dickson number system (C-D number system) [KS89; Alf06], which is a standard class of hypercomplex number systems [Bae01], including, e.g., real $\mathbb{R}$, complex $\mathbb{C}$, quaternion $\mathbb{H}$, octonion $\mathbb{O}$ and sedenion $\mathbb{S}$ etc., have been investigated [BB06; V+10; MB10; V+11].

A hypercomplex number has one real part and possibly many imaginary parts, and it can represent multidimensional data as a number for which the four arithmetic operations including multiplication and division are available. It can fulfill the four arithmetic operations for multidimensional information, which are not available with ordinary real multidimensional vectors. Moreover, based on the nontrivial algebraic structure, the multiplication of hypercomplex numbers can enjoy algebraically interactions among real and imaginary parts. For example, in 3D object modeling, each point in 3-dimensional space can have multiple attribute such as color, material, intensity, etc., and each attribute may have correlation with other attributes. Modeling of the correlations among attributes in multidimensional data will be more and more important by the popularization of 3D printer [Fra14], virtual reality, medical imaging etc. Algebraically natural operations in hypercomplex number system has great potential for such modelings of various correlations (see e.g., [Ada+11; SE99; BS03; Mir+06; ES07; GA18] for color image processing applications). However, because of the “singularity” of higher dimensional C-D number systems (see e.g., Example 1), few mathematical tools have been maintained [Wol36; Wie55; Zha97; DM98].

To overcome this situation, in this thesis, we construct some mathematical frameworks for utilizing hypercomplex number system for signal processing applications.

1.1 Algebraic Real Translations of Cayley-Dickson Linear Systems

In these signal processing applications of C-D number system, we often need to evaluate the derivative of a certain cost function for a computational strategy in the hypercomplex optimization framework. However, even if the system is represented by hypercomplex number, such a cost function should be real valued, hence not analytic as a univariate hypercomplex valued function, e.g., *Cauchy-Riemann equation* [Hen88] is not satisfied in the complex domain. As a systematic use of real differentiability of the cost function in the hypercomplex domain, a certain special calculus has been established for each hypercomplex domain. For example, the Wirtinger calculus ($\mathbb{CR}$-calculus) [Wir27; KD09] has been used for design of complex adaptive algorithms [AH09; Ada+11; Din08; Xia+10] and the $\mathbb{HR}$-calculus [Man+11] was proposed specially to design quaternion adaptive algorithms [Man+11; Jah+12; Jah+13]. However, observing $\mathbb{CR}$-calculus and $\mathbb{HR}$-calculus, we see that the complexity for such a special calculus tends to increase w.r.t. the dimension of $\mathbb{A}_m$ (see, (2.1)). Moreover, such a special calculus has not yet been established for the general C-D number systems $\mathbb{A}_m$. This situation could be a burden to create further advanced algorithms in the C-D number systems because the higher the dimensions of the hypercomplex domain, the more "singular" becomes the hypercomplex domain, e.g. quaternions losing their product commutativity, octonions their associativity, etc (see Example 1).

In this thesis, to clarify the relation between C-D linear systems and real vector valued linear systems, first we newly define two isomorphisms $\widehat{(\cdot)}$ and $\widetilde{(\cdot)}$ between real vector spaces (see Remark 4). Then, by using these isomorphisms jointly, we present an algebraic real translation (see (3.10)) of C-D linear models and show that any C-D linear model can be translated into a real one. These are based on useful as well as nontrivial algebraic properties (Theorem 1) of the proposed isomorphisms. Thanks to these properties, the proposed translation enables us to immediately obtain equivalent real models for C-D linear models as well as to fulfill almost all missions for the C-D model without requiring any special calculus.

1.1.1 Application to Online Learning

Motivated by the discussion in [Yam+02; NN11; NN12] of the bijection between C-D numbers and multi dimensional real vectors, we present an adaptive algorithm named $\mathbb{A}_m$-APSM by extending the *adaptive projected subgradient method* (APSM) [Yam03; YO04; The+11; SY13] through the proposed algebraic translation. $\mathbb{A}_m$-APSM covers a wide range of the projection based adaptive filtering algorithms in the C-D domain such as $\mathbb{A}_m$*-normalized least mean square* ($\mathbb{A}_m$-NLMS), $\mathbb{A}_m$*-affine projection algorithm* ($\mathbb{A}_m$-APA) and $\mathbb{A}_m$*-adaptive parallel subgradient projection* ($\mathbb{A}_m$-APSP). Moreover, the proposed algorithm can be extended to nonlinear adaptive filtering scenarios by applying the *kernel trick*.

Numerical examples are performed in the context of many C-D (e.g., $\mathbb{C}$, $\mathbb{H}$ and $\mathbb{O}$) valued linear system identification and nonlinear channel equalization problems and show that the effectiveness of the $\mathbb{A}_m$-APSM.

1.2 Hypercomplex Singular Value Decomposition

Our next target is the eigenvalue problem and the singular value decomposition (SVD) in C-D domain. Similar to the situation in optimization and learning, the SVD has not yet been well-established in general C-D domain since eigenvalue problems are known to be hard problems [Zha97; HS01]. In C-D domain, left and right scalar multiplications are distinct since commutativity of product does not hold in general. Therefore, we have to treat two kinds of eigenvalues, left and right eigenvalues separately. For example, quaternion right eigenvalue problems can be solved by reducing them to the equivalent complex eigenvalue problems with quaternion-complex matrix translations (complex adjoints) [Lee49; Bre51; Zha97]. However, this procedure cannot to be applied to the left eigenvalue problems and hard to be generalized for higher dimensional C-D domain. Moreover, because of these difficulties of generalization, the rank of hypercomplex matrices, which have to been discussed carefully (see Section 2.1.4), have not yet been established in general C-D domain. This situation could be a burden not only to establish low rank approximation frameworks but also to design further advanced algorithms, e.g., tensor completion which would utilize eigenvalues and SVD in C-D domain.

In this thesis, first, we propose a computational framework for Cayley-Dickson singular value decomposition. To achieve it, we introduce a new notion $\mathbb{R}$-*eigenvalue* for clarifying the relation between the eigenvalues of C-D matrices and real ones. The $\mathbb{R}$-eigenvalue is defined based on the algebraic real translation of C-D linear systems proposed in [MY14] and can be calculated for general C-D matrices. We also clarify the relation between the $\mathbb{R}$-eigenvalues and existing well-defined quaternion right eigenvalues [Lee49]. Then, we propose a definition of hypercomplex singular value decomposition (SVD) based on the $\mathbb{R}$-eigenvalues. We also clarify the relation between the proposed SVD, ranks and the known results [BM04] in well-studied quaternion case.

1.3 Hypercomplex Matrix Recovery via Convex Optimization

1.3.1 Hypercomplex Principal Component Pursuit

One of standard approximation methods utilizing simpler structure of matrices is the *robust principal component analysis (RPCA)* [Can+11], which separates an input matrix into a low-rank and sparse ones:

$$\underset{\boldsymbol{L},\boldsymbol{S}\in\mathbb{R}^{M\times N}}{\text{minimize}} \quad \operatorname{rank}(\boldsymbol{L}) + \lambda \|\boldsymbol{S}\|_0 \quad \text{s.t.} \quad \boldsymbol{M} = \boldsymbol{L} + \boldsymbol{S}, \tag{1.1}$$

where $\|\cdot\|_0$ denotes the ℓ_0-norm, which counts the number of non-zero entries of $\boldsymbol{S}$. The RPCA has been successfully used for various signal processing applications such as source separation [Hua+12], face recognition [Pen+12] and so on. Unfortunately, the formulation (1.1) is NP-hard, so its convex relaxation called the principal component pursuit [Can+11] is often considered instead of the RPCA. Recently, the PCP is extended to simpler hypercomplex domains $\mathbb{C}$ and $\mathbb{H}$ using the complex matrix isomorphism [CY16]. However, it can be only applied up to four dimensional data and hard to be generalized to octonion and higher dimensional C-D domain only with this isomorphism.

In this thesis, to establish a RPCA framework in hypercomplex domain, first, we present Cayley-Dickson principal component analysis (C-D PCP) as a convex relaxation of the C-D extension of RPCA. This relaxation is based on the $\mathbb{R}$-rank proposed in [MY18b] and a new sparsity measure, ℓ_1-norm of C-D matrices. This sparsity measure can be interpreted as evaluating a group sparsity of real matrices, so the proximity operator can be easily calculated. Hence, the C-D PCP is a convex optimization in real domain which can be solved by applying proximal splitting techniques to a certain structured convex optimization problem. We finally propose an hypercomplex PCP algorithm in $\mathbb{A}_m$ based on the *Douglas-Rachford splitting (DRS)* [CP07]. The proposed algorithm is a C-D generalization of a PCP algorithm (DR-PCP) proposed in [GY10] and can be applied to general C-D domains.

Numerical experiments are performed in the context of recovering sparsely corrupted low rank matrices in octonion domain and show that the proposed algorithm successfully utilizes algebraically natural correlations of real and all imaginary parts to recover much more faithfully the original matrices, corrupted randomly by noise, than a part-wise real, complex and quaternion PCP "algorithms".

1.3.2 Hypercomplex Low Rank Matrix Completion

In real world applications, the range of each attribute is often restricted. For example, the observations of color images are intensities of RGB color spaces and thus they are always non-negative. In an audio context, it is typical to consider only the non-negative magnitudes without taking care about phases [SM18]. Moreover, in both cases, the low rank assumption is typically justified by the fact that natural images have certain patterns and audio signals are superpositions of relatively few component signals. As seen in these applications, therefore, recovering hypercomplex low-rank matrix with non-negative constraint is expected to play an important role in many applications.

In this thesis, to establish a matrix completion framework with input restriction, first, we formulate hypercomplex matrix completion problem with non-negative constraint. To achieve it, we introduce a *part-wise non-negativeness* of hypercomplex numbers. Thanks to the simple definition of general hypercomplex non-negativeness, the hypercomplex non-negative matrix completion problem can be recasted to equivalent to a structured convex optimization problem in real domain by utilizing algebraic translations proposed in [MY14]. We then propose an algorithmic solution to hypercomplex non-negative low rank completion completion algorithm based on a proximal splitting method, *Douglas-Rachford splitting (DRS)* [CP07]. The proposed algorithm is a C-D generalization of the dual of non-negative matrix completion algorithm proposed in [SM18] and can be applied to general C-D domains.

Numerical experiments including a scenario of high-dimensional hypercomplex non-negative matrix completion problem shows that the proposed algorithm successfully utilizes algebraically natural correlations of each attribute to recover much more faithfully the original information, masked randomly by noise, than a part-wise state-of-art non-negative matrix completion algorithm.

1.4 Hypercomplex Low Rank Tensor Completion via Convex Optimization

The benefits from the algebraically natural operations of C-D number systems can be inherited as well in hypercomplex multi-way arrays, i.e., tensors (see Appendix A.7.1). In the areas of signal, image processing, and data science, representing huge volume of multi-relational datasets by tensors has achieved remarkable results in many applications [SB00; KB09; Cic+09; Cic+17; Cic+15]. In the contexts of such applications, although all entries in the tensors have been real otherwise at most complex numbers, canonical polyadic decomposition (CPD) is typically used to decompose a data tensor into the sum of rank one tensors for maximum interpretability. Computation of CPD of a given tensor has been reduced to simultaneous matrix diagonalization [Lat06; DL14] based on matrix eigenvalue analysis [HJ12]. Therefore, to extend existing tensor data analysis to tensors of hypercomplex data for broader applications, we have to consider certain eigenvalue problems of hypercomplex matrices.

Unfortunately, since complete data tensors are not always available in real world applications, estimation of entire information from observed real or complex data tensors is crucial and it has been achieved by utilizing simpler structures such as low-rankness.

As an application of the proposed frameworks, we next formulate hypercomplex low N-rank tensor completion problem as a convex optimization, which can be solved by applying proximal splitting techniques to a certain structured convex optimization problem. We finally propose an hypercomplex low rank tensor completion algorithm based on *Douglas-Rachford splitting (DRS)* [CP07]. The proposed algorithm is a C-D generalization of a tensor completion algorithm (DR-TR) proposed in [Gan+11] and can be applied to general C-D domains.

Numerical experiments including a scenario of color tensor completion problem in quaternion domain show that the proposed algorithm successfully utilizes algebraically natural correlations of each color space to recover much more faithfully the original color information, masked randomly by noise, than a part-wise real tensor completion algorithm.

1.5 Organization

The rest of this thesis is organized as follows. In Chapter 2, we first introduce basic definitions and properties of a standard class of hypercomplex number systems, Cayley-Dickson (C-D) number systems. Throughout this thesis, we will discuss almost everything in this number system. We also in this chapter briefly review two optimization methods, the adaptive projected subgradient method (APSM) and the Douglas-Rachford splitting (DRS) technique. These techniques will be utilized for designing further advanced hypercomplex algorithms in Chapter 4-7.

In Chapter 3, we will introduce two kinds of isomorphisms, which map C-D matrices and vectors to real ones and clarify their algebraic properties. By utilizing these two isomorphisms jointly, we will then propose algebraic real translations of Cayley-Dickson linear systems and show that by these translations any C-D linear systems can be translated into the equivalent real linear systems. These techniques will be key ideas throughout this study.

In Chapter 4, we formulate a hypercomplex online parameter estimation (adaptive filtering) problem in C-D domain and show that any parameter estimation problems in C-D linear systems can be reduced to the equivalent ones in real linear systems by utilizing the algebraic real translations proposed in Chapter 3. We then propose a new hypercomplex adaptive algorithm named $\mathbb{A}_m$-adaptive projected subgradient method ($\mathbb{A}_m$-APSM) by applying the APSM to the translated real problem. The proposed algorithm is C-D generalization of the APSM and inherits the powerful properties of the original APSM such as monotone approximation and asymptotic optimality. Moreover, similar to the real case, the proposed algorithm includes wide hypercomplex algorithm such as the C-D counterparts of the normalized least mean squares ($\mathbb{A}_m$-NLMS), the affine projection ($\mathbb{A}_m$-APA) and the adaptive parallel subgradient projection ($\mathbb{A}_m$-APSP) as special cases of $\mathbb{A}_m$-APSM.

In Chapter 5, we show that nonlinear online learning problem in C-D domain can be reduced to linear one in read domain by combining the algebraic real translation proposed in Chapter 3 and the kernel trick. We then propose a new hypercomplex kernelized online learning algorithm named $\mathbb{A}_m$-kernel adaptive projected subgradient projection ($\mathbb{A}_m$-KAPSP) by applying the APSM to the reduced real linear online learning problem. Similar to the linear case, the proposed method includes wide hypercomplex nonlinear online learning algorithms such as the C-D counterparts of the kernel normalized least mean squares ($\mathbb{A}_m$-KNLMS), the kernel affine projection ($\mathbb{A}_m$-APA) as special cases.

In Chapter 6 we will introduce a new notion $\mathbb{R}$-eigenvalue, which can be defined and calculated for any C-D square matrices, and show that the $\mathbb{R}$-eigenvalue is a natural extension of the original eigenvalues keeping consistency with known results. Moreover, we also newly define the C-D singular value decomposition (C-D SVD) and $\mathbb{R}$-rank of general C-D matrices based on the $\mathbb{R}$-eigenvalue, and clarify their properties and the relations to known results.

In Chapter 7, we focus on two practical hypercomplex matrix recovery problems, hypercomplex robust principal component analysis (RPCA) and hypercomplex non-negative low rank matrix completion. For hypercomplex RPCA, we relax it by utilizing a new sparsity measure, ℓ_1 norm of C-D matrices to formulate the C-D counterpart of principal component pursuit (PCP). We then show that the PCP in C-D domain can be reduced to the equivalent structured convex optimization problem in real domain and propose a new PCP algorithm for general C-D domain based on the Douglas-Rachford splitting (DRS). For hypercomplex matrix completion with non-negative constraints, we propose part-wise non-negativeness of hypercomplex matrices. It enables us to recast this problem to equivalent structured convex optimization problem in real domain. We then propose an algorithmic solution to hypercomplex non-negative low rank matrix completion algorithm based on the DRS.

In Chapter 8, we show that a hypercomplex low rank tensor completion problem can be reduced to the equivalent structured convex optimization problem in real domain, and by applying the DRS technique to the reduced problem we propose a new hypercomplex low rank tensor completion algorithm, which is available for general C-D domain.

Chapter 9 concludes this study.

2 Preliminaries

2.1 Cayley-Dickson Number Systems

2.1.1 Definitions

All notations necessary to deal with hypercomplex number systems are summarized in TABLE 2.1. Let $\mathbb{N}$ and $\mathbb{R}$ be respectively the set of all non-negative integers and

Table 2.1.: Notations related to hypercomplex number systems

Symbol	Definition
$\mathbb{N}$	Set of all non-negative integer
$\mathbb{R}$	Set of all real numbers
$\mathbb{C}$	Set of all complex numbers
$\mathbb{H}$	Set of all quaternions
$\mathbb{O}$	Set of all octonions
$\mathbb{A}_m$	Set of all m-dimensional C-D numbers
$\mathbb{A}_m^{M\times N}$	Set of all $M \times N$ C-D matrices/vectors
$\mathbf{i}_\ell$	ℓth imaginary unit in $\mathbb{A}_m$ $(\ell \in \{1,\dots,m\})$
$\Im_\ell$	ℓth imaginary part of C-D numbers/vectors/matrices
$\imath$	Imaginary unit in $\mathbb{C}$
$\jmath$	Second imaginary unit in $\mathbb{H}\setminus\mathbb{C}$
κ	Third imaginary unit in $\mathbb{H}\setminus\mathbb{C}$ defined as $\kappa := \imath\jmath$
$(\cdot)^*$	Conjugate of C-D numbers/vectors/matrices
$(\cdot)^\top$	Transpose of C-D vectors/matrices
$(\cdot)^\mathsf{H}$	Hermitian of C-D vectors/matrices
$\widehat{(\cdot)}$	Trivial mapping of C-D matrices/vectors
$\widecheck{(\cdot)}$	Inverse mapping of $\widehat{(\cdot)}$
$\widetilde{(\cdot)}$	Non-trivial mapping of C-D matrices/vectors
$\underset{\smile}{(\cdot)}$	Inverse mapping of $\widetilde{(\cdot)}$
$\mathfrak{S}_{\mathbb{A}_m}(M,N)$	Image of $\mathbb{A}_m^{M\times N}$ by $\widetilde{(\cdot)}$
$\boldsymbol{L}_M^{(\ell)}$	Real matrix for non-trivial mapping (see (3.6))
$\delta_{\alpha,\beta}^{(\gamma)}$	Value determined by the result of $\mathbf{i}_\alpha\mathbf{i}_\beta$ (see (3.7))
$\chi(\cdot)$	Complex adjoint of quaternion matrices (see (6.1))

the set of all real numbers. An m-dimensional *hypercomplex* number $\mathbb{A}_m$ $(m \in \mathbb{N}\setminus\{0\})$ can be defined as [KS89]

$$a := a_1\mathbf{i}_1 + a_2\mathbf{i}_2 + \cdots + a_m\mathbf{i}_m \in \mathbb{A}_m,\ a_1,\dots,a_m \in \mathbb{R} \tag{2.1}$$

based on imaginary units $\mathbf{i}_1,\dots,\mathbf{i}_m$, where $\mathbf{i}_1 = 1$ represents the vector identity element. Any hypercomplex number is expressed uniquely in the form of (2.1). For

$a \in \mathbb{A}_m$ the coefficient a_ℓ ($\ell = 1, \ldots, m$) of each imaginary unit $\mathbf{i}_\ell \in \mathbb{A}_m$ is represented as $a_\ell = \Im_\ell(a)$. A *multiplication table* defines the products of any imaginary unit with each other or with itself (e.g., $\mathbf{i}_1^2 = 1, \mathbf{i}_2^2 = -1$ and $\mathbf{i}_1\mathbf{i}_2 = \mathbf{i}_2\mathbf{i}_1 = \mathbf{i}_2$ for $\mathbb{A}_2(=: \mathbb{C})$). The *addition* and the *subtraction* of two hypercomplex numbers are defined as commutative binomial operations:

$$a \pm b := (a_1 \pm b_1)\mathbf{i}_1 + (a_2 \pm b_2)\mathbf{i}_2 + \cdots + (a_m \pm b_m)\mathbf{i}_m$$

for $a, b \in \mathbb{A}_m$, where $b := b_1\mathbf{i}_1 + b_2\mathbf{i}_2 + \cdots + b_m\mathbf{i}_m$, $b_1, \ldots, b_m \in \mathbb{R}$. From the unique expression of (2.1), the *multiplication* of two hypercomplex numbers

$$ab = (a_1\mathbf{i}_1 + a_2\mathbf{i}_2 + \cdots + a_m\mathbf{i}_m)(b_1\mathbf{i}_1 + b_2\mathbf{i}_2 + \cdots + b_m\mathbf{i}_m) \tag{2.2}$$

$$:= \sum_{k=1}^{m}\sum_{\ell=1}^{m} a_k b_\ell \mathbf{i}_k \mathbf{i}_\ell \in \mathbb{A}_m \tag{2.3}$$

is determined uniquely according to the multiplication table. The *conjugate* of hypercomplex number a as

$$a^* := a_1\mathbf{i}_1 - a_2\mathbf{i}_2 - \cdots - a_m\mathbf{i}_m. \tag{2.4}$$

In this thesis, we consider the hypercomplex number systems which are constructed recursively by the so-called *Cayley-Dickson construction* (*C-D construction* or *C-D doubling procedure*) [KS89; Bae01]. The C-D construction is a standard method for extending a number system. This method has been used in extending $\mathbb{R}$ to $\mathbb{C}$, $\mathbb{C}$ to $\mathbb{H}$, $\mathbb{H}$ to $\mathbb{O}$ and $\mathbb{O}$ to $\mathbb{S}$. By using the C-D construction, an m-dimensional hypercomplex number $\mathbb{A}_m$ is extended to $\mathbb{A}_{2m}$ as

$$z := a + b\mathbf{i}_{m+1} \in \mathbb{A}_{2m}, \quad a, b \in \mathbb{A}_m,$$

where $\mathbf{i}_{m+1} \notin \mathbb{A}_m$ is the additional imaginary unit for doubling the dimension of $\mathbb{A}_m$ satisfying $\mathbf{i}_{m+1}^2 = -1$, $\mathbf{i}_1\mathbf{i}_{m+1} = \mathbf{i}_{m+1}\mathbf{i}_1 = \mathbf{i}_{m+1}$ and $\mathbf{i}_v\mathbf{i}_{m+1} = -\mathbf{i}_{m+1}\mathbf{i}_v =: \mathbf{i}_{m+v}$ for all $v = 2, \ldots, m$. Moreover, the multiplication in $\mathbb{A}_{2m}$ is defined with use of the multiplication rule in $\mathbb{A}_m$ as

$$\begin{aligned}(a + b\mathbf{i}_{m+1})(c + d\mathbf{i}_{m+1}) &:= (ac - d^*b) + (ad + bc^*)\mathbf{i}_{m+1} \\ &\in \mathbb{A}_{2m} \quad a, b, c, d \in \mathbb{A}_m,\end{aligned} \tag{2.5}$$

which determines the unique multiplication table (e.g., see Remark 1-4)). As an example of C-D construction, the real number system $\mathbb{R}$ ($= \mathbb{A}_1$) is extended to the complex number system $\mathbb{C}$ ($= \mathbb{A}_2$) by the C-D construction with $\mathbf{i}_2 := \imath$ ($\imath \in \mathbb{C}, \imath^2 = -1$). Note that the value m in $\mathbb{A}_m$ is restricted to the form of 2^n ($n \in \mathbb{N}$). The hypercomplex number systems constructed inductively from the real number by the C-D construction are called in this thesis *Cayley-Dickson number system*[1] (*C-D number system*).

Fact 1. *According to the C-D construction, imaginary units of C-D number system* $\mathbb{A}_m$ *have the following properties (some results are found explicitly or implicitly, e.g., in [Alf06]):*

1. $a\mathbf{i}_\alpha = \mathbf{i}_\alpha a$ *for all* $a \in \mathbb{R}$, $\alpha \in \{1, \ldots, m\}$.
2. $\mathbf{i}_\alpha^2 = -1$ *for all* $\alpha \in \{2, \ldots, m\}$.
3. $\mathbf{i}_\alpha \mathbf{i}_\beta = -\mathbf{i}_\beta \mathbf{i}_\alpha$ $(\alpha \neq \beta)$ *for all* $\alpha, \beta \in \{2, \ldots, m\}$.
4. $\mathbf{i}_\alpha(\mathbf{i}_\alpha \mathbf{i}_\beta) = -\mathbf{i}_\beta$ *for all* $\alpha \in \{2, \ldots, m\}, \beta \in \{1, \ldots, m\}$.
5. *For all* $\alpha, \beta \in \{1, \ldots, m\}$, *there exists unique* $\gamma \in \{1, \ldots, m\}$ *s.t.* $\mathbf{i}_\alpha \mathbf{i}_\beta = \mathbf{i}_\gamma$ *or* $-\mathbf{i}_\gamma$.

Remark 1.
1. Fact 1-1) is verified by (2.2) as follows: $\mathbf{i}_\alpha a = 1\mathbf{i}_\alpha \times a\mathbf{i}_1 = a \times 1 \times \mathbf{i}_\alpha \mathbf{i}_1 = a\mathbf{i}_\alpha$.
2. Fact 1-3) shows the anti-commutativity of $\mathbb{A}_m$ $(m \geq 4)$.
3. Fact 1-4) is verified straightforwardly by the definition of multiplication (2.5). Note that this property for $\mathbb{A}_4$ $(= \mathbb{H})$ can also be checked by the associativity of $\mathbb{H}$. Moreover, this property for $\mathbb{A}_8$ $(= \mathbb{O})$ can also be checked because of the fact that $\mathbb{O}$ is an *alternative algebra* [Zhe01].
4. Fact 1-5) is verified straightforwardly by the definition of multiplication (2.5). For simplicity, consider for $\mathbf{i}_3, \mathbf{i}_4 \in \mathbb{A}_4$. In this case, by substituting $\mathbf{i}_4 = \mathbf{i}_2\mathbf{i}_3$ into (2.5), we have $\mathbf{i}_3\mathbf{i}_4 = (0 \times 0 - (-\mathbf{i}_2) \times 1) + (0 \times \mathbf{i}_2 + 1 \times 0)\mathbf{i}_3 = \mathbf{i}_2$.
5. Fact 1 ensures $aa^* = \sum_{\ell=1}^{m} a_\ell^2 \geq 0$ for any $a \in \mathbb{A}_m$ in (2.1) and $a^* \in \mathbb{A}_m$ in (2.4), so we can define the *absolute value* of the hypercomplex number a as

$$|a| := \sqrt{aa^*}.$$

[1] One of the other well-known hypercomplex number systems is the *Clifford algebra* [Cli78]. The dimension of the Clifford algebra is also the form of 2^n, but the products of two imaginary unit are different from that of the C-D number system since this algebra is constructed by a different way (see e.g., [Bae01]). $\mathbb{R}$, $\mathbb{C}$ and $\mathbb{H}$ are also examples of the Clifford algebra. However, $\mathbb{O}$ and higher dimensional C-D number systems are not.

Example 1. 1. A representative example of hypercomplex number is the *quaternion* $\mathbb{H}$. A quaternion number is a 4-dimensional hypercomplex number which is defined as

$$q = q_1 + q_2 \imath + q_3 \jmath + q_4 \kappa \in \mathbb{H}, \quad q_1, q_2, q_3, q_4 \in \mathbb{R}$$

with the multiplication table:

$$\begin{gathered} \imath\jmath = -\jmath\imath = \kappa,\ \jmath\kappa = -\kappa\jmath = \imath,\ \kappa\imath = -\imath\kappa = \jmath, \\ \imath^2 = \jmath^2 = \kappa^2 = -1 \end{gathered} \tag{2.6}$$

by letting $m = 4$, $\mathbf{i}_1 = 1$, $\mathbf{i}_2 = \imath$, $\mathbf{i}_3 = \jmath$ and $\mathbf{i}_4 = \kappa$. From (2.6), quaternions are not *commutative*, i.e., $pq = qp$ or $pq \neq qp$ for $p, q \in \mathbb{H}$ in general.

2. The octonion $\mathbb{O}$ can be constructed from the quaternion $\mathbb{H}$ by the C-D construction. Note that the multiplication in $\mathbb{O}$ is neither commutative nor *associative*, i.e., neither $pq = qp$ nor $(pq)r = p(qr)$ for $p, q, r \in \mathbb{O}$ holds in general [Bae01]. For the octonion multiplication table, see, e.g., [Bae01].

The C-D number system can be seen as an algebraically natural higher dimensional generalization of our familiar fields, i.e., $\mathbb{R}$ and $\mathbb{C}$.

2.1.2 Hypercomplex Vectors and Matrices

Let $\mathbb{A}_m^N := \{[x_1, \ldots, x_N]^\top | x_\ell \in \mathbb{A}_m\,(\ell = 1, \ldots, N)\}$ for $\forall N \in \mathbb{N} \setminus \{0\}$, where $(\cdot)^\top$ stands for the transpose. Define $\langle \boldsymbol{x}, \boldsymbol{y} \rangle_{\mathbb{A}_m^N} := \boldsymbol{x}^{\mathsf{H}} \boldsymbol{y} \in \mathbb{A}_m, \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{A}_m^N$ and $\|\boldsymbol{x}\|_{\mathbb{A}_m^N} := \langle \boldsymbol{x}, \boldsymbol{x} \rangle_{\mathbb{A}_m^N}^{1/2}$, $\forall \boldsymbol{x} \in \mathbb{A}_m^N$, where $(\cdot)^{\mathsf{H}}$ denotes the *Hermitian transpose* of vectors or matrices (e.g., $\boldsymbol{x}^{\mathsf{H}} := [x_1^*, \ldots, x_N^*]$ for $\boldsymbol{x} := [x_1, \ldots, x_N]^\top \in \mathbb{A}_m^N$, where $x_1 \ldots, x_N \in \mathbb{A}_m$). We also define the *addition* of two hypercomplex vectors $\boldsymbol{x} + \boldsymbol{y} := [x_1 + y_1, \cdots, x_N + y_N]^\top \in \mathbb{A}_m^N$ for $\boldsymbol{x}, \boldsymbol{y} (:= [y_1, \ldots, y_N]^\top) \in \mathbb{A}_m^N$. Let $\mathcal{S} := \mathbb{R}$, $\mathcal{S} := \mathbb{C}$ or $\mathcal{S} := \mathbb{A}_m$ $(m \geq 4)$, and call the element of $\mathcal{S}$ *scalar*. If we define the *left scalar multiplication* as $\alpha \boldsymbol{x} := [\alpha x_1, \ldots, \alpha x_N]^\top \in \mathbb{A}_m^N$ for $\alpha \in \mathcal{S}$ and $\boldsymbol{x} \in \mathbb{A}_m^N$, we have $\alpha \boldsymbol{x} + \beta \boldsymbol{y} \in \mathbb{A}_m^N$, $\forall \alpha, \beta \in \mathcal{S}, \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{A}_m^N$. We can also define the *right scalar multiplication* $\boldsymbol{x}\alpha \in \mathbb{A}_m^N$ in a similar way. More generally, by letting $\mathbb{A}_m^{M \times N} := \{[\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N] | \boldsymbol{x}_\ell \in \mathbb{A}_m^M\,(\ell = 1, \ldots, N)\}$, we can define, in the same way, the *addition* of two hypercomplex matrices $\boldsymbol{A} + \boldsymbol{B} \in \mathbb{A}_m^{M \times N}$ and the *left (right) scalar multiplication* $\alpha \boldsymbol{A}$ $(\boldsymbol{A}\alpha) \in \mathbb{A}_m^{M \times N}$ $(\forall \boldsymbol{A}, \boldsymbol{B} \in \mathbb{A}_m^{M \times N}, \forall \alpha \in \mathcal{S})$. If $\mathcal{S} := \mathbb{R}$, the space $\mathbb{A}_m^{M \times N}$ is a vector space over $\mathcal{S}$ $(:= \mathbb{R})$. On the other hand, in the case where $\mathcal{S} := \mathbb{A}_m$ $(m \geq 4)$, $\mathcal{S}$ is non-commutative and $\mathbb{A}_m$ $(m \geq 8)$ is *distributive (non-associative) algebra* [Hjø66], so the space $\mathbb{A}_m^{M \times N}$ over $\mathcal{S}$ $(:= \mathbb{A}_m)$ is not a vector space in the standard sense. For any nonempty closed convex set[2] $C \subset \mathbb{A}_m^N$, the *projection operator* $P_C^{\mathbb{A}_m} : \mathbb{A}_m^N \to C$ assigns a vector $\boldsymbol{x} \in \mathbb{A}_m^N$ to the unique vector $P_C^{\mathbb{A}_m}(\boldsymbol{x}) \in C$ s.t. $d_{\mathbb{A}_m}(\boldsymbol{x}, C) := \left\| \boldsymbol{x} - P_C^{\mathbb{A}_m}(\boldsymbol{x}) \right\|_{\mathbb{A}_m^N} = \min_{\boldsymbol{y} \in C} \|\boldsymbol{x} - \boldsymbol{y}\|_{\mathbb{A}_m^N}$.

2.1.3 Hypercomplex Eigenvalue Problems

Similar to the real and complex case, we can formally consider the eigenvalue problems in C-D domain. However, as seen in the quaternion case, the multiplication in C-D number systems $\mathbb{A}_m$ ($m \geq 4$) is not commutative, that is, the left and right scalar multiplications are different, so we have to treat them separately.

In this thesis, we respectively call a C-D valued scalar λ^ℓ (λ^r) $\in \mathbb{A}_m$ and a C-D valued vector $\boldsymbol{x}^\ell$ ($\boldsymbol{x}^r$) $\in \mathbb{A}_m^N$ a *left* (*right*) *eigenvalue* and a *left* (*right*) *eigenvector* provided that $\boldsymbol{A}\boldsymbol{x}^\ell = \lambda^\ell \boldsymbol{x}^\ell$ ($\boldsymbol{A}\boldsymbol{x}^r = \boldsymbol{x}^r \lambda^r$). Note that both left eigenvalue and eigenvector are distinct from right ones for a common C-D matrix $\boldsymbol{A} \in \mathbb{A}_m^{N \times N}$ in general.

Unfortunately, there has been known few cases where hypercomplex eigenvalues can be computed systematically. In almost all cases, even the existences of them have not been explained explicitly to the best of the author's knowledge. In the quaternion domain, it is known that right eigenvalues always exist and indeed a well-defined method is available for computing quaternion right eigenvalues by reducing the quaternion right eigenvalue problem to the equivalent complex one [Lee49; Bre51; Zha97]. Therefore, the singular valued decomposition over the quaternion domain is based on the right eigenvalue. On the other hand, the method used in the right eigenvalue problem cannot be used in the left one because of the lack of commutativity of multiplication. Wood proved that any $N \times N$ quaternion matrix has at least one left eigenvalue [Woo85] but even for small size matrices the left eigenvalues are still open problem in spite of many previous studies [HS01; Zha07; Che10]. For example, it was proved that a 2×2 quaternion matrix may have one, two, or an infinite number of left eigenvalues [HS01] but the proof seems to be difficult to generalize for $N > 2$. For octonion and higher dimensional C-D domain, the general solution is available for very limited cases [DM98]. Moreover, it seems that there have not been reported any systematic solution even for right eigenvalue problem since the method for solving quaternion right eigenvalue problem cannot be generalized for higher dimensional cases because of the lack of associativity of multiplication.

[2] A set $C \subset \mathbb{A}_m^N$ is said to be *convex* provided that $\forall \boldsymbol{x}, \boldsymbol{y} \in C, \forall \nu \in (0,1)$, $\nu \boldsymbol{x} + (1-\nu)\boldsymbol{y} \in C$.

2.1.4 Rank of Hypercomplex Matrices

We finally discuss the *rank* of hypercomplex matrices. In real and complex domains, the rank of a matrix is defined as the maximum number of column vectors of it which are linearly independent. In quaternion and higher dimensional C-D domain, we can consider *left and right linearly independence* since left and right scalar multiplications are distinct, so we have to define carefully the rank over $\mathbb{A}_m$. In quaternion domain, by convention, the rank is defined as the maximum number of columns which are *right* linearly independent [Zha97], since if we define so, the rank becomes equal to the number of positive singular values with the right eigenvalues. For octonion and higher dimensional C-D domain, the concrete definition of the rank has not yet been established to the best of the author's knowledge.

If we consider the low-rank approximation of hypercomplex matrices, we have to be careful about the set of low-rank matrices. Since a C-D matirx $\boldsymbol{A} \in \mathbb{A}^{M\times N}$ can be represented as $\boldsymbol{A} := \sum_{\ell=1}^{m} \boldsymbol{A}_\ell \mathbf{i}_\ell$ ($\boldsymbol{A}_\ell \in \mathbb{R}^{M\times N}$, $\ell = 1, \ldots, m$), we can immediately consider low-rank approximating $\boldsymbol{A}_1, \ldots, \boldsymbol{A}_m$ separately. However, considering low-rank approximation of one hypercomplex matrix and those of separated real matrices mentioned above are completely different. This can be verified with the simplest $\mathbb{C}^{2\times 2}$ case by the following proposition:

Proposition 1 (Inclusion relation does not hold between the sets of low-rank matrices and the set of separately low-rank matrices in $\mathbb{C}$)**.** *We denote by* $\mathbb{C}^{2\times 2}_{\mathrm{rank}\le(1,1)}$ *the set of all complex matrices whose real and imaginary parts are of rank less than* 1 *as real matrices, i.e.,* $\mathbb{C}^{2\times 2}_{\mathrm{rank}\le(1,1)} := \{\boldsymbol{A}_1 + \boldsymbol{A}_2 \imath \in \mathbb{C}^{2\times 2} | \mathrm{rank}(\boldsymbol{A}_1) \le 1, \mathrm{rank}(\boldsymbol{A}_2) \le 1\}$*. We also denote by* $\mathbb{C}^{2\times 2}_{\mathrm{rank}\le 1}$ *the set of all complex matrices whose ranks are less than* 1 *in the sense of complex vector space, i.e.,* $\mathbb{C}^{2\times 2}_{\mathrm{rank}\le 1} := \{\boldsymbol{A} \in \mathbb{C}^{2\times 2} | \mathrm{rank}(\boldsymbol{A}) \le 1\}$*. Then, there is no inclusion relation between* $\mathbb{C}^{2\times 2}_{\mathrm{rank}\le(1,1)}$ *and* $\mathbb{C}^{2\times 2}_{\mathrm{rank}\le 1}$*, i.e.,*

$$\mathbb{C}^{2\times 2}_{\mathrm{rank}\le(1,1)} \not\subset \mathbb{C}^{2\times 2}_{\mathrm{rank}\le 1} \quad \text{and} \quad \mathbb{C}^{2\times 2}_{\mathrm{rank}\le 1} \not\subset \mathbb{C}^{2\times 2}_{\mathrm{rank}\le(1,1)}.$$

Proof. See Appendix A.1. □

Proposition 1 suggests that by establishing mathematically sound low-rank approximation techniques for hypercomplex matrices or tensors, we could obtain great potential to have variety of novel formulations to approximate multidimensional information in terms of matrices or tensors of algebraically simple structures.

2.2 Optimization Methods

In this section, we briefly review powerful optimization methods, the *adaptive projected subgradient method (APSM)* and *Douglas-Rachford splitting (DRS)* utilized in this thesis. All methods can be discussed in general Hilbert space. Let $\mathcal{H}$ be a (possibly infinite dimensional) real Hilbert space equipped with the inner product $\langle \boldsymbol{x}, \boldsymbol{y} \rangle_{\mathcal{H}}$, $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{H}$, and its induced norm $\|\boldsymbol{x}\|_{\mathcal{H}} := \langle \boldsymbol{x}, \boldsymbol{x} \rangle_{\mathcal{H}}^{1/2}$, $\forall \boldsymbol{x} \in \mathcal{H}$.

2.2.1 Adaptive Projected Subgradient Method

Here, we review the *adaptive projected subgradient method* (APSM) [Yam03; YO04]. The APSM will be applied for designing hypercomplex linear adaptive algorithms in Chapter 4 and kernelized hypercomplex nonlinear adaptive algorithms in Chapter 5.

For any closed convex set $C \subset \mathcal{H}$, the projection operator $P_C : \mathcal{H} \to C$ can be defined as a mapping to a unique point $P_C(\boldsymbol{x}) := \arg\min_{\boldsymbol{y} \in C} \|\boldsymbol{x} - \boldsymbol{y}\|_{\mathcal{H}}$.

Fact 2 (Adaptive projected subgradient method (APSM) [Yam03; YO04])**.** *Let $\Theta_k : \mathcal{H} \to [0, \infty)$ $(k \in \mathbb{N})$ be a sequence of functions and $K \subset \mathcal{H}$ a nonempty closed convex set. For an arbitrarily given $\boldsymbol{h}_0 \in K$, the APSM produces a sequence $(\boldsymbol{h}_k)_{k \in \mathbb{N}}$ by*

$$\boldsymbol{h}_{k+1} = \begin{cases} P_K\left(\boldsymbol{h}_k - \lambda_k \frac{\Theta_k(\boldsymbol{h}_k)}{\|\Theta_k'(\boldsymbol{h}_k)\|_{\mathcal{H}}^2} \Theta_k'(\boldsymbol{h}_k)\right) \\ \qquad\qquad\qquad\qquad (\textit{if } \Theta_k'(\boldsymbol{h}_k) \neq \boldsymbol{0}), \\ P_K(\boldsymbol{h}_k) \qquad\qquad\qquad (\textit{otherwise}), \end{cases} \tag{2.7}$$

where $\Theta_k'(\boldsymbol{h}_k) \in \partial\Theta_k(\boldsymbol{h}_k)$, $0 \leq \lambda_k \leq 2$. Then the sequence $(\boldsymbol{h}_k)_{k \in \mathbb{N}}$ satisfies the following properties:

(1) (Monotone approximation) Suppose that

$$\boldsymbol{h}_k \notin \Omega_k := \{\boldsymbol{h} \in K | \Theta_k(\boldsymbol{h}) = \inf_{\boldsymbol{x} \in K} \Theta_k(\boldsymbol{x})\} \neq \varnothing.$$

Then, by using $\forall \lambda_k \in \left(0, 2\left(\frac{\inf_{\boldsymbol{x} \in K} \Theta_k(\boldsymbol{x})}{\Theta_k(\boldsymbol{h}_k)}\right)\right)$, we have

$$\boldsymbol{h}^{\star(k)} \in \Omega_k, \quad \left\|\boldsymbol{h}_{k+1} - \boldsymbol{h}^{\star(k)}\right\|_{\mathcal{H}} \leq \left\|\boldsymbol{h}_k - \boldsymbol{h}^{\star(k)}\right\|_{\mathcal{H}}.$$

(2) (Asymptotic optimality) Suppose

$$\exists N_0 \in \mathbb{N} \text{ s.t. } \begin{cases} \inf_{\boldsymbol{x}} \Theta(\boldsymbol{x}), \forall k \geq N_0 \text{ and} \\ \Omega := \bigcap_{k \geq N_0} \Omega_k \neq \varnothing. \end{cases} \tag{2.8}$$

Then $(\boldsymbol{h}_k)_{k\in\mathbb{N}}$ *is bounded. Moreover, if we use* $\lambda_n \in [\varepsilon_1, 2-\varepsilon_2] \subset (0,2)$ *for all* k*, we have*

$$\lim_{k\to\infty} \Theta_k(\boldsymbol{h}_k) = 0 \tag{2.9}$$

provided that $(\Theta'_k)_{k\in\mathbb{N}}$ *is bounded.* □

The APSM is highly generalized framework of adaptive algorithm, which includes many useful algorithms such as normalized least square (NLMS) [NN67], affine projection algorithm (APA) [OU84] and adaptive parallel subgradient projection (APSP) [Yam+02] as special cases.

2.2.2 Douglas-Rachford Splitting

Here, we briefly review a classical optimization algorithm, the *Douglas-Rachford splitting (DRS)*, which we will subsequently utilize for solving two optimization problems, hypercomplex low rank tensor completion (Chapter 8) and hypercomplex principal component pursuit (Chapter 7). The DRS itself has a long history [DR56; CP07; EB92]. It solves the minimization of the sum of two functions

$$f(\boldsymbol{x}) + g(\boldsymbol{x}), \tag{2.10}$$

where f and g are assumed to be elements of the class, denoted by $\Gamma_0(\mathcal{H})$, of proper lower semicontinuous convex functions from a real Hilbert space $\mathcal{H}$ to $\mathbb{R} \cup \{+\infty\}$. For given $\gamma \in (0, +\infty)$, the DRS approximates a minimizer of (2.10) with $\left(\operatorname{prox}_{\gamma g}(x_u)\right)_{u\geq 0}$ by generating the following sequence $(\boldsymbol{x}_u)_{u\geq 0}$:

$$\boldsymbol{x}_{u+1} \leftarrow \boldsymbol{x}_u + t_u\{\operatorname{prox}_{\gamma f}[2\operatorname{prox}_{\gamma g}(\boldsymbol{x}_u) - \boldsymbol{x}_u] - \operatorname{prox}_{\gamma g}(\boldsymbol{x}_u)\}, \tag{2.11}$$

where $(t_u)_{u\geq 0} \subset [0,2]$ satisfies $\sum_{u\geq 0} t_u(2-t_u) = +\infty$ and the proximity operator [Mor62] of index γ of $f \in \Gamma_0(\mathcal{H})$ is defined as

$$\operatorname{prox}_{\gamma f} : \mathcal{H} \to \mathcal{H} : \boldsymbol{x} \mapsto \operatorname*{arg\,min}_{\boldsymbol{y}\in\mathcal{H}} \left\{ f(\boldsymbol{y}) + \frac{1}{2\gamma} \|\boldsymbol{x} - \boldsymbol{y}\|_{\mathcal{H}}^2 \right\}$$

with the norm on $\mathcal{H}$ denoted by $\|\cdot\|_{\mathcal{H}}$. The Douglas-Rachford splitting is also identified as an example of a Mann iteration in[Yam+11]. Indeed, if $\dim(\mathcal{H}) < \infty$, $\left(\operatorname{prox}_{\gamma g}(x_u)\right)_{u\geq 0}$ converges to a minimizer of (2.10) (see e.g., [Yam+11]).

2.3 Summary

In this chapter, we first have introduced the basic definitions of Cayley-Dickson number systems and their algebraic properties. C-D number systems is an algebraically natural higher dimensional generalization of our familiar fields $\mathbb{R}$ and $\mathbb{C}$.

If a C-D number system is extended to the next higher number system, it may lose a useful algebraic property. For example, the quaternion loses the commutativity of multiplication and furthermore the octonion loses the associativity of multiplication.

Eigenvalue problems in C-D number systems also can be considered but we have to treat them carefully since the left and the right eigenvalue problems can be considered because of the lack of commutativity of multiplication in quaternion or higher dimensional C-D number systems.

The rank of hypercomplex matrices are well-defined in at most quaternion. We have shown that inclusion relation does not hold between the sets of low-rank matrices and the set of separately low-rank matrices in C-D domain.

It suggests that by establishing mathematically sound low-rank approximation techniques for hypercomplex matrices or tensors, we could obtain great potential to have variety of novel formulations to approximate multidimensional information in terms of matrices or tensors of algebraically simple structures.

We next have reviewed powerful optimization methods, the adaptive projected subgradient method (APSM) and Douglas-Rachford splitting (DRS).

All methods can be discussed in general Hilbert space and all algorithm in this thesis are based on these algorithms.

3 Cayley-Dickson Linear Systems and Algebraic Translations

In this chapter, we present algebraic translations of hypercomplex vectors or matrices. These translations are designed by taking advantage of the isomorphism between hypercomplex numbers $\mathbb{A}_m$ and real valued vectors $\mathbb{R}^m$. We also clarify useful algebraic properties of this translation. Lastly, we present the relation to prior work in the complex case.

3.1 Hypercomplex Linear System

We consider the following Cayley-Dickson linear system:

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}, \tag{3.1}$$

where $\boldsymbol{y}, \boldsymbol{b} \in \mathbb{A}_m^M$, $\boldsymbol{x} := \sum_{\ell=1}^m \boldsymbol{x}_\ell \mathbf{i}_\ell \in \mathbb{A}_m^N$ ($\boldsymbol{x}_\ell \in \mathbb{R}^N$) and $\boldsymbol{A} := \sum_{\ell=1}^m \boldsymbol{A}_\ell \mathbf{i}_\ell \in \mathbb{A}_m^{M\times N}$ ($\boldsymbol{A}_\ell \in \mathbb{R}^{M\times N}$).

Remark 2 (Complex widely linear model)**.** Suppose that the relation between $\boldsymbol{z} \in \mathbb{C}^N$ and $\boldsymbol{y} \in \mathbb{C}^M$ can be expressed in terms of $\boldsymbol{\Phi}, \boldsymbol{\Psi} \in \mathbb{C}^{N\times M}$ as

$$\begin{aligned} \boldsymbol{y} &= \boldsymbol{\Phi}\boldsymbol{z} + \boldsymbol{\Psi}\boldsymbol{z}^* + \boldsymbol{b} \\ &= [\boldsymbol{\Phi}, \boldsymbol{\Psi}][\boldsymbol{z}^\top, \boldsymbol{z}^\mathsf{H}]^\top + \boldsymbol{b} \end{aligned} \tag{3.2}$$

where $\boldsymbol{z}^* := (\boldsymbol{z}^\mathsf{H})^\top$. The form of (3.2) is often called the *complex widely linear system* [MG09; AH09; Ada+11; PC95; Ada+09; Ben+10] which has been used as a convenient expression for utilizing complex second-order statistics. Obviously the second expression of (3.2) can be interpreted as an example of (3.1) by letting $\boldsymbol{x} = [\boldsymbol{z}^\top, \boldsymbol{z}^\mathsf{H}]^\top$. A natural extension of complex widely linear system to the quaternion case is found, e.g., in [TM10].

3.2 Proposed Translations

We propose an algebraic translation of hypercomplex valued linear systems into real systems. For the system (3.1), A trivial correspondence (mapping) of hypercomplex vectors or matrices to real ones is

$$\widehat{(\cdot)} : \mathbb{A}_m^{M\times N} \to \mathbb{R}^{mM\times N} : \boldsymbol{A} \mapsto \widehat{\boldsymbol{A}} := \begin{bmatrix} \boldsymbol{A}_1 \\ \vdots \\ \boldsymbol{A}_m \end{bmatrix}, \tag{3.3}$$

where $\boldsymbol{A} = \boldsymbol{A}_1\mathbf{i}_1 + \cdots + \boldsymbol{A}_m\mathbf{i}_m \in \mathbb{A}_m^{M\times N}$ and $\boldsymbol{A}_1, \ldots, \boldsymbol{A}_m \in \mathbb{R}^{M\times N}$. This correspondence is just concatenating a real and all imaginary parts in the hypercomplex matrices. Obviously, this mapping is invertible and thus we can also define

$$\widetilde{(\cdot)} : \mathbb{R}^{mM\times N} \to \mathbb{A}_m^{M\times N} : \widehat{\boldsymbol{A}} \mapsto \boldsymbol{A}. \tag{3.4}$$

Only in terms of the mappings $\widehat{(\cdot)}$ and $\breve{(\cdot)}$, it is hard to obtain the correspondence of matrix-vector product $\boldsymbol{Ax}$, so we also introduce the following non-trivial mapping:

$$\widetilde{(\cdot)} : \mathbb{A}_m^{M\times N} \to \mathbb{R}^{mM\times mN} : \boldsymbol{A} \mapsto \widetilde{\boldsymbol{A}} := \left[\boldsymbol{L}_M^{(1)\top}\widehat{\boldsymbol{A}}, \ldots, \boldsymbol{L}_M^{(m)\top}\widehat{\boldsymbol{A}}\right], \tag{3.5}$$

where the matrix $\boldsymbol{L}_M^{(\ell)} \in \mathbb{R}^{mM\times mM}$ $(\ell = 1, \ldots, m)$ is defined for the m-dimensional hypercomplex number $\mathbb{A}_m$ as

$$\boldsymbol{L}_M^{(\ell)} = \begin{bmatrix} \delta_{1,1}^{(\ell)}\boldsymbol{I}_M & \delta_{1,2}^{(\ell)}\boldsymbol{I}_M & \cdots & \delta_{1,m}^{(\ell)}\boldsymbol{I}_M \\ -\delta_{2,1}^{(\ell)}\boldsymbol{I}_M & -\delta_{2,2}^{(\ell)}\boldsymbol{I}_M & \cdots & -\delta_{2,m}^{(\ell)}\boldsymbol{I}_M \\ \vdots & \vdots & \ddots & \vdots \\ -\delta_{m,1}^{(\ell)}\boldsymbol{I}_M & -\delta_{m,2}^{(\ell)}\boldsymbol{I}_M & \cdots & -\delta_{m,m}^{(\ell)}\boldsymbol{I}_M \end{bmatrix}, \tag{3.6}$$

with the M-dimensional identity matrix $\boldsymbol{I}_M$ and

$$\delta_{\alpha,\beta}^{(\gamma)} := \begin{cases} 1 & (\text{if } \mathbf{i}_\alpha\mathbf{i}_\beta = \mathbf{i}_\gamma), \\ -1 & (\text{if } \mathbf{i}_\alpha\mathbf{i}_\beta = -\mathbf{i}_\gamma), \\ 0 & (\text{otherwise}). \end{cases} \tag{3.7}$$

By (3.5), the degree of freedom of $\widetilde{\boldsymbol{A}}$ is at most that of $\widehat{\boldsymbol{A}} \in \mathbb{R}^{mM\times N}$. More precisely, $\widetilde{(\cdot)}$ is a mapping onto

$$\begin{aligned} \mathfrak{S}_{\mathbb{A}_m}(M,N) &:= \{\widetilde{\boldsymbol{A}} \in \mathbb{R}^{mM\times mN} | \boldsymbol{A} \in \mathbb{A}_m^{M\times N}\} \\ &= \left\{ \left[\boldsymbol{L}_M^{(1)\top}\boldsymbol{B}, \ldots, \boldsymbol{L}_M^{(m)\top}\boldsymbol{B}\right] \middle| \boldsymbol{B} \in \mathbb{R}^{mM\times N} \right\}. \end{aligned} \tag{3.8}$$

The set $\mathfrak{S}_{\mathbb{A}_m}(M,N)$ will play important roles in Chapter 7 and 8. Using the imaginary unit vector $\boldsymbol{i} := [\mathbf{i}_1, \mathbf{i}_2, \ldots, \mathbf{i}_m]^\top \in \mathbb{A}_m^m$, $\boldsymbol{L}_M^{(\ell)}$ is also compactly represented as

$$\boldsymbol{L}_M^{(\ell)} = \Im_\ell(\boldsymbol{i}\boldsymbol{i}^{\mathsf{H}} \otimes \boldsymbol{I}_M),$$

where '$\otimes$' is the Kronecker product. Similar to the trivial mapping, $\widetilde{(\cdot)}$ is also invertible and thus we can define

$$\underset{\smile}{(\cdot)} : \mathfrak{S}_{\mathbb{A}_m}(M,N) \to \mathbb{A}_m^{M\times N} : \widetilde{\boldsymbol{A}} \mapsto \boldsymbol{A}.$$

Example 2. Consider the quaternion $\mathbb{H}$, $\boldsymbol{L}_M^{(\ell)}$ $(\ell \in \{1,2,3,4\})$ are given as follows:

$$\boldsymbol{L}_M^{(1)} = \begin{bmatrix} \boldsymbol{I}_M & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I}_M & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{I}_M & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{I}_M \end{bmatrix}, \boldsymbol{L}_M^{(2)} = \begin{bmatrix} \mathbf{0} & \boldsymbol{I}_M & \mathbf{0} & \mathbf{0} \\ -\boldsymbol{I}_M & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -\boldsymbol{I}_M \\ \mathbf{0} & \mathbf{0} & \boldsymbol{I}_M & \mathbf{0} \end{bmatrix},$$
$$\boldsymbol{L}_M^{(3)} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \boldsymbol{I}_M & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{I}_M \\ -\boldsymbol{I}_M & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\boldsymbol{I}_M & \mathbf{0} & \mathbf{0} \end{bmatrix}, \boldsymbol{L}_M^{(4)} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{I}_M \\ \mathbf{0} & \mathbf{0} & -\boldsymbol{I}_M & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I}_M & \mathbf{0} & \mathbf{0} \\ -\boldsymbol{I}_M & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix},$$

where $\mathbf{0} \in \mathbb{R}^{M\times M}$ is the $M \times M$ zero matrix.

3.3 Algebraic Properties

For the translations $\widehat{(\cdot)}$, $\widetilde{(\cdot)}$, the matrix $\boldsymbol{L}_M^{(\ell)}$ defined in (3.6), and $\delta_{\alpha,\beta}^{(\gamma)}$ in (3.7), we have the following properties.

Lemma 1. *For all $\mathbb{A}_m$, $\delta_{\alpha,\beta}^{(\gamma)}$ satisfies the following:*

1. $\delta_{\alpha,1}^{(\beta)} = \delta_{1,\alpha}^{(\beta)}$ *for all* $\alpha, \beta \in \{1, \ldots, m\}$.
2. *For all* $\alpha, \beta \in \{1, \ldots, m\}$, $\delta_{\alpha,1}^{(\beta)} = \begin{cases} 1 & (\textit{if } \alpha = \beta), \\ 0 & (\textit{if } \alpha \neq \beta). \end{cases}$
3. $\delta_{1,1}^{(1)} = 1$, $\delta_{\alpha,\alpha}^{(1)} = -1$, $\delta_{\alpha,\alpha}^{(\beta)} = \delta_{\alpha,\beta}^{(\alpha)} = \delta_{\beta,\alpha}^{(\alpha)} = 0$ *for all* $\alpha, \beta \in \{2, \ldots, m\}$.
4. $\delta_{\alpha,\beta}^{(\gamma)} = -\delta_{\alpha,\gamma}^{(\beta)} = -\delta_{\gamma,\beta}^{(\alpha)} = -\delta_{\beta,\alpha}^{(\gamma)}$ *if* $\alpha, \beta, \gamma \in \{2, \ldots, m\}$ *are distinct.*

5. *For all* $\alpha, \beta \in \{1, \ldots, m\}$*, there exists unique* $\gamma \in \{1, \ldots, m\}$ *s.t.* $\delta^{(\zeta)}_{\alpha,\beta} = 0$ $(\forall \zeta \neq \gamma)$ *and* $\delta^{(\gamma)}_{\alpha,\beta} = \begin{cases} +1 \\ \text{or} \\ -1, \end{cases}$ *and hence* $\mathbf{i}_\alpha \mathbf{i}_\beta = \sum_{\gamma=1}^{m} \delta^{(\gamma)}_{\alpha,\beta} \mathbf{i}_\gamma$.

6. $\delta^{(1)}_{\alpha,\beta} = 0$ $(\alpha \neq \beta)$ *for all* $\alpha, \beta \in \{1, \ldots, m\}$.

Proof. We prove only 4) and 6) because it is not difficult to verify other properties by using Fact 1.

- (Proof of 4))
  Without loss of generality, we only prove $\delta^{(\gamma)}_{\alpha,\beta} = -\delta^{(\beta)}_{\alpha,\gamma}$. Let us consider the case: $\delta^{(\gamma)}_{\alpha,\beta} =: \sigma \in \{1, -1\}$, which implies $\mathbf{i}_\alpha \mathbf{i}_\beta = \sigma \mathbf{i}_\gamma$. If we multiple $\mathbf{i}_\alpha$ to both sides from left and by using Fact 1-4), we have $-\mathbf{i}_\beta = \mathbf{i}_\alpha(\mathbf{i}_\alpha \mathbf{i}_\beta) = \sigma \mathbf{i}_\alpha \mathbf{i}_\gamma$ (by Fact 1-1)). Then we have $\mathbf{i}_\alpha \mathbf{i}_\gamma = -\sigma \mathbf{i}_\beta$ which is equivalent to $\delta^{(\beta)}_{\alpha,\gamma} = -\sigma = -\delta^{(\gamma)}_{\alpha,\beta}$. On the other hand, consider the remaining case: $\delta^{(\gamma)}_{\alpha,\beta} = 0$. If $\delta^{(\beta)}_{\alpha,\gamma} = \sigma \in \{1, +1\}$, we have $\delta^{(\gamma)}_{\alpha,\beta} = -\sigma \neq 0$ and this is contradict to $\delta^{(\gamma)}_{\alpha,\beta} = 0$, and thus $\delta^{(\beta)}_{\alpha,\gamma} = 0 = -\delta^{(\gamma)}_{\alpha,\beta}$. Therefore, $\delta^{(\gamma)}_{\alpha,\beta} = -\delta^{(\beta)}_{\alpha,\gamma}$ holds in all cases.

- (Proof of 6))
  If $\alpha = 1$ or $\beta = 1$, it is trivial. Suppose that $\alpha \geq 2$ and $\beta \geq 2$. Assume $\delta^{(1)}_{\alpha,\beta} = 1$ or -1, which implies $\mathbf{i}_\alpha \mathbf{i}_\beta = 1$ or -1. In this case, if we multiple $\mathbf{i}_\alpha$ to both sides from left and by using Fact 1-4), we have $-\mathbf{i}_\beta = \mathbf{i}_\alpha(\mathbf{i}_\alpha \mathbf{i}_\beta) = \mathbf{i}_\alpha$ or $-\mathbf{i}_\alpha$. This result contradicts the independence of each imaginary unit. Hence $\delta^{(1)}_{\alpha,\beta} = 0$ $(\alpha \neq \beta)$ for all $\alpha, \beta \in \{1, \ldots, m\}$.

□

Lemma 2. *(Properties of* $\boldsymbol{L}^{(\ell)}_M$*)*

1. *For any* $\mathbb{A}_m$*,* $\boldsymbol{L}^{(1)}_M = \boldsymbol{I}_{mN}$.

2. *For any* $\mathbb{A}_m$*,* $\boldsymbol{L}^{(\ell)\top}_M = \boldsymbol{L}^{(\ell)-1}_M$ $(\ell = 1, \ldots, m)$.

3. *Suppose that* $\mathbb{A}_m$ *is associative, i.e.,* $(ab)c = a(bc)$ *for all* $a, b, c \in \mathbb{A}_m$ *(Associative C-D number systems are only* $\mathbb{R}, \mathbb{C}$ *and* $\mathbb{H}$*). Then for any* $\alpha, \beta \in \{1, 2, \ldots, m\}$*, we have*
$$\boldsymbol{L}^{(\alpha)}_M \boldsymbol{L}^{(\beta)}_M = \begin{cases} \boldsymbol{L}^{(\gamma)}_M & (\text{if } \mathbf{i}_\alpha \mathbf{i}_\beta = \mathbf{i}_\gamma), \\ -\boldsymbol{L}^{(\gamma)}_M & (\text{if } \mathbf{i}_\alpha \mathbf{i}_\beta = -\mathbf{i}_\gamma). \end{cases}$$

Proof. See Appendix A.2. □

Remark 3. For $\mathbb{A}_m = \mathbb{O}$, we observe that $\boldsymbol{L}_M^{(2)}\boldsymbol{L}_M^{(3)} \neq \boldsymbol{L}_M^{(\ell)}$ $(\ell = 1, \dots, 8)$ as a counter example of Lemma 2-3) for non-associative C-D number.

The following theorem collects useful algebraic properties of the proposed translations $\widehat{(\cdot)}$ and $\widetilde{(\cdot)}$. In particular, by using these two translations jointly, we can deduce nontrivial but very useful algebraic relation for multiplication of two hypercomplex matrices (see 3)-5)).

Theorem 1 (Algebraic correspondence between real and hypercomplex vectors/matrices). *If the hypercomplex number system $\mathbb{A}_m$ is constructed by the C-D construction, the following relations hold true:*

1. *For all $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{A}_m^{M\times N}$ and $\alpha \in \mathbb{R}$, $\widehat{(\boldsymbol{A}+\boldsymbol{B})} = \widehat{\boldsymbol{A}} + \widehat{\boldsymbol{B}}$, $\widehat{(\alpha\boldsymbol{A})} = \alpha\widehat{\boldsymbol{A}}$ and $\widetilde{(\boldsymbol{A}+\boldsymbol{B})} = \widetilde{\boldsymbol{A}} + \widetilde{\boldsymbol{B}}$, $\widetilde{(\alpha\boldsymbol{A})} = \alpha\widetilde{\boldsymbol{A}}$.*

2. *$\widetilde{(\boldsymbol{A}^{\mathsf{H}})} = \widetilde{\boldsymbol{A}}^{\top}$ for all $\boldsymbol{A} \in \mathbb{A}_m^{M\times N}$, where $(\sum_{\ell=1}^m \boldsymbol{A}_\ell \mathbf{i}_\ell)^{\mathsf{H}} := \boldsymbol{A}_1^{\top}\mathbf{i}_1 - \sum_{\ell=2}^m \boldsymbol{A}_\ell \mathbf{i}_\ell^{\top}$.*

3. *$\widehat{(\boldsymbol{A}\boldsymbol{B})} = \widetilde{\boldsymbol{A}}\widehat{\boldsymbol{B}}$ for all $\boldsymbol{A} \in \mathbb{A}_m^{M\times N}$ and $\boldsymbol{B} \in \mathbb{A}_m^{N\times L}$.*

4. *$\widehat{(\boldsymbol{A}\boldsymbol{x})} = \widetilde{\boldsymbol{A}}\widehat{\boldsymbol{x}}$ for all $\boldsymbol{A} \in \mathbb{A}_m^{M\times N}$ and $\boldsymbol{x} \in \mathbb{A}_m^N$.*

5. *$\widehat{(\boldsymbol{x}^{\mathsf{H}}\boldsymbol{y})} = \widehat{\boldsymbol{x}}^{\top}\widehat{\boldsymbol{y}}$ for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{A}_m^N$.*

6. *$\|\boldsymbol{x}\|_{\mathbb{A}_m^N} = \|\widehat{\boldsymbol{x}}\|_{\mathbb{R}^{mN}}$ for all $\boldsymbol{x} \in \mathbb{A}_m^N$.*

7. *$\widehat{P_C^{\mathbb{A}_m}(\boldsymbol{x})} = P_{\widehat{C}}^{\mathbb{R}}(\widehat{\boldsymbol{x}})$ for any closed convex set $C \subset \mathbb{A}_m^N$ and any $\boldsymbol{x} \in \mathbb{A}_m^N$, where $\widehat{C} := \{\widehat{\boldsymbol{x}} \in \mathbb{R}^{mN} | \boldsymbol{x} \in C\}$.*

8. *If $m \leq 4$ i.e., $\mathbb{A}_m = \mathbb{R}, \mathbb{C}$ or $\mathbb{H}$, the following holds*

$$\widetilde{(\boldsymbol{A}\boldsymbol{B})} = \widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{B}} \text{ for all } \boldsymbol{A}, \boldsymbol{B} \in \mathbb{A}_m^{M\times N}. \tag{3.9}$$

Proof. See Appendix A.3. □

Remark 4 (Isomorphism between real vector spaces). Let us regard $\mathbb{A}_m^{M\times N}$ as a vector space over $\mathbb{R}$, i.e., a real vector space (see Section 2.1.2). Then, Theorem 1-1) implies $\widehat{(\cdot)}$ is an isomorphism [HJ85] between the real vector spaces $\mathbb{A}_m^{M\times N}$ and $\mathbb{R}^{mM\times N}$. Moreover, Lemma 2-2) and Theorem 1-1) guarantee $\widetilde{(\cdot)}$ is an isomorphism between the real vector spaces $\mathbb{A}_m^{M\times N}$ and $S_{\mathbb{A}_m}(M, N)$.

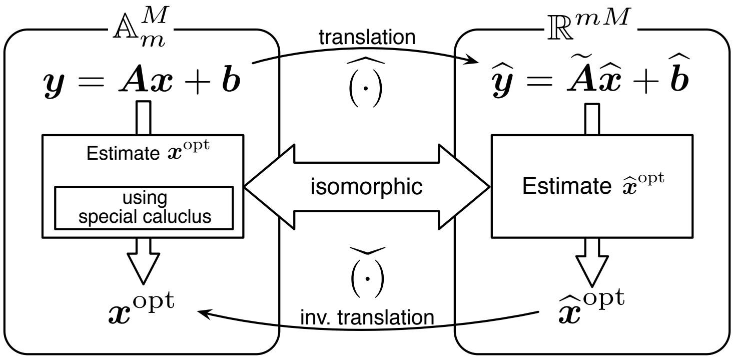


Figure 3.1.: A utilization of proposed translation

Remark 5. The relation (3.9) does not hold for the octonion $\mathbb{O}$ and higher dimensional hypercomplex number systems. Suppose contrary that (3.9) holds for $\mathbb{A}_m$ $(m \geq 8)$. Then we have $(\widetilde{\boldsymbol{A}(\boldsymbol{BC})}) = \widetilde{\boldsymbol{A}}(\widetilde{\boldsymbol{B}}\widetilde{\boldsymbol{C}}) = (\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{B}})\widetilde{\boldsymbol{C}} = (\widetilde{(\boldsymbol{AB})\boldsymbol{C}})$ because of the associativity of real matrices, and thus $\boldsymbol{A}(\boldsymbol{BC}) = (\boldsymbol{AB})\boldsymbol{C}$ for all hypercomplex valued matrices $\boldsymbol{A} \in \mathbb{A}_m^{M\times N}$, $\boldsymbol{B} \in \mathbb{A}_m^{N\times L}$ and $\boldsymbol{C} \in \mathbb{A}_m^{L\times K}$. This contradicts the non-associativity of the octonion $\mathbb{O}$ [Bae01] and higher dimensional hypercomplex number systems.

Based on Theorem 1, we propose an algebraic translation of hypercomplex linear model (3.1) into the following real vector valued linear model:

$$\widehat{\boldsymbol{y}} = \widetilde{\boldsymbol{A}}\widehat{\boldsymbol{x}} + \widehat{\boldsymbol{b}} \in \mathbb{R}^{mM}. \tag{3.10}$$

Fig. 3.1 illustrates a utilization of the proposed translation. A hypercomplex linear system (3.1) is once translated into an equivalent real vector valued linear model (3.10) by the mapping $\widehat{(\cdot)}$. Suppose that we obtain a real vector $\widehat{\boldsymbol{x}}^{\rm opt} \in \mathbb{R}^{mM}$ by applying some computations to (3.10). Then we also obtain the corresponding hypercomplex vector $\boldsymbol{x}^{\rm opt} \in \mathbb{A}_m^M$ by applying the inverse mapping $\breve{(\cdot)}$ to $\widehat{\boldsymbol{x}}^{\rm opt}$. By these steps, the proposed translation enables us to obtain desired estimate $\boldsymbol{x}^{\rm opt}$ without using any calculus designed specially for each hypercomplex number system.

Moreover, we also remark that the proposed translations also enable us to calculate linear system (3.1) without using any packages for computing hypercomplex numbers such as quaternion toolbox for MATLAB® [SB13].

3.4 Computational Complexity

The computational complexity for the mapping $\widetilde{(\cdot)}$ can be evaluated by the complexity for $\boldsymbol{L}_M^{(\ell)\top}\widehat{\boldsymbol{A}}$ $(\ell = 1, \ldots, m)$. As observed in the definition of $\boldsymbol{L}_M^{(\ell)}$, all block elements of $\boldsymbol{L}_M^{(\ell)}$ are $\mathbf{0}_{M\times M}$ or $\pm\boldsymbol{I}_M$. Moreover, only one block matrix of each block column in $\boldsymbol{L}_M^{(\ell)}$ is $\pm\boldsymbol{I}_M$, so the complexity for $\boldsymbol{L}_M^{(\ell)\top}\widehat{\boldsymbol{A}}$ is $\mathcal{O}(mMN)$ real-scalar-multiplications. Therefore, the computational complexity for the mapping $\widetilde{(\cdot)}$ is $\mathcal{O}(m^2MN)$ for $\boldsymbol{A}\in\mathbb{A}_m^{M\times N}$.

We compare the computational complexities for $\boldsymbol{A}\boldsymbol{x}$ and $\widetilde{\boldsymbol{A}}\widehat{\boldsymbol{x}}$, where $\boldsymbol{A}\in\mathbb{A}_m^{M\times N}$ and $\boldsymbol{x}\in\mathbb{A}_m^N$. According to (2.2), multiplication ab ($a,b\in\mathbb{A}_m$) requires $\mathcal{O}(m^2)$ real-scalar-multiplications. Hence $\boldsymbol{A}\boldsymbol{x}$ requires $\mathcal{O}(m^2MN)$ real-scalar-multiplications. On the other hand, $\widetilde{\boldsymbol{A}}\widehat{\boldsymbol{x}}$ requires $\mathcal{O}(m^2MN)$ real-scalar-multiplications. Therefore, the computational complexity of the matrix multiplication is unchanged by the proposed translation.

3.5 Relation to Wirtinger calculus

The Wirtinger calculus[1] ($\mathbb{CR}$-calculus) [Wir27; KD09] have been established for a systematic use of real differentiability of the cost function which is not analytic as a univariate complex valued function, i.e., Cauchy-Riemann equation is not satisfied, and has been applied widely in the complex valued signal processing [AH09; SS10; MG09; Hjø11]. We present a relation between the proposed translation and the Wirtinger calculus.

Fact 3 (Wirtinger ($\mathbb{CR}$-) calculus [Bra83]). *Let $f:\mathbb{R}\times\mathbb{R}\to\mathbb{C}$ be a function of real variables x and y, and $g:\mathbb{C}\times\mathbb{C}\to\mathbb{C}$ be a function of two complex variables z and z^* such that $g(z,z^*)=f(x,y)$, where $z=x+\imath y$, $z^*=x-\imath y$ and that $g(z,z^*)$ is analytic w.r.t. z^* and z independently. Then,*

(a) The partial (Wirtinger) derivatives [Wir27]

$$\frac{\partial g}{\partial z}:=\frac{1}{2}\left(\frac{\partial f}{\partial x}-\imath\frac{\partial f}{\partial y}\right),\ \frac{\partial g}{\partial z^*}:=\frac{1}{2}\left(\frac{\partial f}{\partial x}+\imath\frac{\partial f}{\partial y}\right)$$

can be computed by treating z^* and z as a constant in $f(z,z^*)$ respectively; and

(b) If f is a real valued function, a necessary and sufficient condition for f to have a stationary point is that $\partial g/\partial z=0$. Similarly, $\partial g/\partial z^*=0$ is also a necessary and sufficient condition for f to have a stationary point.

[1] Wirtinger is the name of an Austrian mathematician. In the engineering literature, the Wirtinger calculus was rediscovered in [Bra83] and then extended to the complex valued gradient/Hessian, and the general complex Hilbert space in [Bos94] and [Bou+12], respectively.

Definition 1 (Wirtinger subgradient)**.** Let $\boldsymbol{x} \in \mathbb{C}^N$ and $f : \mathbb{R}^{2N} \to \mathbb{R}$ be a continuous convex function. Let

$$\nabla^s f(\widehat{\boldsymbol{x}}) := \begin{bmatrix} \nabla_r^s f(\widehat{\boldsymbol{x}}) \\ \nabla_i^s f(\widehat{\boldsymbol{x}}) \end{bmatrix} \in \{\boldsymbol{s} \in \mathbb{R}^{2N} | \langle \boldsymbol{x} - \boldsymbol{y}, \boldsymbol{s} \rangle_{\mathbb{R}^{2N}} + f(\boldsymbol{y}) \leq f(\boldsymbol{x}), \forall \boldsymbol{x} \in \mathbb{R}^{2N}\} \tag{3.11}$$

be a *subgradient* of f at $\widehat{\boldsymbol{x}} \in \mathbb{R}^{2N}$, where $\nabla_r^s f(\widehat{\boldsymbol{x}}), \nabla_i^s(\widehat{\boldsymbol{x}}) \in \mathbb{R}^N$. Due to the isomorphism between $\mathbb{C}^N$ and $\mathbb{R}^{2N}$, the function f can also be identified with $g : \mathbb{C}^N \to \mathbb{R} : \boldsymbol{x} \mapsto f(\widehat{\boldsymbol{x}})$. Then, the *Writinger subgradient* of f at $\boldsymbol{x}$ is given by [Bou+12]

$$\nabla_w^s f(\boldsymbol{x}) := \frac{1}{2} \left(\nabla_r^s f(\widehat{\boldsymbol{x}}) - \imath \nabla_i^s f(\widehat{\boldsymbol{x}}) \right)$$

and the *conjugate* Wirtinger subgradient of f at $\boldsymbol{x}$ is given by

$$\nabla_{w^*}^s f(\boldsymbol{x}) := \frac{1}{2} \left(\nabla_r^s f(\widehat{\boldsymbol{x}}) + \imath \nabla_i^s f(\widehat{\boldsymbol{x}}) \right).$$

The set of all conjugate Wirtinger subgradients of f at $\boldsymbol{x}$ is called the *Wirtinger subdifferential* of f at $\boldsymbol{x}$. These are natural extensions of the Wirtinger derivative.

Here, we show a relation between the proposed translation and Wirtinger calculus. Let $\boldsymbol{x} \in \mathbb{C}^N$ and $f : \mathbb{R}^{2N} \to \mathbb{R} : \widehat{\boldsymbol{x}} \mapsto f(\widehat{\boldsymbol{x}})$ be a continuous convex[2] function. Let $\nabla^s f(\widehat{\boldsymbol{x}}) \in \mathbb{R}^{2N}$ be the subgradient of f at $\widehat{\boldsymbol{x}} \in \mathbb{R}^{2N}$. Due to the isomorphism between $\mathbb{C}^N$ and $\mathbb{R}^{2N}$, the function f can also be regarded as $g : \mathbb{C}^N \to \mathbb{R} : \boldsymbol{x} \mapsto g(\boldsymbol{x}) = f(\widehat{\boldsymbol{x}})$. Then, we have the following facts:

Fact 4 (Relation between the proposed algebraic translation and Wirtinger calculus [Yuk+13])**.** *Let $\nabla_w^s g(\boldsymbol{x}) \in \mathbb{C}^N$ and $\nabla_{w^*}^s g(\boldsymbol{x}) \in \mathbb{C}^N$ respectively be the Wirtinger and conjugate Wirtinger subgradient of f at $\boldsymbol{x} \in \mathbb{C}^N$. Then, the following relations hold.*

1. $2(\widehat{\nabla_w^s g(\boldsymbol{x})})^* = \nabla^s f(\widehat{\boldsymbol{x}})$,
2. $2(\widehat{\nabla_{w^*}^s g(\boldsymbol{x})}) = \nabla^s f(\widehat{\boldsymbol{x}})$.

This fact indicates that applying the Wirtinger calculus to real valued complex function can be corresponded to applying the ordinary real calculus to the isomorphic real function. Therefore, this special calculus and the ordinary real calculus are essentially the same operator.

[2] A function $f : \mathbb{R}^N \to \mathbb{R}$ is said to be *convex* if $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^N$ and $\forall \nu \in (0, 1)$, $f(\nu \boldsymbol{x} + (1 - \nu)\boldsymbol{y}) \leq \nu f(\boldsymbol{x}) + (1 - \nu) f(\boldsymbol{y})$.

3.6 Summary

We have presented algebraic translations of hypercomplex vectors or matrices. These translations are designed by taking advantage of the isomorphism between hypercomplex numbers $\mathbb{A}_m$ and real vectors $\mathbb{R}^m$. We also have clarified many useful algebraic properties of this translation and relation to prior work in the complex case.

4 Hypercomplex Online Learning with Algebraic Translations

In Chapter 3, we have proposed an algebraic translation of hypercomplex linear systems and shown that any C-D linear system in the form of (3.1) can be reduced to the equivalent real model (3.10). In this section, as an application to adaptive filtering problem, we propose a new hypercomplex adaptive algorithm.

4.1 Adaptive Filtering Problem

Let us elaborate on the following hypercomplex adaptive filtering estimation problem. Let $\boldsymbol{u}_k := [u_k, u_{k-1}, \ldots, u_{k-N+1}]^\top \in \mathbb{A}_m^N$ be the input vector and $\boldsymbol{U}_k := [\boldsymbol{u}_k, \boldsymbol{u}_{k-1}, \ldots, \boldsymbol{u}_{k-r+1}] \in \mathbb{A}_m^{N\times r}$ be the input matrix at time k. Let also $n_k \in \mathbb{A}_m$ denote the noise process. If $\boldsymbol{h}^\star := [h_1^\star, h_2^\star, \ldots, h_N^\star]^\top \in \mathbb{A}_m^N$ be the unknown system to be estimated, and $\boldsymbol{n}_k := [n_k, n_{k-1}, \ldots, n_{n-r+1}]^\top \in \mathbb{A}_m^r$ be the noise at time k, we introduce the following hypercomplex linear model for the data process $\boldsymbol{d}_k := [d_k, d_{k-1}, \ldots, d_{n-r+1}]^\top \in \mathbb{A}_m^r$:

$$\boldsymbol{d}_k := \boldsymbol{U}_k^H \boldsymbol{h}^\star + \boldsymbol{n}_k. \tag{4.1}$$

A diagram of the adaptive filtering problem is drawn in Fig. 4.1. By using Theorem 1, we immediately obtain the following real valued data process $\widehat{\boldsymbol{d}}_k \in \mathbb{R}^{mr}$:

$$\widehat{\boldsymbol{d}}_k := \widetilde{\boldsymbol{U}}_k^\top \widehat{\boldsymbol{h}}^\star + \widehat{\boldsymbol{n}}_k. \tag{4.2}$$

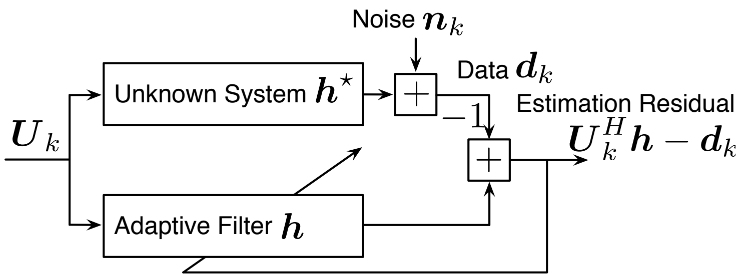


Figure 4.1.: Adaptive filtering scheme

Hence the goal of hypercomplex adaptive filtering problem is reduced to approximating the real valued unknown system $\widehat{\boldsymbol{h}}^{\star} \in \mathbb{R}^{mN}$ by the real valued adaptive filter $\widehat{\boldsymbol{h}}_n \in \mathbb{R}^{mN}$ with the knowledge on $(\widetilde{\boldsymbol{U}}_k, \widehat{\boldsymbol{d}}_k) \in \mathbb{R}^{mN\times mr} \times \mathbb{R}^{mr}$, $\forall k < n$. Note that at any time we can obtain the corresponding hypercomplex adaptive filter $\boldsymbol{h}_n \in \mathbb{A}_m^N$ to $\widehat{\boldsymbol{h}}_n$ by (3.4). This is completely the same form as the real valued adaptive filtering problem, so we can directly apply fairly general methods established in the real domain.

4.2 Proposed Adaptive Algorithms

Here, we propose adaptive algorithms based on the *adaptive projected subgradient method* (APSM). The APSM has been proposed as an efficient algorithm for asymptotic minimization of a certain sequence of nonnegative convex functions.

Let $\Theta_k : \mathbb{A}_m^N \to [0, \infty)$ $(k \in \mathbb{N})$ be a sequence of continuous convex[1] functions and $\partial\Theta_k(\boldsymbol{y})$ be the *subdifferential*[2] of Θ_k at $\boldsymbol{y} \in \mathbb{A}_m^N$. The $\mathbb{A}_m$-APSM provides a vector sequence which minimizes asymptotically the sequence of functions Θ_k over a closed convex set $C \subset \mathbb{A}_m^N$.

Definition 2 ($\mathbb{A}_m$-adaptive projected subgradient method ($\mathbb{A}_m$-APSM))**.** For an arbitrarily given $\boldsymbol{h}_0 \in C$, the $\mathbb{A}_m$-APSM produces a sequence $(\boldsymbol{h}_k)_{k\in\mathbb{N}}$ by

$$\boldsymbol{h}_{k+1} = \begin{cases} P_C^{\mathbb{A}_m}\left(\boldsymbol{h}_k - \lambda_k \frac{\Theta_k(\boldsymbol{h}_k)}{\|\Theta_k'(\boldsymbol{h}_k)\|_{\mathbb{A}_m^N}^2}\Theta_k'(\boldsymbol{h}_k)\right) & \\ \qquad\qquad\qquad\qquad\qquad (\text{if } \Theta_k'(\boldsymbol{h}_k) \neq \boldsymbol{0}), \\ P_C^{\mathbb{A}_m}(\boldsymbol{h}_k) \qquad\qquad\qquad\qquad (\text{otherwise}), \end{cases}$$

where $\Theta_k'(\boldsymbol{h}_k) \in \partial\Theta_k(\boldsymbol{h}_k)$, $0 \leq \lambda_k \leq 2$.

Theorem 2 (Properties of $\mathbb{A}_m$-APSM)**.** *Similar to the real valued case [Yam03; YO04], the sequence $(\boldsymbol{h}_k)_{k\in\mathbb{N}}$ produced by $\mathbb{A}_m$-APSM satisfies the following properties:*

1. *(Monotone approximation)*

$$\|\boldsymbol{h}_{k+1} - \boldsymbol{h}^{\star(k)}\|_{\mathbb{A}_m^N} \leq \|\boldsymbol{h}_k - \boldsymbol{h}^{\star(k)}\|_{\mathbb{A}_m^N},$$
$$\boldsymbol{h}^{\star(k)} \in \Omega_k := \{\boldsymbol{h} \in C | \Theta_k(\boldsymbol{h}) = \inf_{\boldsymbol{x}\in C} \Theta_k(\boldsymbol{x})\}.$$

[1]A function $\Theta : \mathbb{A}_m^N \to \mathbb{R}$ is said to be *convex* if $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{A}_m^N$ and $\forall \nu \in (0, 1)$, $\Theta(\nu\boldsymbol{x} + (1-\nu)\boldsymbol{y}) \leq \nu\Theta(\boldsymbol{x}) + (1-\nu)\Theta(\boldsymbol{y})$.

[2]Similar to the real case in (3.11), we define the *subdifferential* of Θ at $\boldsymbol{y}$ is the set of all the *subgradient* of Θ at $\boldsymbol{y}$; $\partial\Theta(\boldsymbol{y}) := \{\boldsymbol{s} \in \mathbb{A}_m^N | \langle \widehat{\boldsymbol{x}} - \widehat{\boldsymbol{y}}, \widehat{\boldsymbol{s}} \rangle_{\mathbb{R}} + \Theta(\boldsymbol{y}) \leq \Theta(\boldsymbol{x}), \forall \boldsymbol{x} \in \mathbb{A}_m^N\}$

2. *(Asymptotic optimality) Suppose*

$$\exists N_0 \in \mathbb{N} \text{ s.t. } \begin{cases} \inf_{\boldsymbol{x}} \Theta(\boldsymbol{x}), \forall k \geq N_0 \text{ and} \\ \Omega := \bigcap_{k \geq N_0} \Omega_k \neq \varnothing. \end{cases}$$

Then $(\boldsymbol{h}_k)_{k\in\mathbb{N}}$ *is bounded. Moreover, if we use* $\lambda_n \in [\varepsilon_1, 2-\varepsilon_2] \subset (0,2)$ *for all* k*, we have*

$$\lim_{k\to\infty} \Theta_k(\boldsymbol{h}_k) = 0$$

provided that $(\Theta'_k)_{k\in\mathbb{N}}$ *is bounded.* □

Proof. By using Theorem 1, the $\mathbb{A}_m$-APSM is reduced to the ordinary APSM, so these properties hold according to the ordinary properties of the APSM. □

As a class of the $\mathbb{A}_m$-APSM, we present the following scheme.

Scheme 1 ($\mathbb{A}_m$-adaptive parallel projection ($\mathbb{A}_m$-APP))**.** Let $S_i^{(k)} \subset \mathbb{A}_m^N$, $i \in \mathcal{I}_k \subset \mathbb{Z}$ be closed convex sets. Define the sequence of convex function by

$$\Theta_k(\boldsymbol{x}) = \begin{cases} \frac{1}{L_k} \sum_{i\in\mathcal{I}_k} \omega_i^{(k)} d_{\mathbb{A}_m}(\boldsymbol{h}_k, S_i^{(k)}) d_{\mathbb{A}_m}(\boldsymbol{x}, S_i^{(k)}) \\ \qquad\qquad\qquad\qquad\qquad (\text{if } L_k \neq 0), \\ 0 \qquad\qquad\qquad\qquad\qquad (\text{otherwise}), \end{cases}$$

where $\mathbb{Z}$ is the set of all integers, $L_k := \sum_{i\in\mathcal{I}_k} \omega_i^{(k)} d_{\mathbb{A}_m}(\boldsymbol{h}_k, S_i^{(k)})$, $\sum_{i\in\mathcal{I}_k} \omega_i^{(k)} = 1$, $\{\omega_i^{(k)}\}_{i\in\mathcal{I}_k} \subset (0,1]$. In this case, we have

$$\partial\Theta_k(\boldsymbol{x}) \ni \Theta'_k(\boldsymbol{x}) = \begin{cases} \frac{1}{L_k} \sum_{i\in\mathcal{I}_k} \omega_i^{(k)} (\boldsymbol{x} - P^{\mathbb{A}_m}_{S_i^{(k)}}(\boldsymbol{x})) \\ \qquad\qquad\qquad\qquad (\text{if } L_k \neq 0), \\ \boldsymbol{0} \qquad\qquad\qquad\qquad (\text{otherwise}). \end{cases}$$

By applying (1) to Θ_n with $C \subset \mathbb{A}_m^N$, we deduce a following scheme:

$$\boldsymbol{h}_{k+1} = P_C^{\mathbb{A}_m} \left[\boldsymbol{h}_k + \mu_k \left(\sum_{i\in\mathcal{I}_k} \omega_i^{(k)} P^{\mathbb{A}_m}_{S_i^{(k)}}(\boldsymbol{h}_k) - \boldsymbol{h}_k \right) \right],$$

where $\boldsymbol{h}_0 \in C$, $\mu_k \in [0, 2\mathcal{M}_k]$ and

$$\mathcal{M}_k := \begin{cases} \dfrac{\sum_{i\in\mathcal{I}_k} \omega_i^{(k)} \left\| P^{\mathbb{A}_m}_{S_i^{(k)}}(\boldsymbol{h}_k) - \boldsymbol{h}_k \right\|^2_{\mathbb{A}_m^N}}{\left\| \sum_{i\in\mathcal{I}_k} \omega_i^{(k)} P^{\mathbb{A}_m}_{S_i^{(k)}}(\boldsymbol{h}_k) - \boldsymbol{h}_k \right\|^2_{\mathbb{A}_m^N}} \\ \qquad\qquad (\text{if } \boldsymbol{h}_k \notin \bigcap_{i\in\mathcal{I}_k} S_i^{(k)}), \\ 1 \qquad\qquad\qquad (\text{otherwise}). \end{cases}$$

This is a generalization of Algorithm 1 in [Yam+02], hence it includes many useful adaptive algorithms shown as the following examples.

Example 3. If we set $\mathcal{I}_k = \{k\}$, $C = \mathbb{A}_m^N$, and

$$S_i^{(k)} = V_k := \mathop{\arg\min}_{\boldsymbol{h} \in \mathbb{A}_m^N} \left\| \boldsymbol{U}_k^H \boldsymbol{h} - \boldsymbol{d}_k \right\|_{\mathbb{A}_m^r},$$

Scheme 1 reproduces the $\mathbb{A}_m$*-affine projection algorithm* ($\mathbb{A}_m$-APA) as an extension of the APA [HM75; OU84], in particular, $\mathbb{A}_m$-NLMS if $r = 1$ as an extension of the NLMS [NN67]. As the simplest examples, consider the complex case, i.e., the case where $\mathbb{A}_m = \mathbb{C}$. Then we obtain algorithm which agrees with the *complex affine projection algorithm* ($\mathbb{C}$-APA) [Din08]. Note that the complex widely linear model can be expressed in the form of (4.1) (see Remark 2), so this algorithm also covers *widely linear complex affine projection algorithm* (WL-$\mathbb{C}$-APA) [Xia+10] for noncircular input signals. This algorithm is summarized in Algorithm 1. Note that multiplication of hypercomplex valued matrices and vectors are computed by using Theorem 1 in practice. From the discussion in Section 3.4, the computational complexities of $\mathbb{A}_m$-APA and its real translation counterpart described in Algorithm 1 are same.

Example 4. Scheme 1 reproduces the $\mathbb{A}_m$*-adaptive parallel subgradient projection* ($\mathbb{A}_m$-APSP) if we set $\mathcal{I}_k = \{k, k-1, \ldots, k-q+1\}$, $C = \mathbb{A}_m^N$, and

$$S_i^{(k)} = H_i^-(\boldsymbol{h}_k) := \{\boldsymbol{x} \in \mathbb{A}_m^N | (\widehat{\boldsymbol{x}} - \widehat{\boldsymbol{h}}_k)^\top \nabla g_i(\boldsymbol{h}_k) + g_i(\boldsymbol{h}_k) \leq 0\},$$

where q is the number of parallel processors and $g_i(\boldsymbol{x}) = \|\boldsymbol{U}_i^H \boldsymbol{x} - \boldsymbol{d}_i\|_{\mathbb{A}_m^r}^2 - \rho$, $\forall \rho \geq 0$. Similar to the real valued case, the $\mathbb{A}_m$-APSP uses weighted average of multiple subgradient projection to keep low computational cost of $\mathbb{A}_m$-NLMS as well as to achieve fast and stable convergence even in severely noisy environment. This algorithm is summarized in Algorithm 2. Note that the function $g_k(\boldsymbol{x})$, its gradient $\nabla g_k(\boldsymbol{x})$ and the projection $P_{S_i^{(k)}}^{\mathbb{A}_m}(\boldsymbol{h})$ are computed by using Theorem 1 in practice. Therefore, the stochastic analysis of Scheme 1 is reduced to the analysis, e.g., in [Cho+13].

Algorithm 1: Summary of the $\mathbb{A}_m$-APA ($\mathbb{A}_m$-NLMS for $r = 1$)

Input : $\boldsymbol{U}_k \in \mathbb{A}_m^{N \times r}$, $\boldsymbol{d}_k \in \mathbb{A}_m^r$, $\forall k \in \mathbb{N}$, $\boldsymbol{h}_0$
Initialize $k \leftarrow 0$;
repeat
  $\widehat{\boldsymbol{e}}_k \leftarrow \widetilde{\boldsymbol{U}}_k^\top \widehat{\boldsymbol{h}} - \widehat{\boldsymbol{d}}_k$;
  Choose $\mu_k \in [0, 2]$;
  $\widehat{\boldsymbol{h}}_{k+1} \leftarrow \widehat{\boldsymbol{h}}_k - \mu_k \widetilde{\boldsymbol{U}}_k (\widetilde{\boldsymbol{U}}_k^\top \widetilde{\boldsymbol{U}}_k + \varepsilon \boldsymbol{I}_{mr}) \widehat{\boldsymbol{e}}_k$;
until *convergence*;

Algorithm 2: Summary of the $\mathbb{A}_m$-APSP

Input : $\boldsymbol{U}_k \in \mathbb{A}_m^{N\times r}$, $\boldsymbol{d}_k \in \mathbb{A}_m^r$, $\mathcal{I}_k$, $\omega_i^{(k)}$, $\forall i \in \mathcal{I}_k$ $\forall k \in \mathbb{N}$, $\rho \in \mathbb{R}$, $\boldsymbol{h}_0$
Initialize $k \leftarrow 0$;
repeat
  for *each parallel processor* $i \in \mathcal{I}_k$ **do**
    $g_i(\boldsymbol{h}_k) \leftarrow \left\| \widetilde{\boldsymbol{U}}_i^\top \widehat{\boldsymbol{h}}_k - \widehat{\boldsymbol{d}}_i \right\|_{\mathbb{R}^{mr}}^2 - \rho$;
    $\widehat{\nabla g_i(\boldsymbol{h}_k)} \leftarrow 2\widetilde{\boldsymbol{U}}_i(\widetilde{\boldsymbol{U}}_i^\top \widehat{\boldsymbol{h}}_k - \widehat{\boldsymbol{d}}_i)$;
    $\widehat{P_{S_i^{(k)}}^{\mathbb{A}_m}(\boldsymbol{h}_k)} \leftarrow \begin{cases} \widehat{\boldsymbol{h}}_k & (\text{if } g_i(\boldsymbol{h}_k) \le 0) \\ \widehat{\boldsymbol{h}}_k - \frac{g_i(\boldsymbol{h}_k)}{\left\|\widehat{\nabla g_i(\boldsymbol{h}_k)}\right\|_{\mathbb{R}^{mN}}^2} \widehat{\nabla g_i(\boldsymbol{h}_k)} & (\text{otherwise}) \end{cases}$;
  end
  $\mathcal{M}_k \leftarrow \begin{cases} 1 & (\text{if } g_i(\boldsymbol{h}_k) \le 0,\ \forall i \in \mathcal{I}_k) \\ \dfrac{\sum_{i\in\mathcal{I}_k} \omega_i^{(k)} \left\| \widehat{P_{S_i^{(k)}}^{\mathbb{A}_m}(\boldsymbol{h}_k)} - \widehat{\boldsymbol{h}}_k \right\|_{\mathbb{R}^{mN}}^2}{\left\| \sum_{i\in\mathcal{I}_k} \omega_i^{(k)} \widehat{P_{S_i^{(k)}}^{\mathbb{A}_m}(\boldsymbol{h}_k)} - \widehat{\boldsymbol{h}}_k \right\|_{\mathbb{R}^{mN}}^2} & (\text{otherwise}) \end{cases}$;
  Choose $\mu_k \in [0, 2\mathcal{M}_k]$;
  $\widehat{\boldsymbol{h}}_{k+1} \leftarrow \widehat{\boldsymbol{h}}_k + \mu_k \left(\sum_{i\in\mathcal{I}_k} \omega_i^{(k)} \widehat{P_{S_i^{(k)}}^{\mathbb{A}_m}(\boldsymbol{h}_k)} - \widehat{\boldsymbol{h}}_k \right)$;
  $k \leftarrow k+1$;
until *convergence*;

Remark 6. Note that by using the proposed translation, we can reproduce in a unified way the $\mathbb{A}_m$-LMS if we compute the gradient ∇J_k at $\boldsymbol{h} = \boldsymbol{h}_k$ of the following time-varying cost function:

$$J_k(\boldsymbol{h}) = (\boldsymbol{u}_k^H \boldsymbol{h} - d_k)(\boldsymbol{u}_k^H \boldsymbol{h} - d_k)^* \tag{4.3}$$

by passing through the proposed translation. This algorithm includes the $\mathbb{C}$-LMS [Wid75] and the $\mathbb{H}$-LMS (iQLMS in [Jah+12]). However, since the exact steepest descent direction $-\nabla J_k(\boldsymbol{h}_k)$ is not used in the original QLMS in [Man+11] but used in the iQLMS, the original QLMS cannot be derived even through the proposed translation. Indeed, [Jah+12] reported that the iQLMS outperforms the original QLMS. On the other hand, by applying the proposed translation based on (3.2) and its quaternion version, we can reproduce the WL-$\mathbb{C}$-LMS [PC95; Ada+11] and the WL-iQLMS [Jah+12].

4.3 Numerical Examples

In this section, we examine the efficiency of the new algorithms we proposed in Section 4 and 5 in the context of Cayley-Dickson valued linear system identification and nonlinear channel equalization problems, respectively.

First, we examine the efficiency of the algorithms proposed in Section 4 in the context of simple C-D valued linear system identification scenarios. The proposed methods are designed in a unified way for the general C-D number systems, so we performed not only complex and quaternion cases but also octonion case as a higher dimensional hypercomplex example.

We use the C-D valued system $\boldsymbol{h}^\star \in \mathbb{A}_m^N$ $(N := 200)$ with coefficients

$$h_k^\star := \alpha_m \sum_{\ell=1}^{m} (-1)^{\ell-1} \left[1 + \cos\left(\frac{2\pi(k-100)}{200\ell}\right)\right] \mathbf{i}_\ell \tag{4.4}$$

$(k = 1, \ldots, 200)$, $\alpha_m \in \mathbb{R}$ is fixed to ensure unit weight norm., e.g., $(\alpha_2, \alpha_4, \alpha_8) = (0.0684, 0.0422, 0.0274)$. This setting is an extension of [AH09] to the general C-D domain. If we set $m = 2$ or $m = 4$ or $m = 8$ we have a complex or quaternion or octonion system, respectively. For each domain, the input signal u_k ($\in \mathbb{C}$ or $\mathbb{H}$ or $\mathbb{O}$) is generated as follows:

$$\begin{aligned}
u_k &:= \sqrt{1-0.1^2} z_{1,k}\mathbf{i}_1 + 0.1 z_{2,k}\mathbf{i}_2 \in \mathbb{C}, \\
u_k &:= z_{1,k}\mathbf{i}_1 + z_{2,k}\mathbf{i}_2 + z_{3,k}\mathbf{i}_3 + z_{4,k}\mathbf{i}_4 \in \mathbb{H}, \\
u_k &:= \textstyle\sum_{\ell=1}^{8} z_{\ell,k}\mathbf{i}_\ell \in \mathbb{O},
\end{aligned}$$

where $z_{\ell,k} \in \mathbb{R}$ $(\ell = 1, \ldots, 8)$ is i.i.d. from real valued Gaussian distribution with mean 0 and variance 1. The noise $n_k \in \mathbb{A}_m$ is zero mean C-D (in this case, complex or quaternion or octonion) valued circular white Gaussian noise, i.e., $n := \sum_{\ell=1}^{m} n_{\ell,k}\mathbf{i}_\ell$, where $n_{k,\ell} \in \mathbb{R}$ $(\ell = 1, \ldots, m)$ follows the white Gaussian. The signal-to-noise ratio SNR $=$ 30 dB, where SNR $:= 10\log_{10}(\mathbb{E}|\boldsymbol{u}_k^H \boldsymbol{h}^\star|^2 / \mathbb{E}|n_k|^2)$ and $\mathbb{E}(\cdot)$ denotes expectation.

In the complex case, we compare the existing $\mathbb{C}$-LMS [Wid75], $\mathbb{C}$-APA [Din08] and $\mathbb{A}_m$-APSP in the complex domain, that is, $\mathbb{C}$-APSP. In the quaternion case, We compare the existing $\mathbb{H}$-LMS (iQLMS in [Jah+12]), $\mathbb{H}$-APA [Jah+13] and $\mathbb{A}_m$-APSP in the complex domain, that is, $\mathbb{H}$-APSP. The $\mathbb{H}$-LMS and $\mathbb{H}$-APA are implemented by using quaternion toolbox for MATLAB® [SB13]. In the octonion case, there is no existing adaptive algorithms in this domain to the best of our knowledge, so we employ the element-wise $\mathbb{H}$-LMS, which estimates independently a half elements of octonion valued system, as an existing method. We compare this method with new octonion valued adaptive algorithms[3] $\mathbb{O}$-LMS and $\mathbb{O}$-APA, which can be easily derived by using the proposed translations. The set of parameters employed in this experiment is listed in Table 4.1. For each domain, the step-sizes of these methods are fixed so that their initial convergence speeds are same.

[3] MATLAB codes are available by request.

Table 4.1.: Parameter settings for experiments in Section 4.3

	$r = 1$	$r = 2$
$\mathbb{C}$-LMS	λ(step-size) $= 0.06$	N/A
$\mathbb{C}$-APA	$\mu_k = 1.0$	$\mu_k = 1.0$
$\mathbb{C}$-APSP	$\mu_k = 2.0$, $q = 10.0$	$\mu_k = 2.0$, $q = 10.0$
$\mathbb{H}$-LMS	λ(step-size) $= 0.001$	N/A
$\mathbb{H}$-APA	$\mu_k = 1.0$	$\mu_k = 1.0$
$\mathbb{H}$-APSP	$\mu_k = 2.0$, $q = 10.0$	$\mu_k = 2.0$, $q = 10.0$
element-wise $\mathbb{H}$-LMS	λ(step-size) $= 0.001$	N/A
$\mathbb{O}$-LMS	λ(step-size) $= 0.001$	N/A
$\mathbb{O}$-APA	$\mu_k = 0.5$	$\mu_k = 0.25$

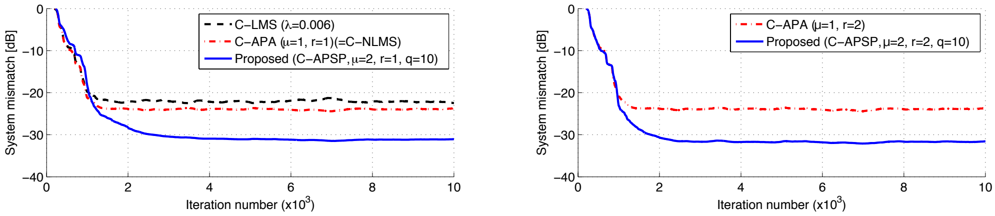


Figure 4.2.: Comparison of the performances of existing and proposed adaptive algorithms in the complex case with $r = 1$ (left figure) and $r = 2$ (right figure).

Fig. 4.2, 4.3 and 4.4 depicts the comparisons of the complex, quaternion and octonion valued adaptive algorithms in the sense of the system-mismatch. The system-mismatch of Cayley-Dickson valued system at k-th iteration $\mathrm{sysmiss}_{\mathbb{A}_m}(k)$ is given by

$$\mathrm{sysmiss}_{\mathbb{A}_m}(k) := 10 \log_{10} \frac{\|\boldsymbol{h}^\star - \boldsymbol{h}_k\|^2_{\mathbb{A}_m^N}}{\|\boldsymbol{h}^\star\|^2_{\mathbb{A}_m^N}} \tag{4.5}$$

and averaged over 300 trials. This measure is a very natural extension of the ordinary (real valued) system mismatch found in many literatures (e.g., [GB00]). In the complex case (Fig. 4.2), the proposed $\mathbb{C}$-APSP achieves the best steady-state performance thanks to complex valued parallel projection which is extended by proposed translations. Similar to this case, the proposed $\mathbb{H}$-APSP achieves the best performance in the quaternion domain (Fig. 4.3). In this case the existing $\mathbb{H}$-LMS achieves better performance than the existing $\mathbb{H}$-APA. This is because the acceleration of convergence speed in the APA affects the steady-state performance. In the octonion case (Fig. 4.4), the newly proposed $\mathbb{O}$-LMS and $\mathbb{O}$-APA reach to steady-state while the element-wise $\mathbb{H}$-LMS, which does not approximate the true system, does not converge. The $\mathbb{O}$-APA with $r = 2$ achieves better steady-state performance than that with $r = 1$ and $\mathbb{O}$-LMS by taking advantage of previous inputs.

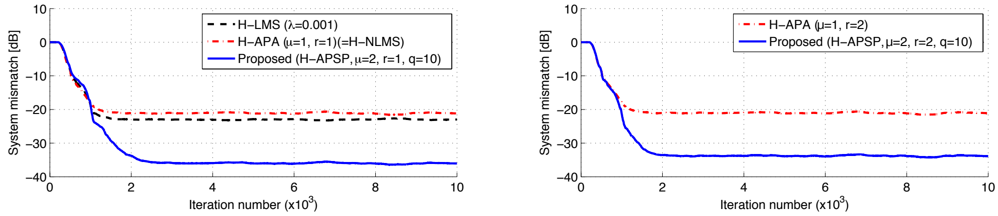


Figure 4.3.: Comparison of the performances of existing and proposed adaptive algorithms in the quaternion case with $r = 1$ (left figure) and $r = 2$ (right figure).

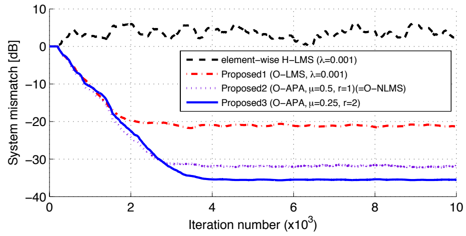


Figure 4.4.: Comparison of the performances of existing and proposed adaptive algorithms in the octonion case with $r = 1$ (dotted) and $r = 2$ (solid).

4.4 Summary

In this chapter, we have formulated a hypercomplex online parameter estimation (adaptive filtering) problem in C-D domain. We then have showed that any parameter estimation problems in C-D linear systems can be reduced to the equivalent ones in real linear systems by utilizing the algebraic real translations proposed in Chapter 3. We next have proposed a new hypercomplex adaptive algorithm named $\mathbb{A}_m$-adaptive projected subgradient method ($\mathbb{A}_m$-APSM) by applying the APSM to the translated real problem. We also have shown that the proposed algorithm is a C-D generalization of the APSM and that it inherits the powerful properties of the original APSM such as monotone approximation and asymptotic optimality. Moreover, similar to the real case, the proposed algorithm includes wide hypercomplex algorithm such as the C-D counterparts of the normalized least mean squares ($\mathbb{A}_m$-NLMS), the affine projection ($\mathbb{A}_m$-APA) and the adaptive parallel subgradient projection (A_m-APSP) as special cases of $\mathbb{A}_m$-APSM. Numerical examples are performed in the context of linear system identification in many C-D domains including complex, quaternion and octonion and shows the effectiveness of the proposed methods.

5 Hypercomplex Nonlinear Estimation with Kernels

5.1 Reproducing Kernel Hilbert Space

Assume a real Hilbert space $(\mathcal{H}, \langle\cdot,\cdot\rangle_{\mathcal{H}}, \|\cdot\|_{\mathcal{H}})$ whose elements are functions defined on $\mathcal{X}$ $(\subset \mathbb{R}^N)$, e.g., $f : \mathcal{X} \to \mathbb{R}$. The Hilbert space $\mathcal{H}$ is called a reproducing kernel Hilbert space (RKHS) [SS01; Aro50] if there exists a function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, called a *reproducing kernel* of $\mathcal{H}$, satisfying the following conditions:

1. For every $\boldsymbol{x} \in \mathcal{X}$, the kernel function $K(\cdot, \boldsymbol{x}) : \mathcal{X} \to \mathbb{R}$ is defined as a point is $\mathcal{H}$. The mapping $\phi : \mathcal{X} \to \mathcal{H} : \boldsymbol{x} \mapsto K(\cdot, \boldsymbol{x})$, translates a sample in data space into a higher dimensional feature space.

2. The *reproducing property* holds:

$$f(\boldsymbol{x}) = \langle f, K(\cdot, \boldsymbol{x})\rangle_{\mathcal{H}}, \quad \forall \boldsymbol{x} \in \mathcal{X}, \forall f \in \mathcal{H}.$$

These properties gurantees $\langle\phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j)\rangle_{\mathcal{H}} = \langle K(\cdot, \boldsymbol{x}_i), K(\cdot, \boldsymbol{x}_j)\rangle_{\mathcal{H}} = \langle K(\cdot, \boldsymbol{x}_j), K(\cdot, \boldsymbol{x}_i)\rangle_{\mathcal{H}} = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$, $\forall \boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{X}$, which is widely known as the *kernel trick*. In the following, we assume K is positive definite i.e., $\sum_{i,j=1}^{L} c_i c_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq 0$ $(\forall c_i, c_j \in \mathbb{R}, \forall \boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{X}$ and any $L \in \mathbb{N})$. The symmetricity of the reproducing kernel, i.e., $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = K(\boldsymbol{x}_j, \boldsymbol{x}_i)$ $\forall \boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{X}$ holds automatically from the kernel trick. The Gaussian kernel $K(\boldsymbol{x}, \boldsymbol{y}) := \exp(-\|\boldsymbol{x} - \boldsymbol{y}\|^2 / \sigma^2)$ $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}$ $(\mathcal{X} = \mathbb{R}^N)$ is a well-known positive definite kernel. The kernel trick has been widely used to transform algorithms expressed in terms of inner products into nonlinear ones. In the studies of machine learning, for example, the support vector machine [Vap95], the principal component analysis [Sch+98], the Fisher discriminant analysis [Mik+99] and so on. Recently, kernel-based online predictions of time series are attracting strong attention [Eng+04; Liu+08].

5.2 Reproducing kernel for C-D domain

Several extensions of RKHS to complex domain are found, e.g., [Pau06; BT11].

In this thesis, we extend this technique to the general C-D domain by using the proposed translation. Let $\mathcal{X} \subset \mathbb{R}^N$, define $\mathcal{X}^m \subset \mathbb{R}^{mN}$ and $\mathcal{X}_{\mathbb{A}_m} := \{\boldsymbol{x}_1\mathbf{i}_1 + \boldsymbol{x}_2\mathbf{i}_2 + \cdots + \boldsymbol{x}_m\mathbf{i}_m | \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m \in \mathcal{X}\} \subset \mathbb{A}_m^N$. Let also $(\mathcal{H}, \langle\cdot,\cdot\rangle_{\mathcal{H}}, \|\cdot\|_{\mathcal{H}})$ be a real RKHS whose elements are real valued functions defined on $\mathcal{X}^m$ and associated with a real kernel $K : \mathcal{X}^m \times \mathcal{X}^m \to \mathbb{R}$. Note that the dimension of data space is m times of that in Section 5.1. Then, every $f \in \mathcal{H}$ can be regarded as a function defined on either $\mathcal{X}^m$ or $\mathcal{X}_{\mathbb{A}_m}$, i.e., for $\boldsymbol{x} := \boldsymbol{x}_1\mathbf{i}_1 + \boldsymbol{x}_2\mathbf{i}_2 + \cdots + \boldsymbol{x}_m\mathbf{i}_m \in \mathcal{X}_{\mathbb{A}_m}$ $f(\boldsymbol{x}) = f(\boldsymbol{x}_1\mathbf{i}_1 + \boldsymbol{x}_2\mathbf{i}_2 + \cdots + \boldsymbol{x}_m\mathbf{i}_m) := f(\widehat{\boldsymbol{x}})$, where $\widehat{(\cdot)}$ is the mapping introduced in (3.3).

Next, we define $\mathcal{H}^m = \underbrace{\mathcal{H} \times \mathcal{H} \times \cdots \times \mathcal{H}}_{m}$. It is easy to verify that $\mathcal{H}^m$ is also a real Hilbert space with inner product

$$\langle \boldsymbol{f}, \boldsymbol{g} \rangle_{\mathcal{H}^m} := \sum_{\ell=1}^{m} \langle f_\ell, g_\ell \rangle_{\mathcal{H}},$$

for $\boldsymbol{f} = [f_1, f_2, \ldots, f_m]^\top \in \mathcal{H}^m$ and $\boldsymbol{g} = [g_1, g_2, \ldots, g_m]^\top \in \mathcal{H}^m$ and its induced norm $\|\boldsymbol{f}\|_{\mathcal{H}^m} := \sqrt{\langle \boldsymbol{f}, \boldsymbol{f} \rangle_{\mathcal{H}^m}}$. Our objective is to enrich $\mathcal{H}$ with a hypercomplex product structure defined by the multiplication table. Here, we define the space $\mathcal{H}_{\mathbb{A}_m} := \{f = f_1\mathbf{i}_1 + f_2\mathbf{i}_2 + \cdots + f_m\mathbf{i}_m | f_1, f_2, \ldots, f_m \in \mathcal{H}\}$. Note that $\mathcal{H}_{\mathbb{A}_m}$ is not a vector space over $\mathbb{A}_m$ in general since octonion $\mathbb{O}$ and higher dimensional hypercomplex number systems are not fields[1]. For this space, we define the following $\mathbb{A}_m$-valued function like an inner product.

$$\langle f, g \rangle_{\mathcal{H}_{\mathbb{A}_m}} := \sum_{\ell=1}^{m} \left\langle \boldsymbol{f}, \boldsymbol{L}^{(\ell)\top} \boldsymbol{g} \right\rangle_{\mathcal{H}^m} \mathbf{i}_\ell, \tag{5.1}$$

for $f = f_1\mathbf{i}_1 + f_2\mathbf{i}_2 + \cdots + f_m\mathbf{i}_m$ and $g = g_1\mathbf{i}_1 + g_2\mathbf{i}_2 + \cdots + g_m\mathbf{i}_m$ (Note: $f, g : \mathcal{X}_{\mathbb{A}_m} (\subset \mathbb{A}_m^N) \to \mathbb{A}_m$). In (5.1), we introduce, in analogy with Section 3.2,

$$\boldsymbol{L}^{(\ell)} := \begin{bmatrix} \delta_{1,1}^{(\ell)} I & \delta_{1,2}^{(\ell)} I & \cdots & \delta_{1,m}^{(\ell)} I \\ -\delta_{2,1}^{(\ell)} I & -\delta_{2,2}^{(\ell)} I & \cdots & -\delta_{2,m}^{(\ell)} I \\ \vdots & \vdots & \ddots & \vdots \\ -\delta_{m,1}^{(\ell)} I & -\delta_{m,2}^{(\ell)} I & \cdots & -\delta_{m,m}^{(\ell)} I \end{bmatrix},$$

with the identity operator $I : \mathcal{H} \to \mathcal{H}$ and $\delta_{\alpha,\beta}^{(\gamma)}$ $(\alpha, \beta, \gamma \in \{1, \ldots, m\})$ defined in (3.7). Then following proposition holds:

Proposition 2. *The function $\langle \cdot, \cdot \rangle_{\mathcal{H}_{\mathbb{A}_m}}$ defined in (5.1) satisfies the following axioms of inner product:*

1. $\langle f, f \rangle_{\mathcal{H}_{\mathbb{A}_m}} \geq 0$ *and* $\langle f, f \rangle_{\mathcal{H}_{\mathbb{A}_m}} = 0 \iff f = 0 \; \forall f \in \mathcal{H}_{\mathbb{A}_m}$,

[1] $\mathcal{H}_{\mathbb{A}_m}$ $(m = 1, 2)$ can be seen as vector spaces over $\mathbb{R}$ or $\mathbb{C}$, respectively.

2. $\langle f, g\rangle_{\mathcal{H}_{\mathbb{A}_m}} = \langle g, f\rangle^*_{\mathcal{H}_{\mathbb{A}_m}}\ \forall f, g \in \mathcal{H}_{\mathbb{A}_m}$,

3. $\langle \alpha f + \beta g, h\rangle_{\mathcal{H}_{\mathbb{A}_m}} = \alpha \langle f, h\rangle_{\mathcal{H}_{\mathbb{A}_m}} + \beta \langle g, h\rangle_{\mathcal{H}_{\mathbb{A}_m}}\ \forall \alpha, \beta \in \mathbb{A}_m, \forall f, g, h \in \mathcal{H}_{\mathbb{A}_m}$.

Moreover, $\mathcal{H}_{\mathbb{A}_m}$ satisfies the following reproducing property:

$$f_1(\boldsymbol{x})\mathbf{i}_1 + f_2(\boldsymbol{x})\mathbf{i}_2 + \cdots + f_m(\boldsymbol{x})\mathbf{i}_m = \langle f_1\mathbf{i}_1 + f_2\mathbf{i}_2 + \cdots + f_m\mathbf{i}_m, K(\widehat{\cdot}, \widehat{\boldsymbol{x}})\rangle_{\mathcal{H}_{\mathbb{A}_m}} \tag{5.2}$$

Proof. See Appendix A.4. □

Remark 7 (on Proposition 2).

- Since $\mathcal{H}_{\mathbb{A}_m}$ is not a vector space over $\mathbb{A}_m$ ($m \geq 4$), so $\langle\cdot,\cdot\rangle_{\mathcal{H}_{\mathbb{A}_m}}$ is not an inner product in the strictly sense even if the above properties hold. For the same reason, $\mathcal{H}_{\mathbb{A}_m}$ is not a Hilbert space in the strictly sense. Fortunately, Proposition 2 guarantees that $(\mathcal{H}_{\mathbb{A}_m}, \langle\cdot,\cdot\rangle_{\mathcal{H}_{\mathbb{A}_m}})$ becomes a Hilbert-like space and posseses the essential properties of RKHS. Therefore, the real kernel K can be used to kernel learning in the C-D domain.

- If we regard $\mathcal{H}_{\mathbb{A}_m}$ as a vector space over $\mathbb{R}$, we can define its norm by $\|f\|_{\mathcal{H}_{\mathbb{A}_m}} := \sqrt{\langle f, f\rangle_{\mathcal{H}_{\mathbb{A}_m}}}$ for $f \in \mathcal{H}_{\mathbb{A}_m}$. Since this norm is exactly same as the norm $\|\cdot\|_{\mathcal{H}^m}$, the real normed space $(\mathcal{H}_{\mathbb{A}_m}, \|\cdot\|_{\mathcal{H}_{\mathbb{A}_m}})$ satisfies completeness, hence a Banach space.

5.2.1 Nonlinear Regression with real Kernels

We consider a nonlinear regression problem with kernels in a C-D domain. Let $\boldsymbol{u}_k \in \mathcal{U}_{\mathbb{A}_m}$ be the input sample, where $\mathcal{U}_{\mathbb{A}_m}$ denotes the input space which is a compact subset, i.e., a closed bounded subset of the finite dimensional normed space $(\mathbb{A}_m^N, \|\cdot\|_{\mathbb{A}_m})$. Let $\mathcal{U}^m := \{\widehat{\boldsymbol{u}} | \boldsymbol{u} \in \mathcal{U}_{\mathbb{A}_m}\} \subset \mathbb{R}^{mN}$ and $K : \mathcal{U}^m \times \mathcal{U}^m \to \mathbb{R}$ be a real kernel function, where $\widehat{(\cdot)}$ is the mapping introduced in (3.3). Let also $\mathcal{H}_{\mathbb{A}_m}$ be the C-D extension of real RKHS $\mathcal{H}$ associated with a real kernel K as defined in Section 5.2. Consider the empirical risk minimization approach, the problem is to determine a function $\psi(\cdot) := \psi_1\mathbf{i}_1 + \cdots + \psi_m\mathbf{i}_m \in \mathcal{H}_{\mathbb{A}_m}$, $\psi_\ell \in \mathcal{H}$ ($\ell = 1, \ldots, m$) that minimizes the sum of squared errors between desired outputs $d_k := d_{1,k}\mathbf{i}_1 + \cdots + d_{m,k} \in \mathbb{A}_m$, $d_{\ell,k} \in \mathbb{R}$ ($\ell = 1, \ldots, m$) and the corresponding output samples of $\psi(\boldsymbol{u}_k) = \langle \psi(\cdot), K(\widehat{\cdot}, \widehat{\boldsymbol{u}}_k)\rangle_{\mathcal{H}_{\mathbb{A}_m}}$, namely

$$\underset{\psi \in \mathcal{H}_{\mathbb{A}_m}}{\text{minimize}} \quad \sum_{k=1}^{n} |\psi(\boldsymbol{u}_k) - d_k|^2 \tag{5.3a}$$

which is equivalent to

$$\underset{\psi_1,\ldots,\psi_m\in\mathcal{H}}{\text{minimize}} \quad \sum_{k=1}^{n}\left(|\psi_1(\boldsymbol{u}_k)-d_{1,k}|^2+\cdots+|\psi_m(\boldsymbol{u}_k)-d_{m,k}|^2\right). \tag{5.3b}$$

By applying the representer theorem [KW71; Sch+01] for each $\ell=1,\ldots,m$, the function $\psi\in\mathcal{H}_{\mathbb{A}_m}$ minimizing (5.3a) can be written as a kernel expansion in terms of data

$$\psi(\cdot)=\sum_{j=1}^{n}K(\widehat{\cdot},\widehat{\boldsymbol{u}}_j)\alpha_j. \tag{5.4}$$

Let $\boldsymbol{K}\in\mathbb{R}^{n\times n}$ be the kernel (Gram) matrix whose (i,j)-th entry is $K(\boldsymbol{u}_i,\boldsymbol{u}_j)$, $\boldsymbol{\alpha}:=\boldsymbol{\alpha}_1\mathbf{i}_1+\cdots+\boldsymbol{\alpha}_m\mathbf{i}_m=[\alpha_1,\ldots,\alpha_n]^\top\in\mathbb{A}_m^n$, $\boldsymbol{\alpha}_\ell\in\mathbb{R}^n$ $(\ell=1,\ldots,m)$ and $\boldsymbol{d}=\boldsymbol{d}_1\mathbf{i}_1+\cdots+\boldsymbol{d}_m\mathbf{i}_m=[d_1,\ldots,d_n]^\top\in\mathbb{A}_m^n$, $\boldsymbol{d}_\ell\in\mathbb{R}^n$ $(\ell=1,\ldots,m)$ then the problem (5.3a) becomes

$$\underset{\boldsymbol{\alpha}\in\mathbb{A}_m^n}{\text{minimize}} \quad \|\boldsymbol{K}\boldsymbol{\alpha}-\boldsymbol{d}\|_{\mathbb{A}_m^n}^2, \tag{5.5a}$$

which is equivalent to

$$\underset{\boldsymbol{\alpha}_1,\ldots,\boldsymbol{\alpha}_m\in\mathbb{R}^n}{\text{minimize}} \quad \|\boldsymbol{K}\boldsymbol{\alpha}_1-\boldsymbol{d}_1\|_{\mathbb{R}^n}^2+\cdots+\|\boldsymbol{K}\boldsymbol{\alpha}_m-\boldsymbol{d}_m\|_{\mathbb{R}^n}^2. \tag{5.5b}$$

The solution $\boldsymbol{\alpha}$ of (5.5b) is obtained by solving the normal equations $\boldsymbol{K}^\top\boldsymbol{K}\alpha_\ell=\boldsymbol{K}^\top\boldsymbol{d}_\ell$ for each $\ell=1,\ldots,m$.

5.3 Kernel Adaptive Filtering on $\mathbb{A}_m$

Next, we formalize the C-D nonlinear adaptive filtering problem. In this section, we reuse the notations introduced in Section 5.2.1. Let $\boldsymbol{u}_k\in\mathcal{U}_{\mathbb{A}_m}$ be the input vector at time k. As observed in (5.4), the main problem with adaptive learning is increasing number of observations with time and the model (5.4) becomes more and more complex. To avoid this problem, a certain sparsification technique. Let $\mathcal{J}_k:=\{j_1^{(k)},j_2^{(k)},\ldots j_{s_k}^{(k)}\}\subset\{1,2,\ldots,k-1\}$ be the index set at time k. The set of kernel functions $\{K(\widehat{\cdot},\widehat{\boldsymbol{u}}_j)\}_{j\in\mathcal{J}_k}$ is called *dictionary*. Note that the dictionary is not always updated when the new data is observed. If we consider the data process $d_k=\psi(\boldsymbol{u}_k)$, where $\psi\in\mathbb{H}_{\mathbb{A}_m}$ is to be estimated. The kernel adaptive filtering problem is to approximate the nonlinear function ψ by

$$\varphi_k(\cdot)=\sum_{j\in\mathcal{J}_k}K(\widehat{\cdot},\widehat{\boldsymbol{u}}_j)\alpha_{j,k}, \tag{5.6}$$

where $\alpha_{j,k}\in\mathbb{A}_m$. The estimate of the data process $d_k':=\varphi_k(\boldsymbol{u}_k)$ can be written in a vector form as follows:

$$d_k':=\varphi_k(\boldsymbol{u}_k)=\boldsymbol{K}_k^\top\boldsymbol{\alpha}_k, \tag{5.7}$$

where $\boldsymbol{K}_k := [K(\widehat{\boldsymbol{u}}_{j_1^{(k)}}, \widehat{\boldsymbol{u}}_k), K(\widehat{\boldsymbol{u}}_{j_2^{(k)}}, \widehat{\boldsymbol{u}}_k), \ldots, K(\widehat{\boldsymbol{u}}_{j_{s_k}^{(k)}}, \widehat{\boldsymbol{u}}_k)]^\top \in \mathbb{R}^{s_k}$ and $\boldsymbol{\alpha}_k = [\alpha_{j_1^{(k)},k}, \alpha_{j_2^{(k)},k}, \ldots, \alpha_{j_{s_k}^{(k)},k}]^\top \in \mathbb{A}_m^{s_k}$. Note that this form is linear w.r.t. $\boldsymbol{\alpha}_k$. Suppose for simplicity that the real kernel K has a unit norm, i.e., $K(\widehat{\boldsymbol{u}}, \widehat{\boldsymbol{u}}) = 1$, $\forall \boldsymbol{u} \in \mathcal{U}$; the Gaussian kernel satisfies this condition. For updating the index set $\mathcal{J}_k$, we employ the *coherent-based criterion* [Ric+09], that is, we add k into $\mathcal{J}_k$ if the following condition is satisfied:

$$\max_{j \in \mathcal{J}_k} |K(\widehat{\boldsymbol{u}}_k, \widehat{\boldsymbol{u}}_j)| \leq \eta, \tag{5.8}$$

where $\eta > 0$ is the threshold. Similar to the real valued case [Ric+09], the compactness of the input space $\mathcal{U}_{\mathbb{A}_m}$ ensures the finite dictionary even for infinite number of input samples. Note that, if $\mathcal{J}_k = \varnothing$, then $\varphi_k(\boldsymbol{u}) := 0$, $\forall \boldsymbol{u} \in \mathcal{U}$, and the condition (5.8) is regarded to be satisfied automatically.

5.4 Proposed Nonlinear Adaptive Algorithms

If we adjust Scheme 1 for this formulation, we obtain the following scheme.

Scheme 2 ($\mathbb{A}_m$-kernel adaptive parallel projection ($\mathbb{A}_m$-KAPP)). Let $S_i^{(k)} \subset \mathbb{A}_m^{s_k}$ and $\bar{S}_i^{(k)} \subset \mathbb{A}_m^{s_k+1}$ ($\forall i \in \mathcal{I}_k \subset \mathbb{Z}$) be closed convex sets, and $\sum_{i \in \mathcal{I}_k} \omega_i^{(k)} = 1$, $\{\omega_i^{(k)}\}_{i \in \mathcal{I}_k} \subset (0, 1]$. The update rules are given as follows:

(1) If (5.8) is unsatisfied, $\mathcal{J}_{k+1} = \mathcal{J}_k$ and

$$\boldsymbol{\alpha}_{k+1} = \boldsymbol{\alpha}_k + \mu_k \left(\sum_{i \in \mathcal{I}_k} \omega_i^{(k)} P_{S_i^{(k)}}^{\mathbb{A}_m}(\boldsymbol{\alpha}_k) - \boldsymbol{\alpha}_k \right), \tag{5.9}$$

(2) If (5.8) is satisfied, $\mathcal{J}_{k+1} = \mathcal{J}_k \cup \{k\}$ and

$$\boldsymbol{\alpha}_{k+1} = \bar{\boldsymbol{\alpha}}_k + \bar{\mu}_k \left(\sum_{i \in \mathcal{I}_k} \omega_i^{(k)} P_{\bar{S}_i^{(k)}}^{\mathbb{A}_m}(\bar{\boldsymbol{\alpha}}_k) - \bar{\boldsymbol{\alpha}}_k \right), \tag{5.10}$$

where $\bar{\boldsymbol{\alpha}}_k := [\boldsymbol{\alpha}_k^\top, 0]^\top \in \mathbb{A}_m^{s_k+1}$, $\mu_k \in [0, 2\mathcal{M}_k(\boldsymbol{\alpha}_k, S_i^{(k)})]$, $\bar{\mu}_k \in [0, 2\mathcal{M}_k(\bar{\boldsymbol{\alpha}}_k, \bar{S}_i^{(k)})]$ and

$$\mathcal{M}_k(\boldsymbol{x}, S_i) := \begin{cases} \dfrac{\sum_{i \in \mathcal{I}_k} \omega_i^{(k)} \left\| P_{S_i}^{\mathbb{A}_m}(\boldsymbol{x}) - \boldsymbol{x} \right\|_{\mathbb{A}_m^s}^2}{\left\| \sum_{i \in \mathcal{I}_k} \omega_i^{(k)} P_{S_i}^{\mathbb{A}_m}(\boldsymbol{x}) - \boldsymbol{x} \right\|_{\mathbb{A}_m^s}^2} \\ \text{(if } \boldsymbol{x} \notin \bigcap_{i \in \mathcal{I}_k} S_i (\subset \mathbb{A}_m^s)), \\ 1 \qquad \text{(otherwise)}. \end{cases}$$

Similar to Scheme 1, this scheme includes the following useful algorithms as simple examples.

Example 5 ($\mathbb{A}_m$-kernel affine projection ($\mathbb{A}_m$-KAPA))**.** Let $\boldsymbol{H}_k \in \mathbb{R}^{s_k \times r}$ is the $s_k \times r$ matrix whose (p, q)-th entry $(\boldsymbol{H}_k)_{p,q} = K(\widehat{\boldsymbol{u}}_{j_p^{(k)}}, \widehat{\boldsymbol{u}}_{k-q+1})$. In (5.9) and (5.10), if we set $\mathcal{I}_k = \{k\}$ and

$$S_i^{(k)} := \underset{\boldsymbol{\alpha} \in \mathbb{A}_m^{s_k}}{\arg\min} \left\| \boldsymbol{H}_k^\top \boldsymbol{\alpha} - \boldsymbol{d}_k \right\|_{\mathbb{A}_m^r}, \tag{5.11}$$

$$\bar{S}_i^{(k)} := \underset{\bar{\boldsymbol{\alpha}} \in \mathbb{A}_m^{s_k+1}}{\arg\min} \left\| \bar{\boldsymbol{H}}_k^\top \bar{\boldsymbol{\alpha}} - \boldsymbol{d}_k \right\|_{\mathbb{A}_m^r}, \tag{5.12}$$

where $\boldsymbol{d}_k = [d_k, d_{k-1}, \ldots, d_{k-r+1}]^\top \in \mathbb{A}_m^r$, $\bar{\boldsymbol{H}}_k := [\boldsymbol{H}_k^\top, \bar{\boldsymbol{h}}_k^\top]^\top \in \mathbb{R}^{(s_k+1)\times r}$, $\bar{\boldsymbol{h}}_k := [K(\widehat{\boldsymbol{u}}_k, \widehat{\boldsymbol{u}}_k), K(\widehat{\boldsymbol{u}}_k, \widehat{\boldsymbol{u}}_{k-1}), \ldots, K(\widehat{\boldsymbol{u}}_k, \widehat{\boldsymbol{u}}_{k-r+1})]$, the $\mathbb{A}_m$-kernel affine projection algorithm ($\mathbb{A}_m$-KAPA) is derived. This algorithm is an extension to $\mathbb{A}_m$ of the kernel affine projection (KAPA) proposed in [Ric+09]. In particular, the $\mathbb{A}_m$-KAPA reproduces the $\mathbb{A}_m$-NLSM similar to the real valued KAPA. The $\mathbb{A}_m$-KAPA is summarized in Algorithm 3.

Algorithm 3: Summary of the $\mathbb{A}_m$-KAPA ($\mathbb{A}_m$-KNLMS for $r = 1$)

Input : $\boldsymbol{u}_k \in \mathbb{A}_m^N$, $\boldsymbol{d}_k \in \mathbb{A}_m^r$, $\forall k \in \mathbb{N}$, $\eta > 0$
Initialize $k \leftarrow 0$, $\boldsymbol{\alpha}_k \leftarrow 0$ and $\mathcal{J}_k \leftarrow \{0\}$;
repeat
  Compute $K(\widehat{\boldsymbol{u}}_k, \widehat{\boldsymbol{u}}_j)\ \forall j \in \mathcal{J}_k$;
  if $\max_{j \in \mathcal{J}_k} |K(\widehat{\boldsymbol{u}}_k, \widehat{\boldsymbol{u}}_j)| \leq \eta$ **then**
    $\mathcal{J}_{k+1} \leftarrow \mathcal{J}_k \cup \{k\}$;
    $\boldsymbol{\alpha}_k \leftarrow \begin{bmatrix} \boldsymbol{\alpha}_k \\ 0 \end{bmatrix}$;
  else
    $\mathcal{J}_{k+1} \leftarrow \mathcal{J}_k$;
  end
until *convergence*;
$(\boldsymbol{H}_k)_{p,q} \leftarrow K(\widehat{\boldsymbol{u}}_{j_p^{(k)}}, \widehat{\boldsymbol{u}}_{k-q+1}) \begin{array}{c} \forall p \in \{1, 2, \ldots, s_{k+1}\}, \\ \forall q \in \{1, 2, \ldots, r\} \end{array}$;
$\widehat{\boldsymbol{e}}_k \leftarrow \widetilde{\boldsymbol{H}}_k^\top \widehat{\boldsymbol{\alpha}}_k - \widehat{\boldsymbol{d}}_k$;
Choose $\mu_k \in [0, 2]$;
$\widehat{\boldsymbol{\alpha}}_{k+1} \leftarrow \widehat{\boldsymbol{\alpha}}_k - \mu_k \widetilde{\boldsymbol{H}}_k (\widetilde{\boldsymbol{H}}_k^\top \widetilde{\boldsymbol{H}}_k + \varepsilon \boldsymbol{I}_{mr}) \widehat{\boldsymbol{e}}_k$;
$k \leftarrow k + 1$;

5.5 Numerical Examples

In this section, we examine the performance of the proposed nonlinear adaptive algorithm in both complex and quaternion domains.

Consider the following complex or quaternion valued nonlinear channel equalization problem. The input signal s_k is given as

$$s_k := 0.70\left(\sqrt{1-\beta}z_{1,k}\mathbf{i}_1 + \beta z_{2,k}\mathbf{i}_2\right) \in \mathbb{C},$$
$$s_k := 0.1z_{1,k}\mathbf{i}_1 + 0.5z_{2,k}\mathbf{i}_2 + 0.3z_{3,k}\mathbf{i}_3 + 0.3z_{4,k}\mathbf{i}_4 \in \mathbb{H},$$

where $z_{\ell,k}$ ($\ell = 1,\dots,4$) is i.i.d. from real valued Gaussian distribution with mean 0 and variance 1. For each domain, the signal observed at a receiver is given as follows:

$$\begin{aligned} y_k &:= x_k + (0.2\mathbf{i}_1 + 0.25\mathbf{i}_2)x_k^2 + (0.12\mathbf{i}_1 + 0.09\mathbf{i}_2)x_k^3 + n_k \in \mathbb{C}, \\ x_k &:= (-0.9\mathbf{i}_1 + 0.8\mathbf{i}_2)s_k + (0.6\mathbf{i}_1 - 0.7\mathbf{i}_2)s_{k-1} \in \mathbb{C}, \\ y_k &:= x_k + (0.2\mathbf{i}_1 + 0.25\mathbf{i}_2 + 0.2\mathbf{i}_3 + 0.15\mathbf{i}_4)x_k^2 \\ &\quad + (0.12\mathbf{i}_1 + 0.09\mathbf{i}_2 + 0.06\mathbf{i}_3 + 0.03\mathbf{i}_4)x_k^3 + n_k \in \mathbb{H}, \\ x_k &:= (-0.9\mathbf{i}_1 + 0.8\mathbf{i}_2 - 0.7\mathbf{i}_3 + 0.6\mathbf{i}_4)s_k \\ &\quad + (0.6\mathbf{i}_1 - 0.7\mathbf{i}_2 + 0.6\mathbf{i}_3 - 0.7\mathbf{i}_4)s_{k-1} \in \mathbb{H}. \end{aligned}$$

Namely, the channel is modeled by a nonlinear model consisting of a serial connection of a linear time-invariant system and a memoryless nonlinearity, followed by the additive C-D valued circular white Gaussian noise $n_k \in \mathbb{A}_m$ ($= \mathbb{C}$ or $\mathbb{H}$) with $\mathrm{SNR} := 10\log_{10}(\mathbb{E}\,|w_k|^2 \,/\, \mathbb{E}\,|n_k|^2) = 18$ dB, where $w_k := y_k - n_k \in \mathbb{C}$ or $\mathbb{H}$. The problem of channel equalization is to construct an inverse filter, that is, to recover the signal s_k from the received signal $(y_n)_{n<k}$. The diagram of this problem is drawn in Fig. 5.1. We let $\boldsymbol{u}_k := [y_{n+D}, y_{n+D-1}, \dots, y_{n+D-N+1}]^\top \in \mathbb{C}^N$ or $\mathbb{H}^N$ for $N = 5$ and $d_k := s_k$, where $D = 2$ represents the equalization time lag. This is a standard models that has been extensively used in the literature for such tasks [Liu+08].

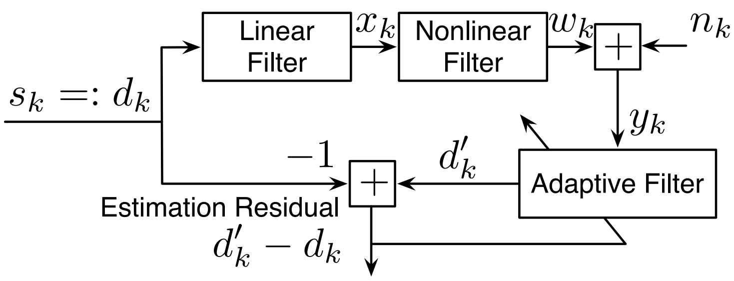


Figure 5.1.: Nonlinear channel equalization task

In the complex case, we compare the exiting nonlinear complex adaptive algorithms, the complex valued multilayer perceptron ($\mathbb{C}$-MLP) [AH09], and the complex nonlinear gradient descent ($\mathbb{C}$-NGD) [MG09], and the proposed $\mathbb{A}_m$-KAPA for the complex case, $\mathbb{C}$-KAPA with $r = 1$ (that is, $\mathbb{C}$-NLMS) as the simplest example. We employ the activation function fully complex hyperbolic tangent $\tanh(\cdot)$ for the $\mathbb{C}$-MLP and

$\mathbb{C}$-NGD, and employ the real valued Gaussian kernel $K(\boldsymbol{x}, \boldsymbol{y}) := \exp(-\|\widehat{\boldsymbol{x}} - \widehat{\boldsymbol{y}}\|_{\mathbb{R}}^2 / 5^2)$, $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{C}^N$ for $\mathbb{C}$-KAPA.

In the quaternion case, we compare the exiting nonlinear quaternion adaptive algorithms, the quaternion nonlinear gradient descent ($\mathbb{H}$-NGD) (QNGD in [Uja+11]), and the proposed $\mathbb{A}_m$-KAPA for the quaternion case, $\mathbb{H}$-KAPA with $r = 1$ ($\mathbb{H}$-NLMS) as the simplest example. The $\mathbb{H}$-NGD is implemented by using quaternion toolbox for MATLAB® [SB13]. We employ the activation function fully quaternion hyperbolic tangent $\tanh(\cdot)$ for the $\mathbb{C}$-NGD this is the same setting as [Uja+11]. We employ the real valued Gaussian kernel $K(\boldsymbol{x}, \boldsymbol{y}) := \exp(-\|\widehat{\boldsymbol{x}} - \widehat{\boldsymbol{y}}\|_{\mathbb{R}}^2 / \sigma^2)$, $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{H}^N$, where $\sigma^2 = 7.0$ for $\mathbb{H}$-KAPA.

For each domain, the set of parameters employed in this experiment is listed in Table 5.1. The step-sizes are chosen to be the best steady-state performance.

Table 5.1.: Parameter settings for experiments in Section 5.5

$\mathbb{C}$-MLP	λ(step-size) $= 0.01$ (# of hidden layers) $= 40$
$\mathbb{C}$-NGD	λ(step-size) $= 0.01$
$\mathbb{C}$-KAPA	$\mu_k = 0.5$, $r = 1.0$, $\eta = 0.7$
$\mathbb{H}$-NGD	λ(stepsize) $= 0.01$
$\mathbb{H}$-KAPA	$\mu_k = 1.0$, $r = 1.0$, $\eta = 0.01$

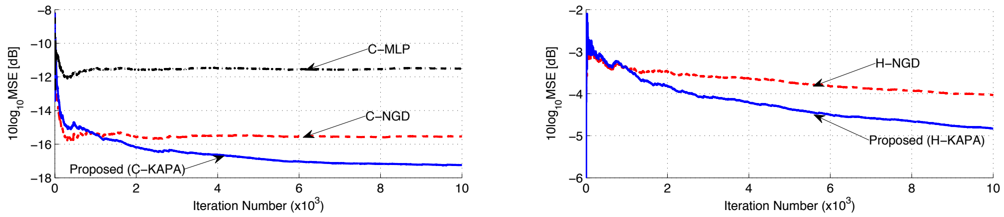


Figure 5.2.: Comparison of the performances of existing and proposed adaptive algorithms. From left to right, figures are the comparisons of MSE in the complex and quaternion domain.

Fig. 5.2 depicts the comparison of the proposed and existing nonlinear adaptive algorithms. It indicates that the proposed $\mathbb{C}$-KAPA achieves better performance than the existing $\mathbb{C}$-MLP and $\mathbb{C}$-NGD in the complex domain. Moreover, it shows that the proposed $\mathbb{H}$-APA achieves better performance than the existing $\mathbb{H}$-NGD.

5.6 Summary

In this chapter, we have shown that nonlinear online learning problem in C-D domain can be reduced to linear one in real domain by combining the kernel trick and the algebraic real translation proposed in Chapter 3. We then have proposed a new hypercomplex kernelized online learning algorithm named $\mathbb{A}_m$-kernel adaptive projected subgradient projection ($\mathbb{A}_m$-KAPSP) by applying the APSM to the reduced real linear online learning problem. Similar to the linear case, the proposed method includes wide hypercomplex nonlinear online learning algorithms such as the C-D counterparts of the kernel normalized least mean squares ($\mathbb{A}_m$-KNLMS), the kernel affine projection ($\mathbb{A}_m$-APA) as special cases. Numerical examples are performed in the context of nonlinear channel equalization problem in several C-D domains including complex and quaternion and shows the effectiveness of the proposed methods.

6 Hypercomplex Singular Value Decomposition

6.1 $\mathbb{R}$-eigenvalues and Their Properties

In this section, we introduce a new notion $\mathbb{R}$-*eigenvalue* which can be defined for general C-D matrices.

Definition 3 ($\mathbb{R}$-eigenvalues of C-D matrices)**.** For a C-D matrix $\boldsymbol{A} \in \mathbb{A}_m^{N\times N}$, we respectively call a complex-valued scalar $\lambda \in \mathbb{C}$ and a complex vector $\boldsymbol{x} \in \mathbb{C}^{mN} \backslash \{\boldsymbol{0}\}$ an $\mathbb{R}$-eigenvalue and $\mathbb{R}$-eigenvector of $\boldsymbol{A}$ provided that $\widetilde{\boldsymbol{A}}\boldsymbol{x} = \lambda\boldsymbol{x}$, where $\widetilde{\boldsymbol{A}} \in \mathbb{R}^{mN\times mN}$ is defined in (3.5).

The $\mathbb{R}$-eigenvalue has very strong relation with eigenvalues of complex (i.e., $m = 2$) square matrices.

Lemma 3 ($\mathbb{R}$-eigenvalues of complex matrices)**.** *Suppose that a complex square matrix $\boldsymbol{A} \in \mathbb{C}^{N\times N}$ has an eigenvalue $\lambda \in \mathbb{C}$, then $\boldsymbol{A}$ also has $\mathbb{R}$-eigenvalues λ and λ^*. If $\boldsymbol{A}$ is Hermitian, it has real-valued $\mathbb{R}$-eigenvalues λ with multiplicity 2.*

Proof. Let $\boldsymbol{x}$ $(:= \boldsymbol{x}_1 + \boldsymbol{x}_2\imath \in \mathbb{C}^N,\ \boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{R}^N)$ be the eigenvector of $\boldsymbol{A}$ $(:= \boldsymbol{A}_1 + \boldsymbol{A}_2\imath,\ \boldsymbol{A}_1, \boldsymbol{A}_2 \in \mathbb{R}^{N\times N})$ corresponding to λ, i.e., $\boldsymbol{x}$ satisfies $\boldsymbol{A}\boldsymbol{x} = \lambda\boldsymbol{x}$. Then, with $\begin{bmatrix}\boldsymbol{x}\\ -\boldsymbol{x}\imath\end{bmatrix} \in \mathbb{C}^{2N}$, we have

$$\widetilde{\boldsymbol{A}}\begin{bmatrix}\boldsymbol{x}\\ -\boldsymbol{x}\imath\end{bmatrix} = \begin{bmatrix}\boldsymbol{A}_1 & -\boldsymbol{A}_2\\ \boldsymbol{A}_2 & \boldsymbol{A}_1\end{bmatrix}\begin{bmatrix}\boldsymbol{x}\\ -\boldsymbol{x}\imath\end{bmatrix} = \begin{bmatrix}(\boldsymbol{A}_1 + \boldsymbol{A}_2\imath)\boldsymbol{x}\\ -(\boldsymbol{A}_1 + \boldsymbol{A}_2\imath)\boldsymbol{x}\imath\end{bmatrix}$$
$$= \begin{bmatrix}\boldsymbol{A}\boldsymbol{x}\\ -\boldsymbol{A}\boldsymbol{x}\imath\end{bmatrix} = \lambda\begin{bmatrix}\boldsymbol{x}\\ -\boldsymbol{x}\imath\end{bmatrix}.$$

Hence, λ is also an $\mathbb{R}$-eigenvalue of $\boldsymbol{A}$. Since $\widetilde{\boldsymbol{A}}$ is a real matrix, its non-real eigenvalues occurs in conjugate pairs, so λ^* is also an eigenvalue of $\widetilde{\boldsymbol{A}}$, that is, an $\mathbb{R}$-eigenvalue of $\boldsymbol{A}$. This can be easily verified by multiplying an eigenvector $\begin{bmatrix}\boldsymbol{x}^*\\ \boldsymbol{x}^*\imath\end{bmatrix} \in \mathbb{C}^{2N}$ to $\widetilde{\boldsymbol{A}}$ from the right.

If $\boldsymbol{A}$ is Hermitian, all eigenvalues of $\boldsymbol{A}$ is real-values, so we have $\widetilde{\boldsymbol{A}\boldsymbol{x}} = \lambda\widetilde{\boldsymbol{x}}$ by applying $\widetilde{(\cdot)}$ to both sides to $\boldsymbol{A}\boldsymbol{x} = \lambda\boldsymbol{x}$. From Theorem 1-1) and 1-8), we have $\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{x}} = \lambda\widetilde{\boldsymbol{x}}$, i.e.,

$$\widetilde{\boldsymbol{A}} \begin{bmatrix} \boldsymbol{x}_1 & -\boldsymbol{x}_2 \\ \boldsymbol{x}_2 & \boldsymbol{x}_1 \end{bmatrix} = \lambda \begin{bmatrix} \boldsymbol{x}_1 & -\boldsymbol{x}_2 \\ \boldsymbol{x}_2 & \boldsymbol{x}_1 \end{bmatrix}.$$

This implies that $\widetilde{\boldsymbol{A}}$ has eigenvalues λ with multiplicity 2. □

Here, we discuss hypercomplex cases ($m \geq 4$). As explained in Section 2.1.3, the difficulties for computing hypercomplex eigenvalues is mainly due to the lack of commutativity and associativity of hypercomplex multiplications. On the other hand, any C-D matrix can be translated into a real matrix without losing any information by the algebraic translation introduced in Section 3.2. Once the translated real matrix is obtained from a C-D matrix, we are completely freed from any computational difficulty in C-D number system. Moreover, since the translated real matrix is just a real matrix, any C-D square matrix $\boldsymbol{A} \in \mathbb{A}_m^{N\times N}$ always has mN number of $\mathbb{R}$-eigenvalues in $\mathbb{C}$.

The most well-studied hypercomplex eigenvalue problem has been the quaternion right eigenvalue problem. In general, the quaternion square matrix $\boldsymbol{A} \in \mathbb{H}^{N\times N}$ has an infinite number of right eigenvalues in $\mathbb{H}$ [Zha97]. It is also well-known that any $N \times N$ quaternion matrix $\boldsymbol{A}$ has exactly N right eigenvalues in $\mathbb{C}$ with non-negative imaginary parts [Lee49; Bre51]. These eigenvalues are called the *standard eigenvalues of* $\boldsymbol{A}$. The standard eigenvalues can be systematically computed by calculating eigenvalues of the *complex adjoint matrix*

$$\chi_{\boldsymbol{A}} := \begin{bmatrix} \boldsymbol{A}_r & \boldsymbol{A}_i \\ -\boldsymbol{A}_i^* & \boldsymbol{A}_r^* \end{bmatrix} \in \mathbb{C}^{2N\times 2N} \tag{6.1}$$

of $\boldsymbol{A} := \boldsymbol{A}_r + \boldsymbol{A}_i\jmath$ ($\boldsymbol{A}_r, \boldsymbol{A}_i \in \mathbb{C}^{N\times N}$) [Lee49; Bre51]. The standard eigenvalue is defined only up to quaternion since the complex adjoints can be defined only for quaternion matrices.

For quaternion matrices, we have the following relation between the $\mathbb{R}$-eigenvalues and the standard eigenvalues of quaternion matrices.

Theorem 3 ($\mathbb{R}$-eigenvalue of quaternion matrices)**.** *Suppose that a quaternion square matrix* $\boldsymbol{A} \in \mathbb{H}^{N\times N}$ *has a standard eigenvalue* $\lambda \in \mathbb{C}$. *Then* $\boldsymbol{A}$ *also has* $\mathbb{R}$*-eigenvalues* λ *and* λ^* *with multiplicity 2. If* $\boldsymbol{A}$ *is Hermitian, it has real-valued* $\mathbb{R}$*-eigenvalues* λ *with multiplicity 4.*

Proof. Since λ is a standard eigenvalue of $\boldsymbol{A}$, λ is an eigenvalue of the complex adjoint matrix $\chi_{\boldsymbol{A}}$. From Lemma 3 and Lemma A.5.1 (See Appendix A.5), $\widetilde{\chi}_{\boldsymbol{A}}$ and $\widetilde{\boldsymbol{A}}$ have eigenvalues λ and λ^*. On the other hand, $\chi_{\boldsymbol{A}}$ is similar[1] to its conjugate $\chi^*_{\boldsymbol{A}}$, so λ^* is also an eigenvalue of $\chi_{\boldsymbol{A}}$, so $\widetilde{\chi}_{\boldsymbol{A}}$ and $\widetilde{\boldsymbol{A}}$ also have eigenvalues λ^* and $(\lambda^*)^* = \lambda$. Hence $\boldsymbol{A}$ has $\mathbb{R}$-eigenvalues λ and λ^* with multiplicity 2.

If $\boldsymbol{A}$ is Hermitian, its complex adjoint is also Hermitian [Zha97], so all standard eigenvalues of $\boldsymbol{A}$ is real. Hence, we have $\widetilde{\boldsymbol{A}\boldsymbol{x}} = \widetilde{\boldsymbol{x}}\lambda$ by applying $\widetilde{(\cdot)}$ to both sides of $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{x}\lambda$. From Theorem 1-1) and 1-8), we have $\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{x}} = \lambda\widetilde{\boldsymbol{x}}$, i.e.,

$$\widetilde{\boldsymbol{A}}\begin{bmatrix} \boldsymbol{x}_1 & -\boldsymbol{x}_2 & -\boldsymbol{x}_3 & -\boldsymbol{x}_4 \\ \boldsymbol{x}_2 & \boldsymbol{x}_1 & -\boldsymbol{x}_4 & \boldsymbol{x}_3 \\ \boldsymbol{x}_3 & \boldsymbol{x}_4 & \boldsymbol{x}_1 & -\boldsymbol{x}_2 \\ \boldsymbol{x}_4 & -\boldsymbol{x}_3 & \boldsymbol{x}_2 & \boldsymbol{x}_1 \end{bmatrix} = \lambda \begin{bmatrix} \boldsymbol{x}_1 & -\boldsymbol{x}_2 & -\boldsymbol{x}_3 & -\boldsymbol{x}_4 \\ \boldsymbol{x}_2 & \boldsymbol{x}_1 & -\boldsymbol{x}_4 & \boldsymbol{x}_3 \\ \boldsymbol{x}_3 & \boldsymbol{x}_4 & \boldsymbol{x}_1 & -\boldsymbol{x}_2 \\ \boldsymbol{x}_4 & -\boldsymbol{x}_3 & \boldsymbol{x}_2 & \boldsymbol{x}_1 \end{bmatrix}$$

if we write $\boldsymbol{x} := \boldsymbol{x}_1 + \boldsymbol{x}_2\imath + \boldsymbol{x}_3\jmath + \boldsymbol{x}_4\kappa$ ($\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4 \in \mathbb{R}^N$) as before. This implies that $\widetilde{\boldsymbol{A}}$ has eigenvalues λ with multiplicity 4. □

From Lemma 3 and Theorem 3, eigenvalues of three equivalent matrices $\boldsymbol{A} \in \mathbb{A}^{N\times N}$, $\chi_{\boldsymbol{A}} \in \mathbb{C}^{2N\times 2N}$ and $\widetilde{\boldsymbol{A}} \in \mathbb{R}^{4N\times 4N}$ can be summarized in TABLE 6.1.

Table 6.1.: Eigenvalues of three equivalent matrices in $\mathbb{H}$, $\mathbb{C}$ and $\mathbb{R}$

<table>
<tr><th>Domain</th><th>Matrix</th><th colspan="4">Eigenvalues</th></tr>
<tr><td>$\mathbb{H}^{N\times N}$</td><td>$\boldsymbol{A}$</td><td colspan="4">λ</td></tr>
<tr><td>$\mathbb{C}^{2N\times 2N}$</td><td>$\chi_{\boldsymbol{A}}$</td><td colspan="2">λ</td><td colspan="2">λ^*</td></tr>
<tr><td>$\mathbb{R}^{4N\times 4N}$</td><td>$\widetilde{\boldsymbol{A}}(\Leftrightarrow \widetilde{\chi}_{\boldsymbol{A}})$</td><td>$\lambda$</td><td>$\lambda^*$</td><td>$\lambda^*$</td><td>$\lambda$</td></tr>
</table>

6.2 Singular Value Decompositions and Ranks

In this section, we introduce useful definitions of singular value decomposition (SVD), rank and best low-rank approximation of C-D matrices and clarify their properties.

Definition 4 (C-D singular value decomposition and rank evaluation)**.** For any C-D matrix $\boldsymbol{A} \in \mathbb{A}_m^{M\times N}$, there exist orthogonal real matrices $\boldsymbol{U} \in \mathbb{R}^{mM\times mM}$ and $\boldsymbol{V} \in \mathbb{R}^{mN\times mN}$ such that

$$\widetilde{\boldsymbol{A}} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top, \tag{6.2}$$

[1] Two quaternions (including complex numbers) λ and $q^{-1}\lambda q$ are said to be *similar* for non-zero quaternion q. If the real part of λ is zero and $|q| = 1$ then $q^{-1}\lambda q$ becomes the rotation of λ. Moreover, it is known that any quaternion is similar to a certain complex number (for detail, see Lemma 2.1 in [Zha97]).

where $\mathbf{\Sigma} := \mathrm{diag}(\sigma_1, \dots, \sigma_r, 0, \dots, 0) \in \mathbb{R}^{mM \times mN}$ is a rectangular diagonal matrix with positive singular values $\sigma_1 \geq \cdots \geq \sigma_r (> 0)$ of $\widetilde{\boldsymbol{A}}$ on the diagonal. We call it *C-D singular value decomposition (C-D SVD)* and evaluate the *rank* of $\boldsymbol{A}$ by $r = \mathrm{rank}(\widetilde{\boldsymbol{A}}) \leq \max(mM, mN)$.

Note that the mapping: $\widetilde{(\cdot)} : \mathbb{A}_m^{M \times N} \to \mathbb{R}^{mM \times nN}$ is defined in Section 3.2. It is clear that the C-D SVD and the rank evaluation is always available for any C-D matrix $\boldsymbol{A} \in \mathbb{A}_m^{M \times N}$ since $\widetilde{\boldsymbol{A}} \in \mathbb{R}^{mM \times mN}$ is no longer hypercomplex but just a real matrix. Similar to the eigenvalue problem, the SVDs have not been well-established for almost C-D number systems. However, in the quaternion case, both the SVD and the rank of quaternion matrix are well-established [Zha97]. Moreover, the rank evaluation of C-D matrices has very strong relation to well-established original ranks [Zha97] in C-D domain.

Lemma 4 (Relation between the rank evaluation and the original ranks in C-D domain)**.** *For complex ($m = 2$) or quaternion ($m = 4$) cases,*

$$\mathrm{rank}(\widetilde{\boldsymbol{A}}) = m \cdot \mathrm{rank}(\boldsymbol{A})$$

holds for all $\boldsymbol{A} \in \mathbb{A}_m^{M \times N}$.

Proof. Suppose that $\boldsymbol{A}$ is of rank r ($\leq \min(M, N)$) then $\boldsymbol{A}$ has r positive singular values[2]. In this case, the Hermitian matrix $\boldsymbol{A}^{\mathsf{H}}\boldsymbol{A}$ has r positive eigenvalues and from Lemma 3 ($m = 2$) or Theorem 3 ($m = 4$) it has r positive $\mathbb{R}$-eigenvalues with multiplicity m, i.e., it has mr positive $\mathbb{R}$-eigenvalues. From Theorem 1-8), we have $\widetilde{(\boldsymbol{A}^{\mathsf{H}}\boldsymbol{A})} = \widetilde{\boldsymbol{A}}^{\top}\widetilde{\boldsymbol{A}}$, so $\widetilde{\boldsymbol{A}}$ has mr positive singular values and thus $\mathrm{rank}(\widetilde{\boldsymbol{A}}) = mr$. □

Lemma 4 implies that the rank evaluation of C-D matrices is equivalent to the product of the dimension of C-D number and the original rank. Therefore, minimizing the evaluated rank is equivalent to minimizing the original rank at least in well-established complex and quaternion case. In this section, by passing through the Schmidt-Eckart-Young theorem [Ste93; BIG03], we propose a new best low-rank approximation technique of C-D matrices.

Lemma 5 (Low rank approximation of hypercomplex matrices)**.** *For a C-D matrix $\boldsymbol{A} \in \mathbb{A}_m^{M \times N}$ of* $\mathrm{rank}(\widetilde{\boldsymbol{A}}) = r$*, a best p rank approximation is achieved by*

$$\min_{\substack{\boldsymbol{X} \in \mathbb{R}^{mM \times mN} \\ \mathrm{rank}(\widetilde{\boldsymbol{X}}) \leq p}} \left\| \widetilde{\boldsymbol{A}} - \boldsymbol{X} \right\|_F = \left\| \widetilde{\boldsymbol{A}} - \boldsymbol{A}_p \right\|_F = \sqrt{\sum_{i=p+1}^{r} \sigma_i^2},$$

[2]It also holds for the quaternion case, for detail, see e.g., [Zha97]

where $\|\cdot\|_F$ *is the Frobenius norm,* $\widetilde{\boldsymbol{A}} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$, $\boldsymbol{A}_p = \boldsymbol{U}\boldsymbol{\Sigma}_p\boldsymbol{V}^\top$ *and* $\boldsymbol{\Sigma} = \mathrm{diag}(\sigma_1, \ldots, \sigma_p, 0, \ldots, 0) \in \mathbb{R}^{mM \times mN}$.

Proof. It can be straightforwardly verified by applying Schmidt-Echkart-Young theorem to $\widetilde{\boldsymbol{A}}$. □

Remark 8. Note that this lemma does not always provide $\boldsymbol{A}_p$ in $\mathfrak{S}_{\mathbb{A}_m}(M, N)$ (see Section 3.2) but just in $\mathbb{R}^{mM \times mN}$, that is, $\boldsymbol{A}_p$ does not always enjoy the special structure of non-trivial mapping $\widetilde{(\cdot)}$ and the existence of the equivalent C-D matrix to $\boldsymbol{A}_p$ is not guaranteed in general. However, the following theorem insists that $\boldsymbol{A}_p$ inherits this special structure from $\widetilde{\boldsymbol{A}}$ if we choose the reduced rank p to be a multiple of m.

Theorem 4 (Equivalence to the known low-rank approximations in $\mathbb{C}$ and $\mathbb{H}$)**.** *Consider complex* ($m = 2$) *or quaternion* ($m = 4$) *cases. If* $\boldsymbol{A}_{mp} \in \mathbb{R}^{mM \times mN}$ *achieves the best low* mp *rank approximation of* $\boldsymbol{A} \in \mathbb{A}_m^{M \times N}$ ($p < \mathrm{rank}(\boldsymbol{A})$)*, then* $\boldsymbol{A}_{mp} \in \mathfrak{S}_{\mathbb{A}_m}(M, N)$*. Moreover, the C-D reverted matrix* $\underset{\frown}{\boldsymbol{A}}_{mp} \in \mathbb{A}_m^{M \times N}$ *also achieves the best low* p *rank approximation of* $\boldsymbol{A}$ *in [BM04].*

Proof. See Appendix A.6. □

This theorem implies that Lemma 5 is the generalization of the known result in [BM04].

As seen in Definition 4, the rank of matrices, even in C-D domain, is well-defined as a discrete valued function. However, its numerical evaluation requires reliable counting of non-zero singular values, which is certainly not easy especially for large-scale matrices. Moreover, minimization of the rank function under constraints often turns out to be NP-hard. To suppress the rank of matrix under constraints, many powerful computational strategies become available if we use nuclear norm of real matrix $\widetilde{\boldsymbol{A}}$ in (6.2), i.e.,

$$\left\|\widetilde{\boldsymbol{A}}\right\|_* := \sum_{i=1}^{r} \sigma_i \tag{6.3}$$

as a best convex relaxation of rank function $\mathrm{rank}(\widetilde{\boldsymbol{A}})$(Note: It is well known that the nuclear norm is a greatest convex minorant of the rank function [Faz+01]) For convex optimization problems defined with nuclear norm function, the following shrinkage operator has been playing as a key player (see, e.g., Section 8.1):

$$\mathrm{shrink}(\widetilde{\boldsymbol{A}}, \tau) := \boldsymbol{U}\boldsymbol{\Sigma}_\tau\boldsymbol{V}^\top \tag{6.4}$$

for (6.2) and the shrunk diagonal matrix $\mathbf{\Sigma}_\tau := \mathrm{diag}(\max\{\sigma_1 - \tau, 0\}, \ldots, \max\{\sigma_r - \tau, 0\}, 0, \ldots, 0)$. For the shrinkage operator, we have the following theorem:

Theorem 5 (Inheritance of special structure of non-trivial mapping with the shrinkage operator)**.** *Consider the case where $m \leq 4$. For any $\boldsymbol{A} \in \mathbb{A}_m^{M\times N}$ and $\tau > 0$, the shrinkage operator keeps the special structure of non-trivial mapping $\widetilde{(\cdot)}$, i.e.,* $\mathrm{shrink}(\widetilde{\boldsymbol{A}}, \tau) \in \mathfrak{S}_{\mathbb{A}_m}(M, N)$.

Proof. See Appendix A.6. □

Remark 9. Neither Theorem 4 nor Theorem 5 holds in general for the octonion or higher dimensional C-D domains. This can be easily verified with simple counter examples. Let $\widetilde{\boldsymbol{A}} = \boldsymbol{U}\mathbf{\Sigma}\boldsymbol{V}^\top \in \mathfrak{S}_{\mathbb{A}_8}(2,2) \subsetneq \mathbb{R}^{16\times 16}$ be the C-D singular value decomposition of $\boldsymbol{A} := \begin{bmatrix} 1+\mathbf{i}_5 & \mathbf{i}_2+\mathbf{i}_6 \\ \mathbf{i}_3+\mathbf{i}_7 & \mathbf{i}_4+\mathbf{i}_8 \end{bmatrix} \in \mathbb{A}_8^{2\times 2} = \mathbb{O}^{2\times 2}$. Then it is easy to verify that its best low 8 rank approximation $\boldsymbol{A}_8 \in \mathbb{R}^{16\times 16} \setminus \mathfrak{S}_{\mathbb{A}_8}(2,2)$ and $\mathrm{shrink}(\widetilde{\boldsymbol{A}}, 2.0) \in \mathbb{R}^{16\times 16} \setminus \mathfrak{S}_{\mathbb{A}_8}(2,2)$.

Remark 5 implies that we need additional idea to utilize best low rank approximation in Lemma 5 and shrinkage operator in (6.4) while keeping the consistency with $\mathfrak{S}_{\mathbb{A}_m}(M, N)$. The concrete idea will be discussed in Section 8.2 (Note: In this paper, the projection onto the special subspace M_2 in Section 8.2 is introduced for such a purpose).

6.3 Summary

In this chapter, we have introduced a new notion R-eigenvalue, which can be defined and calculated for any C-D square matrices, and show that the R-eigenvalue is a natural extension of the original eigenvalues keeping consistency with known results. Moreover, we have also newly defined the C-D singular value decomposition (C-D SVD) and rank evaluation method for general C-D matrices based on the R-eigenvalue, and have clarified their properties and the relations to known results.

7 Hypercomplex Matrix Recovery via Convex Optimization

7.1 Convex Relaxation of Hypercomplex Robust Principal Component Analysis

In this section, we formulate the robust principal component analysis (RPCA) in C-D domain. Since the rank evaluation of C-D matrices defined in 2.1.4 is available for general C-D domain, we can formulate it in C-D domain as follows:

$$\underset{\boldsymbol{L},\boldsymbol{S}\in\mathbb{A}_m^{M\times N}}{\text{minimize}} \quad \text{rank}(\widetilde{\boldsymbol{L}}) + \lambda \|\boldsymbol{S}\|_{0,\mathbb{A}_m} \quad \text{s.t.} \quad \boldsymbol{M} = \boldsymbol{L} + \boldsymbol{S}, \tag{7.1}$$

where $\lambda > 0$ and $\|\boldsymbol{A}\|_{0,\mathbb{A}_m}$ is the number of non-zero entries in $\boldsymbol{A} \in \mathbb{A}^{M\times N}$. Obviously from Lemma 4, this is a C-D generalization of RPCA in real domain. Similar to the real case [Can+11], (7.1) is NP-hard. For relaxing (7.1) to a convex optimization problem, we first introduce newly the ℓ_1-norm of a C-D matrix as follows:

$$\|\boldsymbol{A}\|_{1,\mathbb{A}_m} := \sum_{i,j=1}^{M,N} |\boldsymbol{A}_{i,j}|, \quad \boldsymbol{A} \in \mathbb{A}_m^{M\times N}. \tag{7.2}$$

For any C-D matrix $\boldsymbol{A} \in \mathbb{A}_m^{M\times N}$, we can consider the following real matrix:

$$\widehat{\widehat{\boldsymbol{A}}} = \begin{bmatrix} \widehat{\boldsymbol{A}}_{1,1} & \cdots & \widehat{\boldsymbol{A}}_{1,N} \\ \vdots & \ddots & \vdots \\ \widehat{\boldsymbol{A}}_{M,1} & \cdots & \widehat{\boldsymbol{A}}_{M,N} \end{bmatrix} \in \mathbb{R}^{mM\times N} \tag{7.3}$$

with the mapping $\widehat{\widehat{(\cdot)}} : \mathbb{A}_m^{M\times N} \to \mathbb{R}^{mM\times N}$. Note that $\widehat{\widehat{(\cdot)}}$ is just a permutation of $\widehat{(\cdot)}$ in (3.3) and we can define its inverse $\check{\check{(\cdot)}} : \mathbb{R}^{mM\times N} \to \mathbb{A}_m^{M\times N} : \widehat{\widehat{\boldsymbol{A}}} \mapsto \boldsymbol{A}$. Then, we have

$$\|\boldsymbol{A}\|_{1,\mathbb{A}_m} = \sum_{i,j=1}^{M,N} \left\|\widehat{\boldsymbol{A}}_{i,j}\right\|_2 =: \left\|\widehat{\widehat{\boldsymbol{A}}}\right\|_1^{\mathbb{A}_m},$$

where $\|\cdot\|_2$ is the ℓ_2-norm of real vectors and note that $\widehat{\boldsymbol{A}}_{i,j} \in \mathbb{R}^m$ ($i = 1,\ldots,M$, $j = 1,\ldots,N$). This implies that the ℓ_1-norm of a C-D matrix $\boldsymbol{A}$ can be regarded as a convex function $\|\cdot\|_1^{\mathbb{A}_m}$ of the real matrix $\widehat{\widehat{\boldsymbol{A}}} \in \mathbb{R}^{mM\times N}$ and it evaluates the *group sparsity (structured sparsity)* [WR07] of $\boldsymbol{A}'$. Therefore, we have the following theorem:

Theorem 6 (Proximity operator of $\|\cdot\|_1^{\mathbb{A}_m}$)**.** *For any $\boldsymbol{A} \in \mathbb{A}_m^{M\times N}$, the proximity operator (see Appendix 2.2.2) of $\|\cdot\|_1^{\mathbb{A}_m}$ with index $\tau > 0$ can be easily calculated group-wise as*

$$\left[\operatorname{prox}_{\gamma\|\cdot\|_1^{\mathbb{A}_m}}(\widehat{\boldsymbol{A}})\right]_{i,j} = \frac{\widehat{\boldsymbol{A}}_{i,j}}{\left\|\widehat{\boldsymbol{A}}_{i,j}\right\|_2} \max(0, \left\|\widehat{\boldsymbol{A}}_{i,j}\right\|_2 - \tau), \tag{7.4}$$

$$=: [\widehat{\mathrm{ST}}(\widehat{\boldsymbol{A}}, \tau)]_{i,j} \tag{7.5}$$

where the indices $[\cdot]_{i,j}$ ($i = 1, \dots, M$, $j = 1, \dots, N$) stand for the (i,j)-th group of size $m \times 1$ in $\operatorname{prox}_{\gamma\|\cdot\|_1^{\mathbb{A}_m}}(\boldsymbol{A}') \in \mathbb{R}^{mM\times N}$.

If we note that $\left\|\widehat{\boldsymbol{A}}_{i,j}\right\|_2 = |\boldsymbol{A}_{i,j}|$ and by applying $\breve{(\cdot)}$ defined in (3.4) to the right hand side of (7.4), we formally obtain the following entry-wise soft-thresholding function of C-D matrices:

$$[\mathrm{ST}(\boldsymbol{A}, \tau)]_{i,j} := \frac{\boldsymbol{A}_{i,j}}{|\boldsymbol{A}_{i,j}|} \max(0, |\boldsymbol{A}_{i,j}| - \tau).$$

This is obviously equivalent to (7.5) and a C-D generalization of real, complex and quaternion soft-thresholding functions.

By using ℓ_1 norm we have discussed above and approximating $\mathbb{R}$-rank with nuclear norm, we have the following convex optimization problem:

$$\underset{\boldsymbol{L},\boldsymbol{S}\in\mathbb{A}_m^{M\times N}}{\text{minimize}} \ \|\widetilde{\boldsymbol{L}}\|_* + \lambda \|\boldsymbol{S}\|_{1,\mathbb{A}_m} \quad \text{s.t.} \quad \boldsymbol{M} = \boldsymbol{L} + \boldsymbol{S}, \tag{7.6}$$

where $\|\cdot\|_*$ is the nuclear norm of real matrices i.e., the sum of positive singular values. In this thesis, we call the problem (7.6) *Cayley-Dickson principal component pursuit (C-D PCP)*. Obviously, if we set $\mathbb{A}_m = \mathbb{R}$ ($m = 1$) and $\mathbb{A}_m = \mathbb{C}$ or $\mathbb{H}$ ($\mathbb{A}_m = 2$ or 4), (7.6) is respectively identical to the original PCP in real domain [Can+11], and the complex and quaternionic PCP in [CY16]. Therefore, the C-D PCP is a natural generalization of these problems. Moreover, the C-D PCP can be regarded as a convex optimization problem in real domain since the ℓ_1-norm of C-D matrices $\boldsymbol{A}$ can be regarded as a convex function of real matrices, and can be solved by proximal splitting techniques.

7.2 Hypercomplex Principal Component Pursuit Algorithm via Convex Optimization

In this section, we derive a new algorithm based on the Douglas-Rachford splitting technique [CP07] to solve the C-D PCP (7.6) efficiently. Denote the 2-fold Cartesian product of the spaces of real matrices by $\mathcal{H}_0 := \mathbb{R}^{mM\times mN} \times \mathbb{R}^{mM\times N}$. By defining the inner product $\langle \mathcal{X}, \mathcal{Y}\rangle_{\mathcal{H}_0} := \frac{1}{2}\operatorname{tr}(\boldsymbol{X}_1^\top \boldsymbol{Y}_1) + \frac{1}{2}\operatorname{tr}(\boldsymbol{X}_2^\top \boldsymbol{Y}_2) =: \frac{1}{2}\langle \boldsymbol{X}_1, \boldsymbol{Y}_1\rangle_{\mathbb{R}^{mM\times mN}} + \frac{1}{2}\langle \boldsymbol{X}_2, \boldsymbol{Y}_2\rangle_{\mathbb{R}^{mM\times N}}$, where $\mathcal{X} := [\boldsymbol{X}_1, \boldsymbol{X}_2] \in \mathcal{H}_0$ and $\mathcal{Y} := [\boldsymbol{Y}_1, \boldsymbol{Y}_2] \in \mathcal{H}_0$, ($\boldsymbol{X}_1, \boldsymbol{Y}_1 \in \mathbb{R}^{mM\times mN}$, $\boldsymbol{X}_2, \boldsymbol{Y}_2 \in \mathbb{R}^{mM\times N}$) and induced norm $\|\mathcal{X}\|_{\mathcal{H}_0} := \sqrt{\langle \mathcal{X}, \mathcal{X}\rangle_{\mathcal{H}_0}}$, $\mathcal{H}_0$ becomes a real Hilbert space. First, we reformulate the problem (7.6) as an unconstrained the sum of two functions as follows:

$$\underset{\mathcal{Z}\in\mathcal{H}_0}{\text{minimize}} \quad f(\mathcal{Z}) + g(\mathcal{Z}), \tag{7.7}$$

where

$$\begin{cases} f(\mathcal{Z}) := f_1(\boldsymbol{Z}_1) + f_2(\boldsymbol{Z}_2) = \|\boldsymbol{Z}_1\|_* + \|\boldsymbol{Z}_2\|_1^{\mathbb{A}_m}, \\ g(\mathcal{Z}) := \iota_{D_1}(\mathcal{Z}) = \begin{cases} 0 & (\text{if } \mathcal{Z} \in D_1), \\ +\infty & (\text{otherwise}), \end{cases} \end{cases}$$

$$\begin{aligned} \mathcal{Z} :=& [\boldsymbol{Z}_1, \boldsymbol{Z}_2] \in \mathcal{H}_0, \\ D_1 :=& \left\{ [\boldsymbol{Z}_1, \boldsymbol{Z}_1] \in D_2 \,\middle|\, \boldsymbol{M} = \underset{\smile}{\boldsymbol{Z}}_1 + \check{\check{\boldsymbol{Z}}}_2 \right\} \subset D_2, \\ D_2 :=& S \times \mathbb{R}^{mM\times N} \subset \mathcal{H}_0, \\ S :=& S_{\mathbb{A}_m}(M, N) \subset \mathbb{R}^{mN\times mM}. \end{aligned}$$

Note that the subspace D_1 represents the constraint that the observation $\boldsymbol{M}$ is from the sum of low rank and sparse matrices. This requests that both $\boldsymbol{Z}_1$ belong to S, so we need the subspace D_2.

Apparently, this reformulation (7.7) is equivalent to (7.6), so all we need is to identify the concrete calculation of the proximity operators of f and g. In the same way as [GY10], the proximity operator of f is given by

$$\operatorname{prox}_{\gamma f}(\mathcal{X}) = \left[\operatorname{prox}_{2\gamma f}(\boldsymbol{X}_1), \operatorname{prox}_{2\gamma f}(\boldsymbol{X}_2)\right].$$

The proximity operator of f_1, i.e., the nuclear norm with index 2γ is given by

$$\operatorname{prox}_{2\gamma f_1}(\boldsymbol{X}_1) = \operatorname{shrink}(\boldsymbol{X}_1, 2\gamma).$$

By Theorem 6, the proximity operator of f_2, reduces to the group-wise soft-thresholding (7.5) of a real matrix:

$$\mathrm{prox}_{2\gamma f_2}(\boldsymbol{X}_2) = \widehat{\mathrm{ST}}(\boldsymbol{X}_2, 2\gamma\lambda). \tag{7.8}$$

For the function g, the proximity operator of the indicator function ι_{D_1} is the orthogonal projection P_{D_1} onto the subspace D_1, i.e.,

$$\mathrm{prox}_{\gamma g}(\mathcal{X}) = P_{D_1}(\mathcal{X}) := \operatorname*{arg\,min}_{\mathcal{Y}\in D_1} \|\mathcal{X} - \mathcal{Y}\|_{\mathcal{H}_0}.$$

Since $D_1 \subset D_2 \subset \mathcal{H}_0$, we have by [Deu01, 5.14, Reduction principle]

$$P_{D_1}(\mathcal{X}) = P_{D_1}|D_2 \circ P_{D_2}(\mathcal{X}).$$

Note that '$|D_2$' in $P_{D_1}|D_2$ stands for the restriction of the domain to the subspace D_2. The orthogonal projection $P_{D_2} : \mathcal{H}_0 \to D_2$ and $P_{D_1}|D_2 : D_2 \to D_1$ respectively can be calculated as

$$P_{D_2}(\mathcal{X}) = [P_S(\boldsymbol{X}_1), \boldsymbol{X}_2]$$

and

$$P_{D_1}|D_2(\mathcal{X}) = \frac{1}{2}\left[\widetilde{\boldsymbol{M}} + \boldsymbol{X}_1 - \widetilde{\boldsymbol{X}}_2^{\star}, \widehat{\boldsymbol{M}} - \widehat{\boldsymbol{X}}_1^{\star} + \boldsymbol{X}_2\right],$$

where $\boldsymbol{X}_1^{\star} := \underset{\frown}{\boldsymbol{X}}_1 \in \mathbb{A}_m^{M\times N}$ and $\boldsymbol{X}_2^{\star} := \check{\check{\boldsymbol{X}}}_2 \in \mathbb{A}_m^{M\times N}$. For $P_S(\boldsymbol{X}_1)$, let $\boldsymbol{E}_{p,q,\ell} := \boldsymbol{E}_{p,q}\mathbf{i}_\ell \in \mathbb{A}_m^{M\times N}$ $(\ell = 1,\ldots,m)$, where $\boldsymbol{E}_{p,q} \in \mathbb{R}^{M\times N}$ is the matrix only whose (p,q)-th entry $(p = 1,\ldots,M,\ q = 1,\ldots,N)$ is 1 and all other entries are 0. Then, we can easily verify that

$$\langle \widetilde{\boldsymbol{E}}_{p,q,\ell}, \widetilde{\boldsymbol{E}}_{p',q',\ell'} \rangle_{\mathbb{R}^{mM\times mN}} = \begin{cases} m & (\text{if } (p,q,\ell) = (p',q',\ell')), \\ 0 & (\text{otherwise}). \end{cases}$$

and therefore, $\{\frac{1}{\sqrt{m}}\widetilde{\boldsymbol{E}}_{p,q,\ell}\}_{p=1,q=1,\ell=1}^{M,N,m}$ is an orthonormal basis of S and thus $P_S(\boldsymbol{X}_1)$ can be easily calculated as:

$$P_S(\boldsymbol{X}_1) = \frac{1}{m}\sum_{p=1}^{M}\sum_{q=1}^{N}\sum_{\ell=1}^{m} \langle \boldsymbol{X}_1, \widetilde{\boldsymbol{E}}_{p,q,\ell} \rangle_{\mathbb{R}^{mM\times mN}} \widetilde{\boldsymbol{E}}_{p,q,\ell}.$$

Now, we can calculate

$$\begin{aligned} \mathrm{prox}_{\gamma g}(\mathcal{X}) &= P_{D_1}|D_2 \circ P_{D_2}(\mathcal{X}) \\ &= P_{D_1}|D_2\,[P_S(\boldsymbol{X}_1), \boldsymbol{X}_2] \\ &= \frac{1}{2}\left[\widetilde{\boldsymbol{M}} + P_S(\boldsymbol{X}_1) - \widetilde{\boldsymbol{X}}_2^{\star}, \widehat{\boldsymbol{M}} - \widehat{\boldsymbol{X}}_1^{\star\star} + \boldsymbol{X}_2\right], \end{aligned}$$

where $\boldsymbol{X}_1^{\star\star} := \utilde{P_S(\boldsymbol{X}_1)} \in \mathbb{A}_m^{M\times N}$. Since all ingredients are identified, we can summarize the proposed hypercomplex principal component pursuit algorithm in Algorithm 4. Here, $(t_k)_{k\geq 0} \subset [0,2]$ satisfied $\sum_{k\geq 0} t_k(2-t_k) = +\infty$, $\gamma \in (0,+\infty)$.

Algorithm 4: $\mathbb{A}_m$-Douglas-Rachford splitting for hypercomplex principal component pursuit ($\mathbb{A}_m$-DRS-PCP)

Input : $\boldsymbol{M}$, t_k, λ
Output: Low $\mathbb{R}$-rank $\boldsymbol{L}$ and sparse $\boldsymbol{S}$
Initialize $k \leftarrow 0$, $\boldsymbol{L}^{(k)} \leftarrow \boldsymbol{0}$, $\boldsymbol{S}^{(k)} \leftarrow \boldsymbol{0}$;
repeat

$\quad \boldsymbol{L}^{\star\star} \leftarrow \utilde{P_S(\boldsymbol{L}^{(k)})}$, $\boldsymbol{S}^{\star\star} \leftarrow \check{\check{\boldsymbol{S}}}^{(k)}$;

$\quad \boldsymbol{L}^{\star} \leftarrow (\widetilde{\boldsymbol{M}} + P_S(\boldsymbol{L}^{(k)}) - \widetilde{\boldsymbol{S}}^{\star\star})/2$;

$\quad \boldsymbol{S}^{\star} \leftarrow (\widehat{\boldsymbol{M}} - \widehat{\boldsymbol{L}}^{\star\star} + \boldsymbol{S}^{(k)})/2$;

$\quad \boldsymbol{L}^{(k+1)} \leftarrow \boldsymbol{L}^{(k)} + t_k \left(\mathrm{shrink}(2\boldsymbol{L}^{\star} - \boldsymbol{L}^{(k)}, 2\gamma) - \boldsymbol{L}^{\star}\right)$;

$\quad \boldsymbol{S}^{(k+1)} \leftarrow \boldsymbol{S}^{(k)} + t_k \left(\widehat{\mathrm{ST}}(2\boldsymbol{S}^{\star} - \boldsymbol{S}^{(k)}, 2\gamma\lambda) - \boldsymbol{S}^{\star}\right)$;

$\quad k \leftarrow k+1$;

until *convergence*;

$\boldsymbol{L}^{\star\star} \leftarrow \utilde{P_S(\boldsymbol{L}^{(k)})}$, $\boldsymbol{S}^{\star\star} \leftarrow \check{\check{\boldsymbol{S}}}^{(k)}$;

$\boldsymbol{L}^{\star} \leftarrow (\widetilde{\boldsymbol{M}} + P_S(\boldsymbol{L}^{(k)}) - \widetilde{\boldsymbol{S}}^{\star\star})/2$;

$\boldsymbol{S}^{\star} \leftarrow (\widehat{\boldsymbol{M}} - \widehat{\boldsymbol{L}}^{\star\star} + \boldsymbol{S}^{(k)})/2$;

$[\boldsymbol{L}, \boldsymbol{S}] \leftarrow [\utilde{\boldsymbol{L}}^{\star}, \check{\check{\boldsymbol{S}}}^{\star}]$;

Note that the shrinkage operator does not keep the special structure of $\widetilde{(\cdot)}$, i.e., $\mathrm{shrink}(\widetilde{\boldsymbol{A}}, 2\gamma) \notin S$ in general, so we need the projection onto the structure P_S. However, in complex and quaternion domain, it keeps the structure as shown in Fact 5, so $\boldsymbol{L}^{(k)} \in S$ and thus $P_S(\boldsymbol{L}^{(k)}) = \boldsymbol{L}^{(k)}$ for all $k \geq 0$. Especially if $m = 1$ (i.e., $\mathbb{A}_m = \mathbb{R}$), Algorithm 4 is identical to the original DRS for the PCP (DR-PCP) proposed in [GY10]. Lastly, we state the convergence of the proposed algorithm.

Theorem 7 (Convergence of $\mathbb{A}_m$-DRS-PCP)**.** *Let parameters of Algorithm 4 be chosen so that* $\gamma \in (0,+\infty)$*,* $(t_k)_{k\geq 0} \subset [0,2]$ *satisfying* $\sum_{k\geq 0} t_k(2-t_k) = +\infty$*. Then, the output of Algorithm 1 converges to a minimizer of (7.6).*

Remark 10. In this thesis, we employ the DRS for solving (7.6) but it can be also solved by other advanced convex optimization techniques such as the *alternating direction method of multipliers (ADMM)* [Boy+11] and the *primal-dual splitting (PDS)* [Con13; Vũ13].

7.3 Numerical Examples

In this section, we perform some numerical experiments for examining the effectiveness of the proposed method. Following the settings in [Lin+09; GY10], we randomly generate an input pairs $(\boldsymbol{L}, \boldsymbol{S})$ as follows: $\boldsymbol{L} := \boldsymbol{X}_L \boldsymbol{X}_R^{\mathsf{H}} \in \mathbb{A}_m^{M \times N}$, where $\boldsymbol{X}_L \in \mathbb{A}_m^{M \times r}$ and $\boldsymbol{X}_R \in \mathbb{A}_m^{N \times r}$ $(r < \min(M, N))$ with the all real and imaginary parts of each entry of $\boldsymbol{X}_L, \boldsymbol{X}_R$ being i.i.d from $\mathcal{N}(0, 1)$. Note that r is not always agree to $m\mathrm{rank}^{\mathbb{R}}$ since Lemma 4 does not hold for $m > 4$ in general. We choose the support set of $\boldsymbol{S}$ uniformly at random from all support set of size ρMN $(\rho \in (0, 1))$. All real and imaginary parts of the non-zero entries are independently drawn form $\mathcal{U}(-256, 256)$. We fixed $\lambda = 1/\sqrt{\max(M, N)}$ in the experiments. We perform experiments in the case where $\mathbb{A}_m = \mathbb{O}$ $(m = 8)$.We compare the proposed method $\mathbb{A}_m$-DRS-PCP and three part-wise DRS-PCP method, $\mathbb{H}^2$-DRS-PCP, $\mathbb{C}^4$-DRS-PCP and $\mathbb{R}^8$-DRS-PCP. These part-wise methods split $\mathbb{O}$ into $\mathbb{H}^2$, $\mathbb{C}^4$ and $\mathbb{R}^8$ estimate all parts separately.

Table 7.1.: Performance comparison

$\boldsymbol{L}, \boldsymbol{S} \in \mathbb{O}^{32 \times 32}, \rho = 0.2, \mathrm{rank}^{\mathbb{R}}(\boldsymbol{L}) = 29$			$\boldsymbol{L}, \boldsymbol{S} \in \mathbb{O}^{32 \times 32}, \rho = 0.2, \mathrm{rank}^{\mathbb{R}}(\boldsymbol{L}) = 58$		
Algorithm	error	# iter.	Algorithm	error	# iter.
$\mathbb{A}_m$-DRS-PCP	2.0e-6	2,044	$\mathbb{A}_m$-DRS-PCP	6.5e-2	2,971
$\mathbb{H}^2$-DRS-PCP	1.0	3,009	$\mathbb{H}^2$-DRS-PCP	11.9	2,553
$\mathbb{C}^4$-DRS-PCP	8.2e-1	2,265	$\mathbb{C}^4$-DRS-PCP	9.6	1,894
$\mathbb{R}^8$-DRS-PCP	37.5	2,990	$\mathbb{R}^8$-DRS-PCP	78.7	1,924

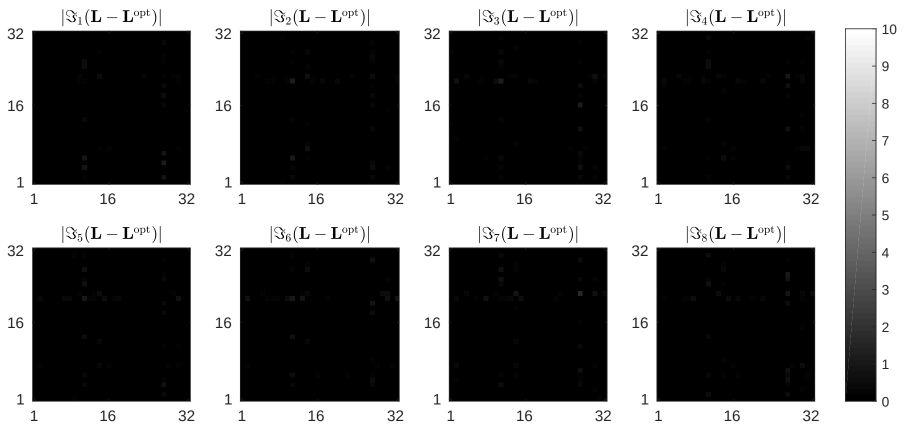


Figure 7.1.: Difference between the original matrix $\boldsymbol{L}^{\mathrm{opt}}$ and the estimated low rank matrix $\boldsymbol{L}$ with $\mathbb{A}_m$-DRS-PCP

Table 7.1 shows the performance comparisons of all four algorithms. Figure 7.1 and Figure 7.2 show the differences of all real and parts between the estimated low rank matrices and original matrices for the right case of Table 7.1 for $\mathbb{A}_m$-DRS-PCP (Figure 7.1) and $\mathbb{H}^2$-DRS-PCP (Figure 7.2). They show that the proposed method $\mathbb{A}_m$-DRS-PCP outperforms all part-wise methods by exploiting all correlations among real and imaginary parts. $\mathbb{H}$-DRS-PCP and $\mathbb{C}$-DRS-PCP much better than $\mathbb{R}$-DRS-PCP since it may utilize these correlations in part.

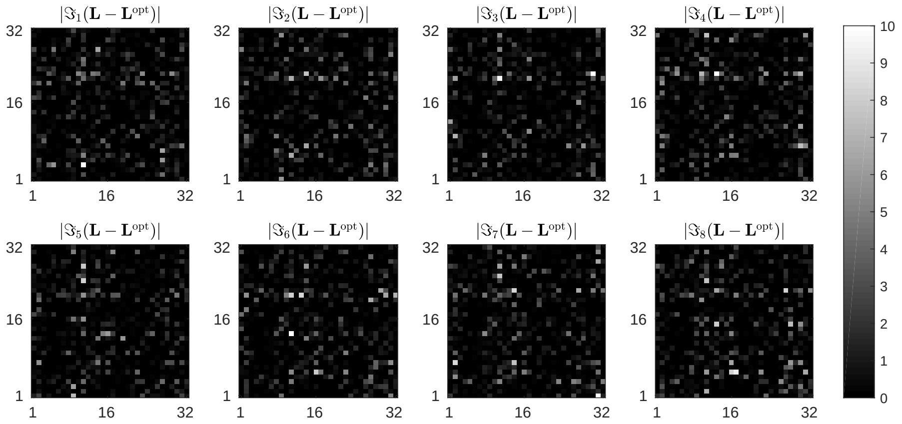


Figure 7.2.: Difference between the original matrix $\boldsymbol{L}^{\text{opt}}$ and the estimated low rank matrix $\boldsymbol{L}$ with $\mathbb{H}^2$-DRS-PCP

7.4 Non-negative Matrix Completion in Hypercomplex Domain

In this section, we first formulate non-negative hypercomplex matrix completion problem and then propose an algorithmic solution to it.

7.4.1 Formulation

First of all we have to define the non-negativeness of hypercomplex number. For simplicity, in this paper, we consider the following non-negativeness:

$$\mathbb{A}_{m+} = \{a \in \mathbb{A}_m \,|\Im_\ell(a) \geq 0, \forall \ell = 1, \ldots, m\} \subset \mathbb{A}_m. \tag{7.9}$$

We call an element in $\mathbb{A}_{m+}$ a *part-wise non-negative* C-D number.

By using this definition, the non-negative low rank hypercomplex matrix completion can be formulated as the following optimization problem:

$$\underset{\boldsymbol{X} \in \mathbb{A}_{m+}^{M \times N}}{\text{minimize}} \quad \text{rank}(\widetilde{\boldsymbol{X}}) \quad \text{s.t.} \quad \boldsymbol{X}_\Omega = \boldsymbol{Y}_\Omega, \tag{7.10}$$

where $\boldsymbol{X}_\Omega$ denotes the restriction of the matrix on the entries given by Ω and $\boldsymbol{Y}_\Omega$ contains the values of those entries of $\boldsymbol{X}$. With the sampling operator $\mathcal{L}_\Omega$: $\mathbb{A}_m^{M \times N} \to \mathbb{A}_m^p$ extracting p observed entries into a vector $\boldsymbol{b} \in \mathbb{A}_m^p$ and the convex relaxation with the nuclear norm and Lagrange multiplier, we obtain the following unconstrained formulation:

$$\underset{\boldsymbol{X}\in\mathbb{A}_{m+}^{M\times N}}{\text{minimize}} \quad \left\|\widetilde{\boldsymbol{X}}\right\|_* + \frac{\lambda}{2}\left\|\mathcal{L}_\Omega(\boldsymbol{X}) - \boldsymbol{b}\right\|_{\mathbb{A}_m^p}^2, \tag{7.11}$$

where $\|\cdot\|_*$ is the nuclear norm of real matrices i.e., the sum of positive singular values. In this paper, we call the problem (7.11) *Cayley-Dickson non-negative matrix completion (C-D NNMC)*.

7.4.2 Algorithm based on Douglas-Rachford Splitting

In this section, we derive a new algorithm based on the Douglas-Rachford splitting technique [CP07] to solve the C-D NNMC (7.11) efficiently. The DRS is briefly summarized in Appendix. Denote the 2-fold Cartesian product of the spaces of real matrices by $\mathcal{H}_0 := \mathbb{R}^{mM\times mN} \times \mathbb{R}^{mM\times mN}$. Define the inner product $\langle \mathcal{X}, \mathcal{Y}\rangle_{\mathcal{H}_0} := \frac{1}{2}\operatorname{tr}(\boldsymbol{X}_1^\top \boldsymbol{Y}_1) + \frac{1}{2}\operatorname{tr}(\boldsymbol{X}_2^\top \boldsymbol{Y}_2)$, where $\mathcal{X} = [\boldsymbol{X}_1, \boldsymbol{X}_2] \in \mathcal{H}_0$ and $\mathcal{Y} = [\boldsymbol{Y}_1, \boldsymbol{Y}_2] \in \mathcal{H}_0$ ($\boldsymbol{X}_1, \boldsymbol{X}_2, \boldsymbol{Y}_1, \boldsymbol{Y}_2 \in \mathbb{R}^{mM\times mN}$) and induced norm $\|\mathcal{X}\|_{\mathcal{H}_0} := \sqrt{\langle \mathcal{X}, \mathcal{X}\rangle_{\mathcal{H}_0}}$, then $\mathcal{H}_0$ becomes a real Hilbert space.

We recast the problem (7.11) into an unconstrained minimization of the sum of two functions f and g:

$$\underset{\mathcal{Z}\in\mathcal{H}_0}{\text{minimize}} \quad f(\mathcal{Z}) + g(\mathcal{Z}), \tag{7.12}$$

where

$$\begin{cases} f(\mathcal{Z}) := f_1(\boldsymbol{Z}_1) + f_2(\boldsymbol{Z}_2) = \|\boldsymbol{Z}_1\|_* + \left\|\widehat{\mathcal{L}}_\Omega(\boldsymbol{Z}_2) - \widehat{\boldsymbol{b}}\right\|_2^2, \\ g(\mathcal{Z}) := \iota_D(\mathcal{Z}) = \begin{cases} 0 & (\text{if } \mathcal{Z} \in D), \\ +\infty & (\text{otherwise}), \end{cases} \end{cases}$$

$\mathcal{Z} := [\boldsymbol{Z}_1, \boldsymbol{Z}_2] \in \mathcal{H}_0$, $D := \{[\boldsymbol{Z}, \boldsymbol{Z}] \in D_1 \,|\, \boldsymbol{Z}_{i,j} \geq 0, \forall (i,j) \in \mathcal{I}\} \subset D_1$, $D_1 := \{[\boldsymbol{Z}_1, \boldsymbol{Z}_2] \in D_2 \,|\, \boldsymbol{Z}_1 = \boldsymbol{Z}_2\} \subset D_2$, $D_2 = \mathfrak{S}_{\mathbb{A}_m}(M,N) \times \mathfrak{S}_{\mathbb{A}_m}(M,N) \subset \mathcal{H}_0$, $\mathcal{I} := \{1,\ldots,M\} \times \{1,\ldots,N\}$ and $\widehat{\mathcal{L}}_\Omega$ satisfies $\widehat{\mathcal{L}}_\Omega(\widetilde{\boldsymbol{X}}) = \widehat{\mathcal{L}_\Omega(\boldsymbol{X})}$ for all $\boldsymbol{X} \in \mathbb{A}_m^{M\times N}$. Apparently this formulation (7.12) is equivalent to (7.11), so we only have to provide the concrete calculation of the proximity operators of f and g. The proximity operator of f is given by

$$\operatorname{prox}_{\gamma f}(\mathcal{X}) = [\operatorname{prox}_{2\gamma f_1}(\boldsymbol{X}_1), \operatorname{prox}_{2\gamma f_2}(\boldsymbol{X}_2)].$$

The proximity operator of f_1 with index $\tau := 2\gamma$ is given by

$$\operatorname{prox}_{\tau f_1}(\boldsymbol{X}_1) = \operatorname{shrink}(\boldsymbol{X}_1, \tau)$$

and

$$\left[\mathrm{prox}_{\tau f_2}(\boldsymbol{X}_2)\right]_{i,j} = \begin{cases} \left[\frac{\tau}{\lambda\tau+1}\{\lambda\widehat{\mathcal{L}}_\Omega^*(\widehat{\boldsymbol{b}})\} + \frac{1}{\tau}\boldsymbol{X}_2\right]_{i,j} & (\text{if } (i,j)\in\Omega), \\ [\boldsymbol{X}_2]_{i,j} & (\text{otherwise}), \end{cases}$$

where $\widehat{\mathcal{L}}_\Omega^* : \mathbb{R}^{mp} \to \mathbb{R}^{mM\times mN}$ is the adjoint operator of $\widehat{\mathcal{L}}_\Omega$ satisfying $\langle \widehat{\mathcal{L}}_\Omega(\boldsymbol{X}), \boldsymbol{v}\rangle_{\mathbb{R}^{mp}} = \langle \boldsymbol{X}, \widehat{\mathcal{L}}_\Omega^*(\boldsymbol{v})\rangle_{\mathbb{R}^{mM\times mN}}$ for all $\boldsymbol{X} \in \mathbb{R}^{mM\times mN}$ and $\boldsymbol{v} \in \mathbb{R}^{mp}$. Note that the shrinkage operator $\mathrm{shrink}(\cdot)$ is the soft-thresholding w.r.t. singular value vector.

For g, the proximity operator of the indicator function ι_{M_1} is the orthogonal projection P_D onto the subspace D, i.e.,

$$\mathrm{prox}_{\gamma g}(\mathcal{X}) = P_D(\mathcal{X}) := \arg\min_{\mathcal{Y}\in D} \|\mathcal{X} - \mathcal{Y}\|_{\mathcal{H}_0}.$$

Since $D \subset D_1 \subset D_2 \subset \mathcal{H}_0$, we have by [Deu01, 5.14, Reduction principle]

$$P_D(\mathcal{X}) = P_D|D_1 \circ P_{D_1}(\mathcal{X}) = P_D|D_1 \circ P_{D_1}|D_2 \circ P_{D_2}(\mathcal{X}).$$

Note that '$|D_i$' ($i = 1, 2$) stands for the restriction of the domain to the subspace D_i. The orthogonal projections $P_D|D_1 : D_1 \to D$, $P_{D_1}|D_2 : D_2 \to D_1$ and $P_{D_2} : \mathcal{H}_0 \to D_2$ respectively can be calculated as

$$\begin{aligned} P_{D_2}(\mathcal{X}) &= [P_{\mathfrak{S}}(\boldsymbol{X}_1), P_{\mathfrak{S}}(\boldsymbol{X}_2)], \\ P_{D_1}|D_2(\mathcal{X}) &= \frac{1}{2}[\boldsymbol{X}_1 + \boldsymbol{X}_2, \boldsymbol{X}_1 + \boldsymbol{X}_2], \\ P_D|D_1(\mathcal{X}) &= [\max^{\mathbb{A}_m}(\underset{\sim}{\boldsymbol{X}}_1, 0), \max^{\mathbb{A}_m}(\underset{\sim}{\boldsymbol{X}}_1, 0)], \end{aligned}$$

where $\mathfrak{S} := \mathfrak{S}_{\mathbb{A}_m}(M, N)$ and

$$[\max^{\mathbb{A}_m}(\boldsymbol{A}, 0)]_{i,j} := \sum_{\ell=1}^{m} \max(\Im_\ell(\boldsymbol{A}_{i,j}), 0)\mathbf{i}_\ell.$$

For $P_{\mathfrak{S}}(\boldsymbol{X}_i)$ ($i = 1, 2$), let $\boldsymbol{E}_{p,q,\ell} := \boldsymbol{E}_{p,q}\mathbf{i}_\ell \in \mathbb{A}_m^{M\times N}$ ($\ell = 1, \ldots, m$), where $\boldsymbol{E}_{p,q} \in \mathbb{R}^{M\times N}$ is the matrix only whose (p, q)-th entry ($p = 1, \ldots, M$, $q = 1, \ldots, N$) is 1 and all other entries are 0. Then, we can easily verify that

$$\langle \widetilde{\boldsymbol{E}}_{p,q,\ell}, \widetilde{\boldsymbol{E}}_{p',q',\ell'}\rangle_{\mathbb{R}^{mM\times mN}} = \begin{cases} m & (\text{if } (p,q,\ell) = (p',q',\ell')), \\ 0 & (\text{otherwise}). \end{cases}$$

Algorithm 5: $\mathbb{A}_m$-Douglas-Rachford splitting for hypercomplex non-negative matrix completion ($\mathbb{A}_m$-DRS-NNMC)

Input : $\boldsymbol{M}$, t_k, λ

Output: Recovered matrix $\boldsymbol{X} \in \mathbb{A}_{m+}^{M\times N}$

$k \leftarrow 0$, $\boldsymbol{X}_i^{(0)} \leftarrow \boldsymbol{0}$ $(\forall i = 1, 2)$;

while *not converged* **do**

$\quad \boldsymbol{X}^\star \leftarrow \frac{1}{2}\{P_{\mathfrak{S}}(\boldsymbol{X}_1^{(k)}) + P_{\mathfrak{S}}(\boldsymbol{X}_2^{(k)})\}$;

$\quad \boldsymbol{X}_+^\star \leftarrow \max^{\mathbb{A}_m}(\underset{\smile}{\boldsymbol{X}}^\star, 0)$;

$\quad \mathrm{prox}_{2\gamma f_1}(2\widetilde{\boldsymbol{X}}_+^\star - \boldsymbol{X}_1^{(k)}) \leftarrow \mathrm{shrink}(2\widetilde{\boldsymbol{X}}_+^\star - \boldsymbol{X}_1^{(k)}, 2\gamma)$;

$$\left[\mathrm{prox}_{2\gamma f_0}(2\widetilde{\boldsymbol{X}}_+^\star - \boldsymbol{X}_2^{(k)})\right]_{i,j} \leftarrow \begin{cases} \left[\frac{2\gamma}{2\lambda\gamma+1}\{\lambda\widehat{\mathcal{L}}_\Omega^*(\widehat{\boldsymbol{b}}) + \frac{1}{2\gamma}(2\widetilde{\boldsymbol{X}}_+^\star - \boldsymbol{X}_2^{(k)})\}\right]_{i,j} & (\text{if } (i,j) \in \Omega) \\ [2\widetilde{\boldsymbol{X}}_+^\star - \boldsymbol{X}_2^{(k)}]_{i,j} & (\text{otherwise}) \end{cases};$$

$\quad$ **for** $i = 1, 2$ **do**

$$\boldsymbol{X}_i^{(k+1)} \leftarrow \boldsymbol{X}_i^{(k)} + t_k\left[\mathrm{prox}_{2\gamma f_i}(2\widetilde{\boldsymbol{X}}_+^\star - \boldsymbol{X}_i^{(k)}) - \widetilde{\boldsymbol{X}}_+^\star\right];$$

$\quad$ **end**

$\quad k \leftarrow k + 1$

end

$\boldsymbol{X}^\star \leftarrow \frac{1}{2}\{P_{\mathfrak{S}}(\boldsymbol{X}_1^{(k)}) + P_{\mathfrak{S}}(\boldsymbol{X}_2^{(k)})\}$;

$\boldsymbol{X} \leftarrow \max^{\mathbb{A}_m}(\underset{\smile}{\boldsymbol{X}}^\star, 0)$;

and therefore, $\{\frac{1}{\sqrt{m}}\widetilde{\boldsymbol{E}}_{p,q,\ell}\}_{p=1,q=1,\ell=1}^{M,N,m}$ is an orthonormal basis of $\mathfrak{S}$ and thus $P_{\mathfrak{S}}(\boldsymbol{X}_i)$ can be easily calculated as:

$$P_{\mathfrak{S}}(\boldsymbol{X}_i) = \frac{1}{m}\sum_{p=1}^{M}\sum_{q=1}^{N}\sum_{\ell=1}^{m}\langle \boldsymbol{X}_i, \widetilde{\boldsymbol{E}}_{p,q,\ell}\rangle_{\mathbb{R}^{mM\times mN}}\widetilde{\boldsymbol{E}}_{p,q,\ell}.$$

Now, we can calculate

$$\begin{aligned} \mathrm{prox}_{\gamma g}(\mathcal{X}) &= P_D|D_1 \circ P_{D_1}|D_2 \circ P_{D_2}(\mathcal{X}) \\ &= P_D|D_1 \circ P_{D_1}|D_2[P_{\mathfrak{S}}(\boldsymbol{X}_1), P_{\mathfrak{S}}(\boldsymbol{X}_2)] \\ &= P_D|D_1[\boldsymbol{X}^\star, \boldsymbol{X}^\star] \\ &= [\max^{\mathbb{A}_m}(\underset{\smile}{\boldsymbol{X}}^\star, 0), \max^{\mathbb{A}_m}(\underset{\smile}{\boldsymbol{X}}^\star, 0)], \end{aligned}$$

where $\boldsymbol{X}^\star := \frac{1}{2}\{P_{\mathfrak{S}}(\boldsymbol{X}_1) + P_{\mathfrak{S}}(\boldsymbol{X}_2)\}$. Since all ingredients are identified, we can summarize the proposed matrix completion algorithm in Algorithm 5. Here, $(t_k)_{k\geq 0} \subset [0, 2]$ satisfied $\sum_{k\geq 0} t_k(2 - t_k) = +\infty$, $\gamma \in (0, +\infty)$. Note that the shrinkage operator $\mathrm{shrink}(\cdot)$ does not keep the special structure of $\widetilde{(\cdot)}$, i.e., $\mathrm{shrink}(\widetilde{\boldsymbol{A}}, 2\gamma) \notin \mathfrak{S}$ in general, so we need the projection onto the structure $P_{\mathfrak{S}}$. However, in complex and quaternion domain, it keeps the structure [MY18a]. Especially if $m = 1$ (i.e., $\mathbb{A}_m = \mathbb{R}$), Algorithm 5 is identical to the dual of the non-negative matrix completion in real domain proposed in [SM18]. Lastly, we state the convergence of the proposed algorithm.

Theorem 8 (Convergence of $\mathbb{A}_m$-DRS-NNMC)**.** *Let parameters of Algorithm 5 be chosen so that* $\gamma \in (0, +\infty)$*,* $(t_k)_{k\geq 0} \subset [0, 2]$ *satisfying* $\sum_{k\geq 0} t_k(2 - t_k) = +\infty$*. Then, the output of Algorithm 5 converges to a minimizer of (7.11).*

Remark 11. In this paper, we employ the DRS for solving (7.11) but it can be also solved by other advanced convex optimization techniques such as the *alternating direction method of multipliers (ADMM)* [Boy+11] and the *primal-dual splitting (PDS)* [Con13; Vũ13].

7.5 Numerical Examples

In this section, we perform some numerical experiments for examining the effectiveness of the proposed method. Following the settings in [Lin+09; GY10], we randomly generate part-wise non-negative C-D matrices as follows: $\boldsymbol{X} := \boldsymbol{X}_L \boldsymbol{X}_R^{\mathsf{H}} \in \mathbb{A}_m^{M\times N}$, where $\boldsymbol{X}_L \in \mathbb{A}_m^{M\times r}$ and $\boldsymbol{X}_R \in \mathbb{A}_m^{N\times r}$ $(r < \min(M, N))$ with all real and imaginary parts of each entry of $\boldsymbol{X}_L, \boldsymbol{X}_R$ being i.i.d. from $\mathcal{U}(0, 1)$. Only with this procedure, $\boldsymbol{X}$ is not always part-wise non-negative, so we add an absolute value of minimum negative value for each imaginary part. Note that r is not always agree to $m\mathrm{rank}(\widetilde{\boldsymbol{X}})$ (for detail, see [MY18b]). In these experiments, we fix $r = 2$ and $\mathrm{rank}(\widetilde{\boldsymbol{X}})$ becomes 66 (of full rank $8 \times 32 = 256$). For investigating the limitation of recovery, we try various percentage ρ of the entries to be known and randomly chose the support of the known entries. The value and the locations of the known entries of $\mathcal{X}_0$ are used as inputs for the algorithms. For the parameters, we set $\lambda = 2$ and $t_k = 1$. We perform experiments in the case where $\mathbb{A}_m = \mathbb{O}$ $(m = 8)$. Since hypercomplex non-negative matrix completion is itself completely a new, so we compare the proposed method $\mathbb{A}_m$-DRS-NNMC and three quaternion part-wise methods, $\mathbb{H}^2$-DRS-NNMC, $\mathbb{C}^4$-DRS-NNMC and $\mathbb{R}^4$-DRS-NNMC. These part-wise methods split $\mathbb{O}$ into $\mathbb{H}^2$, $\mathbb{C}^4$ etc. and estimate separately. Table 7.2 shows the performance com-

Table 7.2.: Performance comparison

$\boldsymbol{X} \in \mathbb{O}^{32\times 32}, \rho = 0.4, \mathrm{rank}(\widetilde{\boldsymbol{X}}) = 66$			$\boldsymbol{X} \in \mathbb{O}^{32\times 32}, \rho = 0.1, \mathrm{rank}(\widetilde{\boldsymbol{X}}) = 66$		
Algorithm	error	# iter.	Algorithm	error	# iter.
$\mathbb{A}_m$-DRS-NNMC	3.0e-2	1,628	$\mathbb{A}_m$-DRS-NNMC	1.3	44,867
$\mathbb{H}^2$-DRS-NNMC	5.0e-1	1,530	$\mathbb{H}^2$-DRS-NNMC	1.0e+1	43,327
$\mathbb{C}^4$-DRS-NNMC	1.1e+1	1,272	$\mathbb{C}^4$-DRS-NNMC	2.6e+1	64,137
$\mathbb{R}^8$-DRS-NNMC	7.8e-1	1,155	$\mathbb{R}^8$-DRS-NNMC	1.1e+1	42,340

parisons of all four algorithms. It shows that the proposed method $\mathbb{A}_m$-DRS-NNMC outperforms part-wise methods by exploiting all correlations among real and imaginary parts for both case. In the case where $\rho = 0.4$, $\mathbb{R}^4$-DRS-NNMC outperforms

$\mathbb{C}^4$-DRS-NNMC. It indicates that the performance can be worse if we consider wrong correlations. If ρ is less than 0.1, even $\mathbb{A}_m$-DRS-NNMC cannot recover the original matrix accurately so the recovery limitation is around here but it still better than the part-wise method.

7.6 Summary

We focused on two applicable hypercomplex matrix recovery problems, hypercomplex robust principal component analysis and hypercomplex matrix completion with non-negative constraints.

For hypercomplex robust principal component analysis, we first have introduced a sparsity measure of hypercomplex matrices. We have shown that evaluating this sparsity measure of hypercomplex matrix is equivalent to evaluate the group sparsity of translated real matrix. By jointly utilized this measure and rank evaluation introduced in Chapter 6 and through a convex relaxation, we have shown that the hypercomplex robust principal component analysis can be relaxed and reduced to structured convex optimization problem in real domain. We then have proposed an algorithmic solution to this reduced problem based on a proximal splitting technique.

For hypercomplex matrix completion with non-negative constraints, we have proposed part-wise non-negativeness of hypercomplex matrices. Thanks to the simple definition of general hypercomplex non-negativeness, the hypercomplex non-negative matrix completion problem can be recasted to equivalent structured convex optimization problem in real domain. We then have proposed an algorithmic solution to hypercomplex non-negative low rank matrix completion algorithm based on a proximal splitting method.

Numerical experiments the proposed algorithms successfully utilized algebraically natural correlations of each attribute to recover much more faithfully the original information.

8 Hypercomplex Tensor Completion with Convex Optimization

8.1 Hypercomplex Extensions of Tensor Basics

In this section, we extend the fundamental tensor algebra to hypercomplex domain. Original tensor basics in real domain are summarized in Appendix A.7.1. We basically adopt the nomenclature of [KB09]. Basic tensor notations in hypercomplex domain are summarized in TABLE 8.1.

A tensor is a generalization of a matrix to higher dimension. In this paper we denote it by a calligraphic letter e.g., $\mathcal{X} \in \mathbb{A}_m^{N_1 \times \cdots \times N_n} =: \mathscr{T}^{\mathbb{A}_m}$. The *order* (also called *ways* or *modes*) n of tensor is the number of dimensions. *Fibers* are the higher-order analogue of matrix rows and columns. A fiber is defined by fixing every index but one. The mode-k ($k = 1, \ldots, n$) fibers are all vectors $\boldsymbol{x}_{i_1 \ldots i_{k-1} : i_{k+1} \ldots i_n}$ which are obtained by fixing the value of $\{i_1, \ldots, i_n\} \setminus i_k$. The *mode-$k$ unfolding* (also called *matricization* or *flattening*) of a tensor $\mathcal{X} \in \mathscr{T}^{\mathbb{A}_m}$ denoted by the corresponding bold upper case $\boldsymbol{X}_{(k)} \in \mathbb{A}_m^{N_k \times I_k}$ is a $N_k \times I_k$ ($I_k = \prod_{\ell=1, \ell \neq k}^{n} N_\ell$) matrix and obtained by concatenating all mode-k fibers along columns. In this paper, just for convenience in notation, we define the mode-0 unfolding $\boldsymbol{X}_{(0)} \in \mathbb{A}_m^{N_0 \times I_0}$ as $\boldsymbol{X}_{(0)} = \boldsymbol{X}_{(1)}$ (i.e., $N_0 = N_1$ and $I_0 = I_1$). Note that the mode-0 unfolding can be arbitrary chosen from mode-k ($k = 1, \ldots, n$) unfoldings but must be fixed. There are several notions of tensor rank but the n-*rank* is easy to compute. Originally, the n-rank is defined as the tuple of the rank of the mode-k unfoldings. However, the rank is not well-defined for general C-D domain, so we newly define the n-*rank* of a n-dimensional hypercomplex tensor $\mathcal{X} \in \mathscr{T}^{\mathbb{A}_m}$ as the tuple of the rank evaluations of the mode-k unfoldings, i.e.,

$$n\text{-rank}(\mathcal{X}) := \left[\text{rank}(\widetilde{\boldsymbol{X}}_{(1)}), \ldots, \text{rank}(\widetilde{\boldsymbol{X}}_{(n)})\right].$$

In this paper, we will only focus on it as a rank of hypercomplex tensor.

Based on the n-rank introduced above, we formulate a hypercomplex low-rank tensor completion problem. Given a linear map $\mathcal{L}_\Omega : \mathscr{T}^{\mathbb{A}_m} \to \mathbb{A}_m^p$ with $p \leq \prod_{i=1}^{n} N_i$ and given $\boldsymbol{b} \in \mathbb{A}_m^p$, where $\mathcal{L}_\Omega$ is a sampling operator, which extracts the entries of a

Table 8.1.: Hypercomplex matrix and tensor notations

Symbol	Definition
$\mathscr{T}_{\mathbb{A}_m}$	Set of all order n C-D tensors
$\mathcal{X}$	A tensor in $\mathscr{T}_{\mathbb{A}_m}$
$\boldsymbol{X}$	A matrix in $\mathbb{A}_m^{M\times N}$
$\boldsymbol{x}$	A vector in $\mathbb{A}_m^N$
x	A scalor in $\mathbb{A}$
$\boldsymbol{X}_{(k)}$	Mode-k unfolding of $\mathcal{X}$ ($k \in \{0, 1, \dots, n\}$)
$[\cdot]_{i,j}$	Element of a matrix at the position (i, j)
$\times_k$	k-mode product of a tensor and a matrix (see Fig. 8.1)
$\lVert\cdot\rVert_F$	Frobenius norm of matrices
$\lVert\cdot\rVert_*$	Nuclear norm of real matrices
$\lVert\cdot\rVert_2$	ℓ_2-norm of real vectors
$\mathcal{L}_\Omega$	Sampling operator with revealed entries Ω
$\mathcal{L}'_{\Omega(0)}$	Sampling operator equivalent to $\mathcal{L}_\Omega$ for mode-0 unfolded and then non-trivial mapped matrices
$\mathcal{L}'^*_{\Omega(0)}$	Adjoint operator of $\mathcal{L}'_{\Omega(0)}$

tensor in $\mathscr{T}^{\mathbb{A}_m}$ into a p-dimensional hypercomplex vector in $\mathbb{A}_m^p$ at positions given by the set of revealed entries denoted by Ω.

The goal of the low-rank tensor completion problem is to find the hypercomplex tensor $\mathcal{X} \in \mathscr{T}^{\mathbb{A}_m}$ that minimizes all entries in n-rank fulfilling the linear measurements $\mathcal{L}_\Omega(\mathcal{X}) = \boldsymbol{b}$. This can be expressed as the following optimization problem:

$$\underset{\mathcal{X}\in\mathscr{T}^{\mathbb{A}_m}}{\text{minimize}} \quad \sum_{i=1}^{n} \text{rank}(\widetilde{\boldsymbol{X}}_{(i)}) \quad \text{s.t.} \quad \mathcal{L}_\Omega(\mathcal{X}) = \boldsymbol{b}.$$

Following the convex relaxation in [Gan+11], we obtain the unconstrained formulation as follows:

$$\underset{\mathcal{X}\in\mathscr{T}^{\mathbb{A}_m}}{\text{minimize}} \quad \sum_{i=1}^{n} \left\lVert \widetilde{\boldsymbol{X}}_{(i)} \right\rVert_* + \frac{\lambda}{2} \lVert \mathcal{L}_\Omega(\mathcal{X}) - \boldsymbol{b} \rVert^2_{\mathbb{A}_m^p}, \tag{8.1}$$

where $\lVert\cdot\rVert_*$ is the nuclear norm of real matrices (see (6.3) in Section 6.2), i.e., the sum of all singular values.

8.2 Hypercomplex Tensor Completion Algorithm via Convex Optimization

In this section, we will derive an algorithm based on the *Douglas-Rachford splitting (DRS)* [EB92; CP07] to solve the problem (8.1) efficiently. The main ideas of DRS are summarized in Appendix 2.2.2. Denote the $(n + 1)$-fold Cartesian product of the spaces of real matrices by $\mathcal{H}_0 := \mathbb{R}^{mN_0 \times mI_0} \times \mathbb{R}^{mN_1 \times mI_1} \times \cdots \times \mathbb{R}^{mN_n \times mI_n}$. By defining the inner product $\langle \mathcal{X}, \mathcal{Y} \rangle_{\mathcal{H}_0} := \frac{1}{n+1} \sum_{k=0}^{n} \mathrm{tr}(\boldsymbol{X}_k^\top \boldsymbol{Y}_k) =: \frac{1}{n+1} \sum_{k=0}^{n} \langle \boldsymbol{X}_k, \boldsymbol{Y}_k \rangle_{\mathbb{R}^{mN_k \times mI_k}}$, where $\mathcal{X} := [\boldsymbol{X}_0, \boldsymbol{X}_1 \ldots, \boldsymbol{X}_n] \in \mathcal{H}_0$ ($\boldsymbol{X}_k \in \mathbb{R}^{mN_k \times mI_k}$, $k = 0, \ldots, n$) and $\mathcal{Y} := [\boldsymbol{Y}_0, \boldsymbol{Y}_1 \ldots, \boldsymbol{Y}_n] \in \mathcal{H}_0$ ($\boldsymbol{Y}_k \in \mathbb{R}^{mN_k \times mI_k}$, $k = 0, \ldots, n$), and induced norm $\|\mathcal{X}\|_{\mathcal{H}_0} := \sqrt{\langle \mathcal{X}, \mathcal{X} \rangle_{\mathcal{H}_0}}$, $\mathcal{H}_0$ becomes a real Hilbert space. First, we reformulate the problem (8.1) as the unconstrained minimization of the sum of two functions as follows:

$$\underset{\mathcal{Z} \in \mathcal{H}_0}{\text{minimize}} \quad f(\mathcal{Z}) + g(\mathcal{Z}), \tag{8.2}$$

where

$$\begin{cases} f(\mathcal{Z}) := \displaystyle\sum_{k=0}^{n} f_k(\boldsymbol{Z}_k) = \frac{\lambda}{2} \left\| \mathcal{L}'_{\Omega^{(0)}}(\boldsymbol{Z}_0) - \widehat{\boldsymbol{b}} \right\|_2^2 + \sum_{k=1}^{n} \|\boldsymbol{Z}_k\|_*, \\ g(\mathcal{Z}) := \iota_{M_1}(\mathcal{Z}) = \begin{cases} 0 & (\text{if } \mathcal{Z} \in M_1), \\ +\infty & (\text{otherwise}), \end{cases} \end{cases}$$

$$\begin{aligned} \mathcal{Z} :=& [\boldsymbol{Z}_0, \ldots, \boldsymbol{Z}_n] \in \mathcal{H}_0, \\ M_1 :=& \{[\boldsymbol{Z}_0, \ldots, \boldsymbol{Z}_n] \in M_2 | \ \mathrm{refold}(\underset{\sim}{\boldsymbol{Z}}_0) = \cdots = \mathrm{refold}(\underset{\sim}{\boldsymbol{Z}}_n)\} \subset M_2, \\ M_2 :=& \mathfrak{S}_0 \times \cdots \times \mathfrak{S}_n \subset \mathcal{H}_0, \\ \mathfrak{S}_k :=& \mathfrak{S}_{\mathbb{A}_m}(N_k, I_k) \subset \mathbb{R}^{mN_k \times mI_k} (k = 1, \ldots, n), \end{aligned}$$

$\mathcal{L}'_{\Omega^{(0)}} : \mathbb{R}^{mN_0 \times mI_0} \to \mathbb{R}^{mp}$ satisfies $\mathcal{L}'_{\Omega^{(0)}}(\widetilde{\boldsymbol{X}}_{(0)}) = \widehat{\mathcal{L}_\Omega(\mathcal{X})}$ for all $\mathcal{X} \in \mathscr{T}^{\mathbb{A}_m}$, $\Omega^{(0)}$ is the set of known entries, of mode-0 unfolding of the input tensor, corresponding to Ω, $\widehat{\boldsymbol{b}} \in \mathbb{R}^{mp}$ is the trivial mapped (see (3.3)) vector of $\boldsymbol{b}$, and $\mathrm{refold}(\cdot)$ denotes the refolding of $\mathbb{A}_m$ matrix into an $\mathbb{A}_m$ tensor. Simplified notation $\mathfrak{S}_k$ has been introduced for extensive use below.

Note that the subspace M_1 represents the constraint that all $(n+1)$ matrices $\boldsymbol{Z}_k$ $(k = 0, \ldots, n)$ are unfolded and then applied $\widetilde{(\cdot)}$ from a common single hypercomplex tensor. However, to represent it, $\boldsymbol{Z}_k$ must belong to $\mathfrak{S}_k$, so we also need the subspace M_2.

Apparently, this reformulation (8.2) is equivalent to (8.1), so all we need is to identify the concrete calculations of the proximity operators of f and g (see (2.11) in Appendix 2.2.2). In the same way as [Gan+11], the proximity operator of f is given by

$$\mathrm{prox}_{\gamma f}(\mathcal{X}) = \left[\mathrm{prox}_{(n+1)\gamma f_0}(\boldsymbol{X}_0), \ldots, \mathrm{prox}_{(n+1)\gamma f_n}(\boldsymbol{X}_n) \right].$$

The proximity operators of f_k $(k = 0, \ldots, n)$ with index τ are given by

$$\operatorname{prox}_{\tau f_k}(\boldsymbol{X}_k) = \operatorname{shrink}(\boldsymbol{X}_k, \tau)$$

for $k = 1, \ldots, n$ and

$$\left[\operatorname{prox}_{\tau f_0}(\boldsymbol{X}_0)\right]_{i,j} = \begin{cases} \left[\frac{\tau}{\lambda\tau+1}\{\lambda \mathcal{L}'^{*}_{\Omega^{(0)}}(\widehat{\boldsymbol{b}})\} + \frac{1}{\tau}\boldsymbol{X}_0\right]_{i,j} & (\text{if } (i,j) \in \Omega^{(0)}), \\ [\boldsymbol{X}_0]_{i,j} & (\text{otherwise}), \end{cases}$$

where $\mathcal{L}'^{*}_{\Omega^{(0)}} : \mathbb{R}^{mp} \to \mathbb{R}^{mN_0 \times mI_0}$ denotes the adjoint operator of $\mathcal{L}'_{\Omega^{(0)}}$ satisfying $\left\langle \mathcal{L}'_{\Omega^{(0)}}(\boldsymbol{X}), \boldsymbol{v} \right\rangle_{\mathbb{R}^{mp}} = \left\langle \boldsymbol{X}, \mathcal{L}'^{*}_{\Omega^{(0)}}(\boldsymbol{v}) \right\rangle_{\mathbb{R}^{mN_0 \times mI_0}}$ for all $\boldsymbol{X} \in \mathbb{R}^{mN_0 \times mI_0}$ and $\boldsymbol{v} \in \mathbb{R}^{mp}$, and $\operatorname{shrink}(\cdot)$ is defined in (6.4). For the function g, the proximity operator of the indicator function ι_{M_1} is the orthogonal projection P_{M_1} onto the subspace M_1, i.e.,

$$\operatorname{prox}_{\gamma g}(\mathcal{X}) = P_{M_1}(\mathcal{X}) := \operatorname*{arg\,min}_{\mathcal{Y} \in M_1} \|\mathcal{X} - \mathcal{Y}\|_{\mathcal{H}_0}.$$

Since $M_1 \subset M_2 \subset \mathcal{H}_0$, we have by [Deu01, 5.14, Reduction principle]

$$P_{M_1}(\mathcal{X}) = P_{M_1}|M_2 \circ P_{M_2}(\mathcal{X}).$$

Note that '$|M_2$' in $P_{M_1}|M_2$ stands for the restriction of the domain $\mathcal{H}_0$ of the operator P_{M_1} to the subspace M_2 $(\subset \mathcal{H}_0)$. The orthogonal projection $P_{M_2} : \mathcal{H}_0 \to M_2$ and $P_{M_1}|M_2 : M_2 \to M_1$ respectively can be calculated as

$$P_{M_2}(\mathcal{X}) := \operatorname*{arg\,min}_{\mathcal{Y} \in M_2} \|\mathcal{X} - \mathcal{Y}\|_{\mathcal{H}_0} = [P_{\mathfrak{S}_0}(\boldsymbol{X}_0), \ldots, P_{\mathfrak{S}_n}(\boldsymbol{X}_n)]$$

and

$$P_{M_1}|M_2(\mathcal{X}) = \left[\widetilde{\overline{\boldsymbol{X}}}_{(0)}, \ldots, \widetilde{\overline{\boldsymbol{X}}}_{(n)}\right],$$

where $P_{\mathfrak{S}_k} : \mathbb{R}^{mN_k \times mI_k} \to \mathfrak{S}_k$ $(k = 0, \ldots, n)$ stands for the orthogonal projection onto the subspace $\mathfrak{S}_k$, and $\widetilde{\overline{\boldsymbol{X}}}_{(k)}$ $(k = 0, \ldots, n)$ is $\widetilde{(\cdot)}$ of the mode-k unfolding of

$$\overline{\mathcal{X}} := \frac{1}{n+1}\left(\operatorname{refold}(\underset{\sim}{\boldsymbol{X}}_0) + \cdots + \operatorname{refold}(\underset{\sim}{\boldsymbol{X}}_n)\right) \in \mathscr{T}^{\mathbb{A}_m}.$$

For $P_{\mathfrak{S}_k}$ $(k = 0, \ldots, n)$, let $\boldsymbol{E}^{(k)}_{p,q,\ell} := \boldsymbol{E}^{(k)}_{p,q}\mathbf{i}_\ell \in \mathbb{A}_m^{N_k \times I_k}$ $(\ell = 1, \ldots, m)$, where $\boldsymbol{E}^{(k)}_{p,q} \in \mathbb{R}^{N_k \times I_k}$ is the matrix only whose (p,q)-th entry $(p = 1, \ldots, N_k,\ q = 1, \ldots, I_k)$ is 1 and all other entries are 0. Then, we can easily verify that

$$\langle \widetilde{\boldsymbol{E}}^{(k)}_{p,q,\ell}, \widetilde{\boldsymbol{E}}^{(k)}_{p',q',\ell'} \rangle_{\mathbb{R}^{mN_k \times mI_k}} = \begin{cases} m & (\text{if } (p,q,\ell) = (p',q',\ell')), \\ 0 & (\text{otherwise}). \end{cases}$$

Therefore, $\{\frac{1}{\sqrt{m}}\widetilde{\boldsymbol{E}}^{(k)}_{p,q,\ell}\}_{p=1,q=1,\ell=1}^{N_k,I_k,m}$ is an orthonormal basis of $\mathfrak{S}_k$ and thus $P_{\mathfrak{S}_k}(\boldsymbol{X}_k)$ can be easily calculated as:

$$P_{\mathfrak{S}_k}(\boldsymbol{X}_k) = \frac{1}{m}\sum_{p=1}^{N_k}\sum_{q=1}^{I_k}\sum_{\ell=1}^{m}\langle \boldsymbol{X}_k, \widetilde{\boldsymbol{E}}^{(k)}_{p,q,\ell}\rangle_{\mathbb{R}^{mN_k\times mI_k}}\widetilde{\boldsymbol{E}}^{(k)}_{p,q,\ell}.$$

Now, we have

$$\begin{aligned}\operatorname{prox}_{\gamma g}(\mathcal{X}) &= P_{M_1}|M_2 \circ P_{M_2}(\mathcal{X})\\ &= P_{M_1}|M_2\left[P_{\mathfrak{S}_0}(\boldsymbol{X}_0),\ldots,P_{\mathfrak{S}_n}(\boldsymbol{X}_n)\right]\\ &= \left[\widetilde{\overline{\boldsymbol{X}}}^{\star}_{(0)},\ldots,\widetilde{\overline{\boldsymbol{X}}}^{\star}_{(n)}\right],\end{aligned}$$

where $\overline{\mathcal{X}}^{\star} := \frac{1}{n+1}\left(\operatorname{refold}(\underset{\sim}{\boldsymbol{X}}^{\star}_0) + \cdots + \operatorname{refold}(\underset{\sim}{\boldsymbol{X}}^{\star}_n)\right) \in \mathscr{T}^{\mathbb{A}_m}$ and $\boldsymbol{X}^{\star}_k := P_{\mathfrak{S}_k}(\boldsymbol{X}_k) \in \mathfrak{S}_k$ $(k = 0,\ldots,n)$. Since all ingredients are identified, we can summarize the proposed hypercomplex tensor completion algorithm in Algorithm 6. Here, $(t_u)_{u\geq 0} \subset [0,2]$ satisfies $\sum_{u\geq 0} t_u(2-t_u) = +\infty$, $\gamma \in (0,+\infty)$. The same as noted in Remark 8, the shrinkage operator does not always keep the special structure of non-trivial mapping, i.e., $\operatorname{shrink}(\widetilde{\boldsymbol{A}},\tau) \in \mathfrak{S}_{\mathbb{A}_m}(M,N)$ for $\boldsymbol{A} \in \mathbb{A}_m^{M\times N}$ does not hold in general. However, it keeps the structure in complex and quaternion case as shown in Corollary 5. Moreover, the shrinkage operator is the only nonlinear operation in Algorithm 6, so especially if $m \leq 4$ we have $\boldsymbol{X}^{(u)}_k \in \mathfrak{S}_k$ i.e., $P_{\mathfrak{S}_k}(\boldsymbol{X}^{(u)}_k) = \boldsymbol{X}^{(u)}_k$ for all $u \geq 0$ and $k = 0,\ldots,n$. Since both unfolding and refolding procedures just change the permutation of entries, so they have no interactions among real and all imaginary parts. Especially if $m = 1$ (i.e., $\mathbb{A}_m = \mathbb{R}$), Algorithm 1 is identical to the original DRS for low-n-rank tensor completion algorithm (DR-TR) proposed in [Gan+11].

Lastly, we state the convergence of the proposed algorithm.

Theorem 9 (Convergence of $\mathbb{A}_m$-DRS)**.** *Let parameters of Algorithm 6 be chosen so that* $\gamma \in (0,+\infty)$*,* $(t_u)_{u\geq 0} \subset [0,2]$ *satisfying* $\sum_{u\geq 0} t_u(2-t_u) = +\infty$*. Then, the output of Algorithm 6 converges to a minimizer of (8.1).*

Proof. From (8.2), it follows immediately that

$$\operatorname{dom}(g) = \{\mathcal{X} | \mathcal{X} \in M_1\} \neq \varnothing,$$

where $\operatorname{dom}(f) := \{x \in \mathcal{H} | f(x) < +\infty\}$ is the domain of a function $f : \mathcal{H} \to \mathbb{R} \cup \{+\infty\}$. Therefore, the convergence of the Algorithm 6 can be verified by the original DRS in [Gan+11]. □

Remark 12. In this paper, we employ the DRS for solving (8.1) but it can be also solved by other advanced convex optimization techniques such as the *alternating*

Algorithm 6: $\mathbb{A}_m$-Douglas-Rachford splitting for hypercomplex tensor completion ($\mathbb{A}_m$-DRS)

Input : $\mathcal{L}'_{\Omega^{(0)}}, \Omega^{(0)}, \boldsymbol{b}, t_u, \lambda, c_\lambda, \gamma$

Output: Recovered tensor $\mathcal{X} \in \mathscr{T}^{\mathbb{A}_m}$

$u \leftarrow 0, \tau \leftarrow \gamma(n+1), \boldsymbol{X}_k^{(0)} \leftarrow \mathbf{0} \; (\forall k = 0, \ldots, n)$;

while *not converged* **do**

$\quad [\boldsymbol{X}_0^\star, \ldots, \boldsymbol{X}_n^\star] \leftarrow \left[P_{\mathfrak{S}_0}(\boldsymbol{X}_0^{(u)}), \ldots, P_{\mathfrak{S}_n}(\boldsymbol{X}_n^{(u)})\right]$;

$\quad \overline{\mathcal{X}}^\star \leftarrow \mathrm{mean}(\mathrm{refold}(\underset{\sim}{\boldsymbol{X}}_0^\star), \ldots, \mathrm{refold}(\underset{\sim}{\boldsymbol{X}}_n^\star))$;

$\quad$ **for** $k = 1, \ldots, n$ **do**

$$\mathrm{prox}_{\tau f_k}(2\widetilde{\overline{\boldsymbol{X}}}^\star_{(k)} - \boldsymbol{X}_k^{(u)}) \leftarrow \mathrm{shrink}(2\widetilde{\overline{\boldsymbol{X}}}^\star_{(k)} - \boldsymbol{X}_k^{(u)}, \tau);$$

$$\boldsymbol{X}_k^{(u+1)} \leftarrow \boldsymbol{X}_k^{(u)} + t_u\left[\mathrm{prox}_{\tau f_k}(2\widetilde{\overline{\boldsymbol{X}}}^\star_{(k)} - \boldsymbol{X}_k^{(u)}) - \widetilde{\overline{\boldsymbol{X}}}^\star_{(k)}\right];$$

$\quad$ **end**

$$\left[\mathrm{prox}_{\tau f_0}(2\widetilde{\overline{\boldsymbol{X}}}^\star_{(0)} - \boldsymbol{X}_0^{(u)})\right]_{i,j} \leftarrow \begin{cases} \left[\frac{\tau}{\lambda\tau+1}\{\lambda\mathcal{L}'^*_{\Omega^{(0)}}(\widehat{\boldsymbol{b}}) + \frac{1}{\tau}(2\widetilde{\overline{\boldsymbol{X}}}^\star_{(0)} - \boldsymbol{X}_0^{(u)})\}\right]_{i,j} & (\text{if } (i,j) \in \Omega^{(0)}) \\ [2\widetilde{\overline{\boldsymbol{X}}}^\star_{(0)} - \boldsymbol{X}_0^{(u)}]_{i,j} & (\text{otherwise}) \end{cases};$$

$$\boldsymbol{X}_0^{(u+1)} \leftarrow \boldsymbol{X}_0^{(u)} + t_u\left[\mathrm{prox}_{\tau f_0}(2\widetilde{\overline{\boldsymbol{X}}}^\star_{(0)} - \boldsymbol{X}_0^{(u)}) - \widetilde{\overline{\boldsymbol{X}}}^\star_{(0)}\right];$$

$\quad u \leftarrow u + 1$;

end

$[\boldsymbol{X}_0^\star, \ldots, \boldsymbol{X}_n^\star] \leftarrow \left[P_{\mathfrak{S}_0}(\boldsymbol{X}_0^{(u)}), \ldots, P_{\mathfrak{S}_n}(\boldsymbol{X}_n^{(u)})\right]$;

$\mathcal{X} \leftarrow \mathrm{mean}(\mathrm{refold}(\underset{\sim}{\boldsymbol{X}}_0^\star), \ldots, \mathrm{refold}(\underset{\sim}{\boldsymbol{X}}_n^\star))$;

direction method of multipliers (ADMM) [Boy+11] and the *primal-dual splitting (PDS)* [Con13; Vũ13].

8.3 Numerical Examples

We examine the efficiency of the proposed algorithm in the context of hypercomplex tensor completion problem. We perform experiments several hypercomplex domains, complex, quaternion and octonion, and compare several state-of-the-art algorithms. The results of performance comparisons are summarized in TABLE 8.2.

For the complex case, we compare three algorithms, the proposed $\mathbb{A}_m$-DRS for the complex case, the original DRS [Gan+11] extended to the complex domain with complex SVD (DRS-CSVD) and the part-wise real DRS ($\mathbb{R}^2$-DRS-PW). For the quaternion case, we compare four algorithms, the proposed $\mathbb{A}_m$-DRS for the quaternion case, the DRS extended to quaternion domain using quaternion SVD, which is available in the Matlab quaternion toolbox [SB13] (DRS-QSVD), part-wise complex-valued DRS ($\mathbb{C}^2$-DRS-PW) and the part-wise real DRS ($\mathbb{R}^4$-DRS-PW). For octonion case, there is no existing method, so we compare the proposed method and four part-wise methods, part-wise quaternion DRS ($\mathbb{H}^2$-DRS-PW), part-wise complex DRS ($\mathbb{C}^4$-DRS-PW) and the part-wise real DRS ($\mathbb{R}^8$-DRS-PW). The $\mathbb{R}^m$-DRS-PW ($m = 2, 4, 8$) optimizes all real and imaginary parts separately with the original real DRS. The $\mathbb{C}^m$-DRS-PW ($m = 2, 4$) decomposes a hypercomplex tensor to two (for quaternion) or four (octonion) complex tensors and optimizes them separately with the complex DRS (i.e., DRS-CSVD or $\mathbb{A}_m$-DRS for the complex case). The $\mathbb{H}^2$-DRS-PW decomposes an octonion tensor into two quaternion tensors and optimizes them separately with the quaternion DRS (i.e., DRS-QSVD or $\mathbb{A}_m$-DRS for the quaternion case). Note that the proposed $\mathbb{A}_m$-DRS can be applied to general C-D case, and that DRS-QSVD and $\mathbb{H}^2$-DRS-PW are also new algorithms since the quaternion tensor completion itself is a new problem to the best of our knowledge. These three algorithms are implemented with Matlab[1] and the SVD in $\mathbb{A}_m$-DRS and the all part-wise methods are based on the QR-decomposition.

In each experiment, we generate low-n-rank hypercomplex tensor $\mathcal{X}_0$ which we used as ground truth. We fix the dimension r of a 'core tensor' $\mathcal{C} \in \mathbb{A}_m^{r\times\cdots\times r}$ ($\mathbb{A}_m = \mathbb{C}$, $\mathbb{H}$ or $\mathbb{O}$). Then we generate matrices $\mathbf{\Psi}^{(1)}, \dots, \mathbf{\Psi}^{(n)}$ with $\mathbf{\Psi}^{(k)} \in \mathbb{A}_m^{N_k\times r}$ and set $\mathcal{X}_0 = \mathcal{C} \times_1 \mathbf{\Psi}^{(1)} \times_2 \cdots \times_n \mathbf{\Psi}^{(n)} \in \mathscr{T}^{\mathbb{A}_m}$, where $\times_k$ ($k = 1, \dots, n$) is the *k-mode product* satisfying $\mathcal{Y} = \mathcal{X} \times_k \mathbf{\Psi}^{(k)} \Leftrightarrow \boldsymbol{Y}_{(k)} = \mathbf{\Psi}^{(k)}\boldsymbol{X}_{(k)}$. An example of k-mode product in the case where the number of modes $n = 3$ is illustrated in Fig. 8.1. All real and imaginary parts of all entries in $\mathcal{C}$ and $\mathbf{\Psi}^{(k)}$ are i.i.d. from $\mathcal{N}(0, 1)$. With this construction, the n-rank of $\mathcal{X}_0$ equals to $[mr, \dots, mr]$ almost surely for $\mathbb{A}_m = \mathbb{C}$ or $\mathbb{H}$ from Lemma 4. Note that it does not hold for the octonion $\mathbb{O}$ case, so we set $\Im_\ell(\mathbf{\Psi}^{(k)}) = \mathbf{0}$ for $\ell = 1, \dots 7$ and $k = 1, \dots, n$ to heuristically ensure the low-rankness. For investigating the limitation of recovery, we try various percentage ρ of the entries to be known and randomly chose the support of the known entries. The value and the locations of the known entries of $\mathcal{X}_0$ are used as inputs for the algorithms. For the parameters, we set $\lambda = n$ and $t_k = 1$.

TABLE 8.2(a) shows the performance comparison in $\mathbb{C}$. It shows that the performance of the $\mathbb{A}_m$-DRS and the DRS-CSVD are the same since these two methods are mathematically equivalent. $\mathbb{R}^2$-DRS-PW do not converge to the optimal solution since it cannot approximate the ground truth. In $\mathbb{C}^{10\times10\times10\times10}$, the performance is significantly worse if ρ is less than 0.3, so it is the recovery limitation for this case.

[1] The codes for these examples are available by request.

TABLE 8.2(b) shows the performance comparison in $\mathbb{H}$. It shows that the performance of the $\mathbb{A}_m$-DRS and the DRS-QSVD are the same since these two methods are mathematically equivalent. Both $\mathbb{C}^2$-DRS-PW and $\mathbb{R}^4$-DRS-PW do not converge to the optimal solution since it cannot approximate the ground truth. However, the performance of $\mathbb{C}^2$-DRS-PW is better than $\mathbb{R}^4$-DRS-PW since it can utilize the part of the correlations among each part. In $\mathbb{H}^{10\times10\times10\times10}$, the performance is significantly worse if ρ is less than 0.2, so it is the recovery limitation for this case.

TABLE 8.2(c) shows the performance comparison in $\mathbb{O}$. Proposed $\mathbb{A}_m$-DRS outperforms all part-wise methods especially in the cases of higher-order tensors. This results may indicate that utilization of correlation among each part is more important in recovering higher-oder tensors.

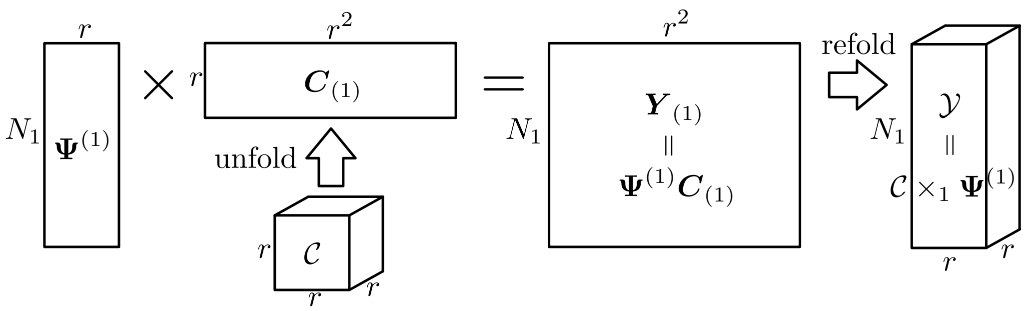
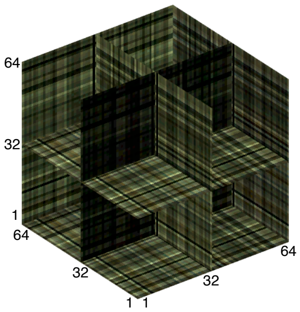


Figure 8.1.: Example of k-mode product ($n = 3$)

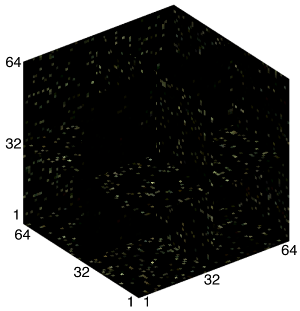
Fig.8.2 depicts some slices of (a) original, (b) observed tensors with $\rho = 0.1$, (c) observed tensors with $\rho = 0.01$, and the completion results of the lower right case in TABLE 8.2 (i.e., $\rho = 0.1$) by (d) $\mathbb{C}^2$-DRS-PW, (e) $\mathbb{R}^4$-DRS-PW, (f) $\mathbb{A}_m$-DRS, and (g) that for $\rho = 0.01$ by $\mathbb{A}_m$-DRS. The observed tensor has $64 \times 64 \times 64 \approx 260K$ quaternion entries (pixels). Each pixel in tensors is represented by the three imaginary parts of a quaternion as the RGB color space (see Appendix A.7.2). It shows that the proposed method indeed recovers well both color information and low-rank structure of the original tensor even for very limited observation, while both $\mathbb{C}^2$-DRS-PW and $\mathbb{R}^4$-DRS-PW cannot recover the color information of the original tensor since it cannot utilize the correlation of each color space. If ρ is less than 0.01, the tensor cannot be recovered even by $\mathbb{A}_m$-DRS.

8.4 Summary

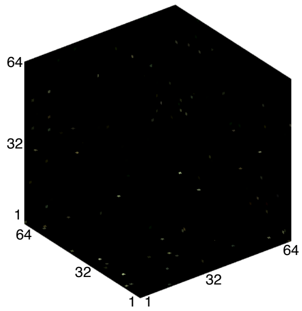
We have proposed an algorithmic solution to hypercomplex tensor completion problem based on Douglas-Rachford splitting. This solution utilizes the useful definitions of SVD that we have proposed and is available in general C-D domains. Numerical experiments show that the proposed algorithm recovers much more faith-



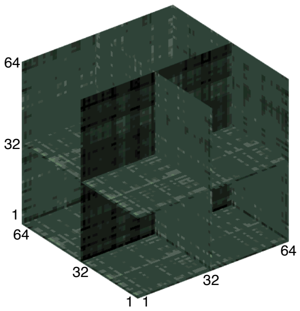
(a) Original tensor



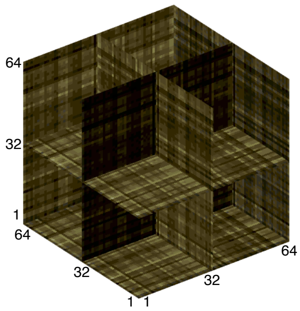
(b) Observed tensor ($\rho = 0.1$)



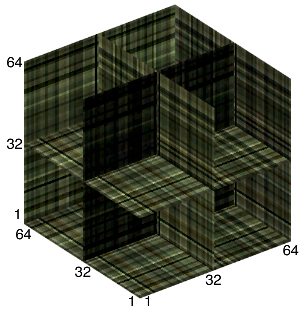
(c) Observed tensor ($\rho = 0.01$)



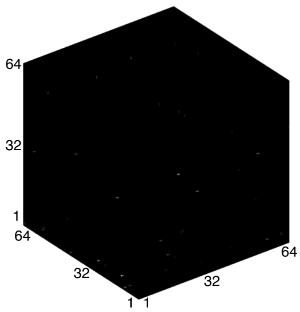
(d) Completed by $\mathbb{C}^2$-DRS-PW (error=1.9e-2, $\rho = 0.1$)



(e) Completed by $\mathbb{R}^4$-DRS-PW (error=9.5e-2, $\rho = 0.1$)



(f) Completed by $\mathbb{A}_m$-DRS (error=1.3e-4, $\rho = 0.1$)



(g) Completed by $\mathbb{A}_m$-DRS (error=3.3e-2, $\rho = 0.01$)

Figure 8.2.: Completion of 3-dimensional quaternion tensor

fully the original hypercomplex tensor information from very limited observations than state-of-the-art existing algorithms.

Table 8.2.: Performance comparisons

(a) Performance comparison in $\mathbb{C}$

$\mathbb{C}^{10\times10\times10\times10}$, $\rho = 0.4$, $r = 2$		
Algorithm	# iter.	error
$\mathbb{A}_m$-DRS	1,154	6.9e-4
DRS-CSVD	1,154	6.9e-4
$\mathbb{R}^2$-DRS-PW	1,259	8.2

$\mathbb{C}^{10\times10\times10\times10}$, $\rho = 0.3$, $r = 2$		
Algorithm	# iter.	error
$\mathbb{A}_m$-DRS	15,077	2.8e-1
DRS-CSVD	15,077	2.8e-1
$\mathbb{R}^2$-DRS-PW	17,048	6.3

$\mathbb{C}^{64\times64\times64}$, $\rho = 0.6$, $r = 4$		
Algorithm	# iter.	error
$\mathbb{A}_m$-DRS	1,371	2.1e-5
DRS-CSVD	1,371	2.1e-5
$\mathbb{R}^2$-DRS-PW	1,428	1.0e-1

(b) Performance comparison in $\mathbb{H}$

$\mathbb{H}^{10\times10\times10\times10}$, $\rho = 0.6$, $r = 2$		
Algorithm	# iter.	error
$\mathbb{A}_m$-DRS	898	1.5e-4
DRS-QSVD	898	1.5e-4
$\mathbb{C}^2$-DRS-PW	1,025	11.5
$\mathbb{R}^4$-DRS-PW	1,077	35.6

$\mathbb{H}^{10\times10\times10\times10}$, $\rho = 0.2$, $r = 2$		
Algorithm	# iter.	error
$\mathbb{A}_m$-DRS	14,796	19.5
DRS-QSVD	14,796	19.5
$\mathbb{C}^2$-DRS-PW	11,363	30.5
$\mathbb{R}^4$-DRS-PW	15,059	53.9

$\mathbb{H}^{64\times64\times64}$, $\rho = 0.1$, $r = 4$		
Algorithm	# iter.	error
$\mathbb{A}_m$-DRS	1,708	1.2e-4
DRS-QSVD	1,708	1.2e-4
$\mathbb{C}^2$-DRS-PW	405	1.9e-2
$\mathbb{R}^4$-DRS-PW	969	9.5e-2

(c) Performance comparison in $\mathbb{O}$

$\mathbb{O}^{10\times10\times10\times10}$, $\rho = 0.6$, $r = 2$		
Algorithm	# iter.	error
$\mathbb{A}_m$-DRS	10,812	9.12e-4
$\mathbb{H}^2$-DRS-PW	7,930	1.8
$\mathbb{C}^4$-DRS-PW	5,991	1.8
$\mathbb{R}^8$-DRS-PW	4,831	1.9

$\mathbb{O}^{10\times10\times10\times10}$, $\rho = 0.4$, $r = 2$		
Algorithm	# iter.	error
$\mathbb{A}_m$-DRS	23,986	7.9
$\mathbb{H}^2$-DRS-PW	23,429	50.8
$\mathbb{C}^4$-DRS-PW	17,339	50.9
$\mathbb{R}^8$-DRS-PW	13,092	51.6

$\mathbb{O}^{32\times32\times32}$, $\rho = 0.6$, $r = 2$		
Algorithm	# iter.	error
$\mathbb{A}_m$-DRS	3,056	6.1e-5
$\mathbb{H}^2$-DRS-PW	3,408	1.9e-4
$\mathbb{C}^4$-DRS-PW	1,997	3.9e-4
$\mathbb{R}^8$-DRS-PW	969	8.0e-4

9 Conclusion

In this thesis, we have studied the algebraic translations of Cayley-Dickson linear systems and their signal processing applications.

In Chapter 2, we have introduced the basic definitions and properties of Cayley-Dickson number systems and two powerful optimization methods.

In Chapter 3, we have proposed an algebraic real translation of the hypercomplex valued (C-D) linear systems and show that this translation enables us to immediately obtain equivalent real models to hypercomplex linear models. This translation is designed by using jointly two new isomorphisms between real vector spaces. We have clarified the algebraic properties of the translation.

In Chapter 4, we then have presented a new hypercomplex adaptive algorithm named $\mathbb{A}_m$-APSM based on the APSM as an example of many potential algorithms through the proposed translation. The proposed algorithm covers wide range of hypercomplex adaptive filtering algorithms. Indeed, many existing adaptive algorithms in hypercomplex domain have been reproduced as special cases without using any special calculus such as $\mathbb{CR}$-calculus and $\mathbb{HR}$-calculus etc. Moreover, many new adaptive algorithms have also been established as special cases in the octonion domain, which have not yet been reported elsewhere to the best of the authors knowledge.

In Chapter 5, as a nonlinear extension, we have extended the real RKHS into the general C-D domain for nonlinear estimation problem. The extended RKHS inherits many useful properties of the real RKHS.

In Chapter 6, we have proposed the new notion $\mathbb{R}$-eigenvalue. It can be defined and calculated for any C-D square matrices We also have shown that the $\mathbb{R}$-eigenvalues are natural extensions of the original eigenvalues. Moreover, we have newly defined the C-D singular value decomposition (C-D SVD) and $\mathbb{R}$-rank of general C-D matrices based on the $\mathbb{R}$-eigenvalue, and have clarified their properties and the relations to known results.

In Chapter 7, we have proposed algorithmic solutions to hypercomplex principal component pursuit and non-negative low rank matrix completion based on a proximal splitting technique. These solutions solve the hypercomplex principal component pursuit, which is a convex relaxation of hypercomplex robust principal component analysis with a new sparsity measure of C-D matrices, and low rank matrix completion problem with non-negative constraint. Both solusions utilize useful mathematical tools including C-D SVD and $\mathrm{rank}^{\mathbb{R}}$ proposed in Chapter 6.

In Chapter 8, we have proposed an algorithmic solution to hypercomplex tensor completion problem based on a convex optimization technique. This solution utilizes new definitions of SVD and best low rank approximation of matrices proposed in Chapter 6 and it is based on algebraic translations of C-D number systems proposed in Chapter 3.

Since the proposed frameworks are designed for general C-D domain, so numerical examples can be performed in many kinds of hypercomplex domain including complex, quaternion and octonion throughout this thesis. All of them show that the proposed methods achieve excellent performances than state-of-the-art part-wise algorithms.

Appendices

A

A.1 Proof of Proposition 1

Proof. $\mathbb{C}^{2\times 2}_{\mathrm{rank}\leq(1,1)}$ and $\mathbb{C}^{2\times 2}_{\mathrm{rank}\leq 1}$ can be respectively also represented as

$$\mathbb{C}^{2\times 2}_{\mathrm{rank}\leq(1,1)} = \left\{ \begin{bmatrix} \alpha_1 \boldsymbol{a} \\ \alpha_2 \boldsymbol{a} \end{bmatrix} + \begin{bmatrix} \beta_1 \boldsymbol{b} \\ \beta_2 \boldsymbol{b} \end{bmatrix} \imath \,\middle|\, (\alpha_1, \alpha_2), (\beta_1, \beta_2), \boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^{1\times 2} \right\}$$

and

$$\begin{aligned} \mathbb{C}^{2\times 2}_{\mathrm{rank}\leq 1} &= \left\{ \begin{bmatrix} (\alpha_1' + \beta_1'\imath)(\boldsymbol{a}' + \boldsymbol{b}'\imath) \\ (\alpha_2' + \beta_2'\imath)(\boldsymbol{a}' + \boldsymbol{b}'\imath) \end{bmatrix} \,\middle|\, (\alpha_1', \alpha_2'), (\beta_1', \beta_2'), \boldsymbol{a}', \boldsymbol{b}' \in \mathbb{R}^{1\times 2} \right\} \\ &= \left\{ \begin{bmatrix} \alpha_1'\boldsymbol{a}' - \beta_1'\boldsymbol{b}' \\ \alpha_2'\boldsymbol{a}' - \beta_2'\boldsymbol{b}' \end{bmatrix} + \begin{bmatrix} \beta_1'\boldsymbol{a}' + \alpha_1'\boldsymbol{b}' \\ \beta_2'\boldsymbol{a}' + \alpha_2'\boldsymbol{b}' \end{bmatrix} \imath \,\middle|\, (\alpha_1', \alpha_2'), (\beta_1', \beta_2'), \boldsymbol{a}', \boldsymbol{b}' \in \mathbb{R}^{1\times 2} \right\}. \end{aligned}$$

We first prove $\mathbb{C}^{2\times 2}_{\mathrm{rank}\leq(1,1)} \not\subset \mathbb{C}^{2\times 2}_{\mathrm{rank}\leq 1}$. Let

$$\boldsymbol{C} := \begin{bmatrix} \alpha_1 \boldsymbol{a} \\ t\alpha_1 \boldsymbol{a} \end{bmatrix} + \begin{bmatrix} \beta_1 \boldsymbol{b} \\ s\beta_1 \boldsymbol{b} \end{bmatrix} \imath \in \mathbb{C}^{2\times 2}_{\mathrm{rank}\leq(1,1)},$$

where $\boldsymbol{a}$ and $\boldsymbol{b}$ in $\mathbb{R}^{1\times 2}$ are linearly independent, $\alpha_1 \neq 0$, $\beta_1 \neq 0$ and $s \neq t$. Suppose $\boldsymbol{C} \in \mathbb{C}^{2\times 2}_{\mathrm{rank}\leq 1}$, i.e.,

$$\begin{bmatrix} \alpha_1 \boldsymbol{a} \\ t\alpha_1 \boldsymbol{a} \end{bmatrix} + \begin{bmatrix} \beta_1 \boldsymbol{b} \\ s\beta_1 \boldsymbol{b} \end{bmatrix} \imath = \begin{bmatrix} \alpha_1'\boldsymbol{a}' - \beta_1'\boldsymbol{b}' \\ \alpha_2'\boldsymbol{a}' - \beta_2'\boldsymbol{b}' \end{bmatrix} + \begin{bmatrix} \beta_1'\boldsymbol{a}' + \alpha_1'\boldsymbol{b}' \\ \beta_2'\boldsymbol{a}' + \alpha_2'\boldsymbol{b}' \end{bmatrix} \imath. \tag{A.1}$$

From (A.1), we have

$$\begin{aligned} t\alpha_1'\boldsymbol{a}' - t\beta_1'\boldsymbol{b}' &= \alpha_2'\boldsymbol{a}' - \beta_2'\boldsymbol{b}', \\ s\beta_1'\boldsymbol{a}' + s\alpha'\boldsymbol{b}' &= \beta_2'\boldsymbol{a}' + \alpha_2'\boldsymbol{b}'. \end{aligned}$$

i) If $\boldsymbol{a}'$ and $\boldsymbol{b}'$ are linearly independent, we have

$$(\alpha_2', \beta_2') = t(\alpha_1', \beta_1') = s(\alpha_1', \beta_1')$$

and hence $(\alpha_1', \beta_1') = (0,0)$ since $s \neq t$. This implies $\alpha_1 \boldsymbol{a} = \beta_1 \boldsymbol{b} = \boldsymbol{0}$ from (A.1) and contradicts to the linearly independence of $\boldsymbol{a}, \boldsymbol{b}$.

ii) If $\boldsymbol{a}'$ and $\boldsymbol{b}'$ are linearly dependent, there exists $u \in \mathbb{R}$ such that $\boldsymbol{b}' = u\boldsymbol{a}'$. From (A.1) this implies that

$$\alpha_1 \boldsymbol{a} = (\alpha_1' - \beta_1' u)\boldsymbol{a}',$$
$$\beta_1 \boldsymbol{b} = (\beta_1' + \alpha_1' u)\boldsymbol{a}'$$

and also contradicts to the linearly independence of $\boldsymbol{a}, \boldsymbol{b}$.

Therefore, $\boldsymbol{C} \notin \mathbb{C}^{2\times 2}_{\text{rank}\leq 1}$ and thus $\mathbb{C}^{2\times 2}_{\text{rank}\leq (1,1)} \not\subset \mathbb{C}^{2\times 2}_{\text{rank}\leq 1}$.

We next prove $\mathbb{C}^{2\times 2}_{\text{rank}\leq 1} \not\subset \mathbb{C}^{2\times 2}_{\text{rank}\leq (1,1)}$. Let

$$\boldsymbol{C}' := \begin{bmatrix} \alpha_1' \boldsymbol{a}' - \beta_1' \boldsymbol{b}' \\ s\alpha_1' \boldsymbol{a}' - t\beta_1' \boldsymbol{b}' \end{bmatrix} + \begin{bmatrix} \beta_1' \boldsymbol{a}' + \alpha_1' \boldsymbol{b}' \\ t\beta_1' \boldsymbol{a}' + s\alpha_1' \boldsymbol{b}' \end{bmatrix} \imath \in \mathbb{C}^{2\times 2}_{\text{rank}\leq 1},$$

where $\boldsymbol{a}'$ and $\boldsymbol{b}'$ in $\mathbb{R}^{1\times 2}$ are linearly independent, and $s \neq t$ satisfy $\alpha_2' = s\alpha_1'$ and $\beta_2' = t\beta_1'$. Suppose $\boldsymbol{C}' \in \mathbb{C}^{2\times 2}_{\text{rank}\leq (1,1)}$ .i.e,

$$\begin{bmatrix} \alpha_1' \boldsymbol{a}' - \beta_1' \boldsymbol{b}' \\ s\alpha_1' \boldsymbol{a}' - t\beta_1' \boldsymbol{b}' \end{bmatrix} + \begin{bmatrix} \beta_1' \boldsymbol{a}' + \alpha_1' \boldsymbol{b}' \\ t\beta_1' \boldsymbol{a}' + s\alpha_1' \boldsymbol{b}' \end{bmatrix} \imath = \begin{bmatrix} \alpha_1 \boldsymbol{a} \\ \alpha_2 \boldsymbol{a} \end{bmatrix} + \begin{bmatrix} \beta_1 \boldsymbol{b} \\ \beta_2 \boldsymbol{b} \end{bmatrix} \imath. \tag{A.2}$$

By comparing the real and imaginary parts of both sides in (A.2), we have

$$\alpha_2(\alpha_1' \boldsymbol{a}' - \beta_1' \boldsymbol{b}') = \alpha_1(s\alpha_1' \boldsymbol{a}' - t\beta_1' \boldsymbol{b}'),$$
$$\beta_2(\beta_1' \boldsymbol{a}' - \alpha_1' \boldsymbol{b}') = \beta_1(t\beta_1' \boldsymbol{a}' - s\alpha_1' \boldsymbol{b}')$$

and

$$\alpha_2 \alpha_1' = s\alpha_1 \alpha_1', \quad \alpha_2 \beta_1' = t\alpha_1 \beta_1',$$
$$\beta_2 \beta_1' = t\beta_1 \beta_1', \quad \beta_2 \alpha_1' = s\beta_1 \alpha_1',$$

since $\boldsymbol{a}'$ and $\boldsymbol{b}'$ are linearly independent. Hence, $(\alpha_2, \beta_2) = s(\alpha_1, \beta_1) = t(\alpha_1, \beta_1)$ and hence $(\alpha_1, \beta_1) = (0,0)$ since $s \neq t$. This implies $\alpha_1' \boldsymbol{a}' - \beta_1' \boldsymbol{b}' = \boldsymbol{0}$ from (A.2) and contradicts to the linear independence of $\boldsymbol{a}', \boldsymbol{b}'$. Therefore, $\boldsymbol{C}' \notin \mathbb{C}^{2\times 2}_{\text{rank}\leq (1,1)}$ and thus $\mathbb{C}^{2\times 2}_{\text{rank}\leq 1} \not\subset \mathbb{C}^{2\times 2}_{\text{rank}\leq (1,1)}$. □

A.2 Proof of Lemma 2

Proof. Proof of 1)

This is obvious from (3.6) and Lemma 1-3) and Lemma 1-6).
Proof of 2)

If $\ell = 1$, it is obvious from Lemma 2-1). Suppose $\ell \geq 2$, and let

$$\boldsymbol{L}_M^{(\ell)\top} \boldsymbol{L}_M^{(\ell)} = \begin{bmatrix} \boldsymbol{B}_{1,1}^{(\ell)} & \boldsymbol{B}_{1,2}^{(\ell)} & \cdots & \boldsymbol{B}_{1,m}^{(\ell)} \\ \boldsymbol{B}_{2,1}^{(\ell)} & \boldsymbol{B}_{2,2}^{(\ell)} & \cdots & \boldsymbol{B}_{2,m}^{(\ell)} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{B}_{m,1}^{(\ell)} & \boldsymbol{B}_{m,2}^{(\ell)} & \cdots & \boldsymbol{B}_{m,m}^{(\ell)} \end{bmatrix},$$

where $\boldsymbol{B}_{p,q}^{(\ell)} \in \mathbb{R}^{M \times M}$, $\forall \ell = 2, \ldots, m$ and $\forall p, q = 1, \ldots, m$. Then, by using (3.6), Lemma 1-4) and Lemma 1-5), $\boldsymbol{B}_{p,q}^{(\ell)}$ can be expressed as

$$\boldsymbol{B}_{p,q}^{(\ell)} = \sum_{k=1}^{m} \delta_{k,p}^{(\ell)} \delta_{k,q}^{(\ell)} \boldsymbol{I}_M = \sum_{k=1}^{m} \delta_{k,\ell}^{(p)} \delta_{k,\ell}^{(q)} \boldsymbol{I}_M = \begin{cases} \boldsymbol{I}_M & (\text{if } p = q) \\ \boldsymbol{0}_{M \times M} & (\text{otherwise}). \end{cases}$$

Hence $\boldsymbol{L}_M^{(\ell)\top} \boldsymbol{L}_M^{(\ell)} = \boldsymbol{I}_{mM}$.

Proof of 3)

Since this property holds obviously for $\mathbb{R}$ and $\mathbb{C}$, we only show the case where $\mathbb{A}_m = \mathbb{H}$. In this case $\boldsymbol{L}_M^{(\ell)}$ $(\ell \in \{1, \ldots, 4\})$ are given as in Example 2. We can easily verify this property by simple algebra. □

A.3 Proof of Theorem 1

A.3.1 Proof of 1)

Proof. Let $\boldsymbol{A} := \sum_{\ell=1}^{m} \boldsymbol{A}_\ell \mathbf{i}_\ell$, $\forall \boldsymbol{A}_\ell \in \mathbb{R}^{M \times N}$ and $\boldsymbol{B} := \sum_{\ell=1}^{m} \boldsymbol{B}_\ell \mathbf{i}_\ell$, , $\forall \boldsymbol{B}_\ell \in \mathbb{R}^{M \times N}$ $(\ell \in \{1, \ldots, m\})$. Since $\boldsymbol{A} + \boldsymbol{B} = \sum_{\ell=1}^{m} (\boldsymbol{A}_\ell + \boldsymbol{B}_\ell) \mathbf{i}_\ell$,

$$\widehat{(\boldsymbol{A} + \boldsymbol{B})} = \begin{bmatrix} \boldsymbol{A}_1 + \boldsymbol{B}_1 \\ \vdots \\ \boldsymbol{A}_m + \boldsymbol{B}_m \end{bmatrix} = \widehat{\boldsymbol{A}} + \widehat{\boldsymbol{B}}.$$

Moreover, by using this equation,

$$\begin{aligned} \widetilde{(\boldsymbol{A} + \boldsymbol{B})} &= \left[\boldsymbol{L}_M^{(1)\top} \widehat{(\boldsymbol{A} + \boldsymbol{B})}, \ldots \boldsymbol{L}_M^{(m)\top} \widehat{(\boldsymbol{A} + \boldsymbol{B})} \right] \\ &= \left[\boldsymbol{L}_M^{(1)\top} \widehat{\boldsymbol{A}}, \ldots, \boldsymbol{L}_M^{(m)\top} \widehat{\boldsymbol{A}} \right] + \left[\boldsymbol{L}_M^{(1)\top} \widehat{\boldsymbol{B}}, \ldots, \boldsymbol{L}_M^{(m)\top} \widehat{\boldsymbol{B}} \right] \\ &= \widetilde{\boldsymbol{A}} + \widetilde{\boldsymbol{B}}. \end{aligned}$$

Remaining statements hold obviously. □

A.3.2 Proof of 2)

Proof. Let $\boldsymbol{A} := \sum_{\ell=1}^{m} \boldsymbol{A}_\ell \mathbf{i}_\ell$, $\forall \boldsymbol{A}_\ell \in \mathbb{R}^{M\times N}$, then $\boldsymbol{A}^{\mathsf{H}} = \boldsymbol{A}_1^\top \mathbf{i}_1 - \sum_{\ell=2}^{m} \boldsymbol{A}_\ell^\top \mathbf{i}_\ell$, so we have

$$\widehat{(\boldsymbol{A}^{\mathsf{H}})} = \begin{bmatrix} \boldsymbol{A}_1^\top \\ -\boldsymbol{A}_2^\top \\ \vdots \\ -\boldsymbol{A}_m^\top \end{bmatrix}$$

and

$$\widetilde{\boldsymbol{A}^{\mathsf{H}}} = \left[\boldsymbol{L}_N^{(1)\top} \widehat{(\boldsymbol{A}^{\mathsf{H}})}, \boldsymbol{L}_N^{(2)\top} \widehat{(\boldsymbol{A}^{\mathsf{H}})}, \cdots, \boldsymbol{L}_N^{(m)\top} \widehat{(\boldsymbol{A}^{\mathsf{H}})}\right].$$

For any $\ell = 1, 2, \ldots, m$, we have

$$\boldsymbol{L}_N^{(\ell)\top} \widehat{\boldsymbol{A}^{\mathsf{H}}} = \begin{bmatrix} \sum_{k=1}^{m} \delta_{k,1}^{(\ell)} \boldsymbol{A}_k^\top \\ \sum_{k=1}^{m} \delta_{k,2}^{(\ell)} \boldsymbol{A}_k^\top \\ \vdots \\ \sum_{k=1}^{m} \delta_{k,m}^{(\ell)} \boldsymbol{A}_k^\top \end{bmatrix}.$$

Hence the (α, β)-th block submatrix of size $\mathbb{R}^{N\times M}$ in $\widetilde{\boldsymbol{A}^{\mathsf{H}}}$ can be express as

$$\sum_{k=1}^{m} \delta_{k,\alpha}^{(\beta)} \boldsymbol{A}_k^\top \quad (\forall \alpha, \beta = 1, 2, \ldots, m). \tag{A.3}$$

On the other hand,

$$\tilde{\boldsymbol{A}}^\top = \left[\delta_{1,1}^{(\ell)} \boldsymbol{A}_1^\top - \sum_{k=2}^{m} \delta_{k,1}^{(\ell)} \boldsymbol{A}_k^\top, \ldots, \delta_{1,m}^{(\ell)} \boldsymbol{A}_1^\top - \sum_{k=2}^{m} \delta_{k,m}^{(\ell)} \boldsymbol{A}_k^\top\right].$$

Hence the (α, β)-th block submatrix of size $\mathbb{R}^{N\times M}$ in $\tilde{\boldsymbol{A}}^\top$ can be express as

$$\delta_{1,\beta}^{(\alpha)} \boldsymbol{A}_1^\top - \sum_{k=2}^{m} \delta_{k,\beta}^{(\alpha)} \boldsymbol{A}_k^\top \quad (\forall \alpha, \beta = 1, 2, \ldots, m). \tag{A.4}$$

If $\alpha = 1$, we have

$$\sum_{k=1}^{m} \delta_{k,\alpha}^{(\beta)} \boldsymbol{A}_k^\top = \sum_{k=1}^{m} \delta_{k,1}^{(\beta)} \boldsymbol{A}_k^\top = \boldsymbol{A}_\beta^\top$$

and from Lemma 1-6) and Lemma 1-5)

$$\begin{aligned}\delta_{1,\beta}^{(\alpha)}\boldsymbol{A}_1^\top - \sum_{k=2}^{m}\delta_{k,\beta}^{(\alpha)}\boldsymbol{A}_k^\top &= \delta_{1,\beta}^{(1)}\boldsymbol{A}_1^\top - \sum_{k=2}^{m}\delta_{k,\beta}^{(1)}\boldsymbol{A}_k^\top \\ &= \boldsymbol{A}_\beta^\top = \sum_{k=1}^{m}\delta_{k,\alpha}^{(\beta)}\boldsymbol{A}_k^\top\end{aligned}$$

for all $\beta \in \{1, \dots, m\}$. If $\beta = 1$, we can show this relation in the same way. For the case where $\alpha \geq 2$ and $\beta \geq 2$, we have, from using Lemma 1-4),

$$\begin{aligned}\delta_{1,\beta}^{(\alpha)}\boldsymbol{A}_1^\top - \sum_{k=2}^{m}\delta_{k,\beta}^{(\alpha)}\boldsymbol{A}_k^\top &= \delta_{1,\alpha}^{(\beta)}\boldsymbol{A}_1^\top + \sum_{k=2}^{m}\delta_{k,\alpha}^{(\beta)}\boldsymbol{A}_k^\top \\ &= \sum_{k=1}^{m}\delta_{k,\alpha}^{(\beta)}\boldsymbol{A}_k^\top \end{aligned} \tag{A.5}$$

for all $\alpha, \beta = 2, \dots, m$. Hence (A.5) agrees with (A.3) for all $\alpha, \beta = 1, \dots, m$, which implies $\widetilde{\boldsymbol{A}^{\mathsf{H}}} = \widetilde{\boldsymbol{A}}^\top$. □

A.3.3 Proof of 3), 4) and 5)

Proof. By using Lemma 1-5),

$$\begin{aligned}\boldsymbol{A}\boldsymbol{B} &= [\boldsymbol{A}_1, \dots, \boldsymbol{A}_m]\begin{bmatrix}\mathbf{i}_1\boldsymbol{I}_N \\ \vdots \\ \mathbf{i}_m\boldsymbol{I}_N\end{bmatrix}[\mathbf{i}_1\boldsymbol{I}_N, \dots, \mathbf{i}_m\boldsymbol{I}_N]\begin{bmatrix}\boldsymbol{B}_1 \\ \vdots \\ \boldsymbol{B}_m\end{bmatrix} \\ &= [\boldsymbol{A}_1, \dots, \boldsymbol{A}_m]\begin{bmatrix}\mathbf{i}_1\mathbf{i}_1\boldsymbol{I}_N & \cdots & \mathbf{i}_1\mathbf{i}_m\boldsymbol{I}_N \\ \vdots & \ddots & \vdots \\ \mathbf{i}_m\mathbf{i}_1\boldsymbol{I}_N & \cdots & \mathbf{i}_m\mathbf{i}_m\boldsymbol{I}_N\end{bmatrix}\widehat{\boldsymbol{B}} \\ &\overset{1\text{-}5)}{=} [\boldsymbol{A}_1, \dots, \boldsymbol{A}_m]\left\{\sum_{\ell=1}^{m}\begin{bmatrix}\delta_{1,1}^{(\ell)}\mathbf{i}_\ell\boldsymbol{I}_N & \cdots & \delta_{1,m}^{(\ell)}\mathbf{i}_\ell\boldsymbol{I}_N \\ \vdots & \ddots & \vdots \\ \delta_{m,1}^{(\ell)}\mathbf{i}_\ell\boldsymbol{I}_N & \cdots & \delta_{m,m}^{(\ell)}\mathbf{i}_\ell\boldsymbol{I}_N\end{bmatrix}\right\}\widehat{\boldsymbol{B}} \\ &= \sum_{\ell=1}^{m}\left[\sum_{k=1}^{m}\delta_{k,1}^{(\ell)}\boldsymbol{A}_k, \dots, \sum_{k=1}^{m}\delta_{k,m}^{(\ell)}\boldsymbol{A}_k\right]\widehat{\boldsymbol{B}}\mathbf{i}_\ell.\end{aligned}$$

Hence

$$\widehat{\boldsymbol{A}\boldsymbol{B}} = \begin{bmatrix}\sum_{k=1}^{m}\delta_{k,1}^{(1)}\boldsymbol{A}_k & \cdots & \sum_{k=1}^{m}\delta_{k,m}^{(1)}\boldsymbol{A}_k \\ \vdots & \ddots & \vdots \\ \sum_{k=1}^{m}\delta_{k,1}^{(m)}\boldsymbol{A}_k & \cdots & \sum_{k=1}^{m}\delta_{k,m}^{(m)}\boldsymbol{A}_k\end{bmatrix}\widehat{\boldsymbol{B}}.$$

By using the expression (A.4) in the proof of 2), we have

$$
\begin{bmatrix} \sum_{k=1}^{m} \delta_{k,1}^{(1)} \boldsymbol{A}_k & \cdots & \sum_{k=1}^{m} \delta_{k,m}^{(1)} \boldsymbol{A}_k \\ \vdots & \ddots & \vdots \\ \sum_{k=1}^{m} \delta_{k,1}^{(m)} \boldsymbol{A}_k & \cdots & \sum_{k=1}^{m} \delta_{k,m}^{(m)} \boldsymbol{A}_k \end{bmatrix}
$$

$$
= \begin{bmatrix} \delta_{1,1}^{(1)} \boldsymbol{A}_1 - \sum_{k=2}^{m} \delta_{k,1}^{(1)} \boldsymbol{A}_k & \cdots & \delta_{1,1}^{(m)} \boldsymbol{A}_1 - \sum_{k=2}^{m} \delta_{k,1}^{(m)} \boldsymbol{A}_k \\ \vdots & \ddots & \vdots \\ \delta_{1,m}^{(1)} \boldsymbol{A}_1 - \sum_{k=2}^{m} \delta_{k,m}^{(1)} \boldsymbol{A}_k & \cdots & \delta_{1,m}^{(m)} \boldsymbol{A}_1 - \sum_{k=2}^{m} \delta_{k,m}^{(m)} \boldsymbol{A}_k \end{bmatrix}
$$

$$
= \left[\boldsymbol{L}_M^{(1)\top} \widehat{\boldsymbol{A}}, \cdots, \boldsymbol{L}_M^{(m)\top} \widehat{\boldsymbol{A}} \right] = \widetilde{\boldsymbol{A}}. \tag{A.6}
$$

Therefore, $\widehat{\boldsymbol{A}\boldsymbol{B}} = \widetilde{\boldsymbol{A}} \widehat{\boldsymbol{B}}$ holds true.

We also have $\widehat{\boldsymbol{A}\boldsymbol{x}} = \widetilde{\boldsymbol{A}} \widehat{\boldsymbol{x}}$, or $\widehat{\boldsymbol{x}^{\mathsf{H}}\boldsymbol{y}} = \widetilde{\boldsymbol{x}}^{\top} \widehat{\boldsymbol{y}}$ if we set $\boldsymbol{B} = \boldsymbol{x}$, or $\boldsymbol{A} = \boldsymbol{x}^{\mathsf{H}}$ and $\boldsymbol{B} = \boldsymbol{y}$, respectively. □

A.3.4 Proof of 6)

Proof.

$$
\|\boldsymbol{x}\|_{\mathbb{A}_m^N} = \sqrt{\boldsymbol{x}^{\mathsf{H}} \boldsymbol{x}} = \sqrt{\sum_{\ell=1}^{m} \boldsymbol{x}_\ell^{\top} \boldsymbol{x}_\ell} = \sqrt{\widehat{\boldsymbol{x}}^{\top} \widehat{\boldsymbol{x}}} = \|\widehat{\boldsymbol{x}}\|_{\mathbb{R}^{mN}}
$$

□

A.3.5 Proof of 7)

Proof. Obviously,

$$
\boldsymbol{y}^{\star} \in \operatorname*{arg\,min}_{\boldsymbol{y} \in C} \|\boldsymbol{x} - \boldsymbol{y}\|_{\mathbb{A}_m^N} \iff \widehat{\boldsymbol{y}}^{\star} \in \operatorname*{arg\,min}_{\widehat{\boldsymbol{y}} \in \widehat{C}} \|\widehat{\boldsymbol{x}} - \widehat{\boldsymbol{y}}\|_{\mathbb{R}^N}
$$

By the well-known fact [Lue69]: $\operatorname{arg\,min}_{\widehat{\boldsymbol{y}} \in \widehat{C}} \|\widehat{\boldsymbol{x}} - \widehat{\boldsymbol{y}}\|_{\mathbb{R}^N} = \{P_{\widehat{C}}^{\mathbb{R}}(\widehat{\boldsymbol{x}})\} \neq \varnothing$, we have

$$
\operatorname*{arg\,min}_{\boldsymbol{y} \in C} \|\boldsymbol{x} - \boldsymbol{y}\|_{\mathbb{A}_m^N} = \left\{ \widecheck{P_{\widehat{C}}^{\mathbb{R}}(\widehat{\boldsymbol{x}})} \right\} = \{P_C^{\mathbb{A}_m}(\boldsymbol{x})\}.
$$

□

A.3.6 Proof of 8)

Proof. We can easily verify this property in the complex case, so we prove the quaternion case, i.e., the case where $\mathbb{A}_4(=\mathbb{H})$. In this case, from (2.6), $\boldsymbol{L}_M^{(\ell)}$, $\forall \ell \in \{1,2,3,4\}$ are determined as in Example 2. Let $\boldsymbol{A} := \boldsymbol{A}_1\mathbf{i}_1 + \boldsymbol{A}_2\mathbf{i}_2 + \boldsymbol{A}_3\mathbf{i}_3 + \boldsymbol{A}_4\mathbf{i}_4 \in \mathbb{H}^{M\times N}$, $\boldsymbol{A}_\ell \in \mathbb{R}^{M\times N}$, $\forall \ell \in \{1,2,3,4\}$, $\widetilde{\boldsymbol{A}}$ can be expressed as

$$\widetilde{\boldsymbol{A}} = \begin{bmatrix} \boldsymbol{A}_1 & -\boldsymbol{A}_2 & -\boldsymbol{A}_3 & -\boldsymbol{A}_4 \\ \boldsymbol{A}_2 & \boldsymbol{A}_1 & -\boldsymbol{A}_4 & \boldsymbol{A}_3 \\ \boldsymbol{A}_3 & \boldsymbol{A}_4 & \boldsymbol{A}_1 & -\boldsymbol{A}_2 \\ \boldsymbol{A}_4 & -\boldsymbol{A}_3 & \boldsymbol{A}_2 & \boldsymbol{A}_1 \end{bmatrix}.$$

Here, by Theorem 1-3), we have

$$(\widetilde{\boldsymbol{A}\boldsymbol{B}}) = \left[\boldsymbol{L}_M^{(1)\top}\widetilde{\boldsymbol{A}}\widehat{\boldsymbol{B}},\, \boldsymbol{L}_M^{(2)\top}\widetilde{\boldsymbol{A}}\widehat{\boldsymbol{B}},\, \boldsymbol{L}_M^{(3)\top}\widetilde{\boldsymbol{A}}\widehat{\boldsymbol{B}},\, \boldsymbol{L}_M^{(4)\top}\widetilde{\boldsymbol{A}}\widehat{\boldsymbol{B}}\right]. \tag{A.7}$$

On the other hand,

$$\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{B}} = \left[\widetilde{\boldsymbol{A}}\boldsymbol{L}_N^{(1)\top}\widehat{\boldsymbol{B}},\, \widetilde{\boldsymbol{A}}\boldsymbol{L}_N^{(2)\top}\widehat{\boldsymbol{B}},\, \widetilde{\boldsymbol{A}}\boldsymbol{L}_N^{(3)\top}\widehat{\boldsymbol{B}},\, \widetilde{\boldsymbol{A}}\boldsymbol{L}_N^{(4)\top}\widehat{\boldsymbol{B}}\right]. \tag{A.8}$$

By comparing (A.7) and (A.8), Theorem 1-8) is reduced to the following claim:

$$\boldsymbol{L}_M^{(\ell)\top}\widetilde{\boldsymbol{A}} = \widetilde{\boldsymbol{A}}\boldsymbol{L}_N^{(\ell)\top}, \quad \forall \ell \in \{1,2,3,4\}. \tag{A.9}$$

If $\ell = 1$, $\boldsymbol{L}_M^{(1)\top} = \boldsymbol{I}_{mM}$ and $\boldsymbol{L}_N^{(1)\top} = \boldsymbol{I}_{mN}$ from Lemma 2-1), so $\boldsymbol{L}_M^{(1)\top}\widetilde{\boldsymbol{A}} = \widetilde{\boldsymbol{A}}\boldsymbol{L}_N^{(1)\top}$ obviously holds. If $\ell = 2$, we have

$$\begin{aligned} \boldsymbol{L}_M^{(2)\top}\widetilde{\boldsymbol{A}} &= \begin{bmatrix} 0 & -\boldsymbol{I}_M & 0 & 0 \\ \boldsymbol{I}_M & 0 & 0 & 0 \\ 0 & 0 & 0 & \boldsymbol{I}_M \\ 0 & 0 & -\boldsymbol{I}_M & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{A}_1 & -\boldsymbol{A}_2 & -\boldsymbol{A}_3 & -\boldsymbol{A}_4 \\ \boldsymbol{A}_2 & \boldsymbol{A}_1 & -\boldsymbol{A}_4 & \boldsymbol{A}_3 \\ \boldsymbol{A}_3 & \boldsymbol{A}_4 & \boldsymbol{A}_1 & -\boldsymbol{A}_2 \\ \boldsymbol{A}_4 & -\boldsymbol{A}_3 & \boldsymbol{A}_2 & \boldsymbol{A}_1 \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{A}_1 & -\boldsymbol{A}_2 & -\boldsymbol{A}_3 & -\boldsymbol{A}_4 \\ \boldsymbol{A}_2 & \boldsymbol{A}_1 & -\boldsymbol{A}_4 & \boldsymbol{A}_3 \\ \boldsymbol{A}_3 & \boldsymbol{A}_4 & \boldsymbol{A}_1 & -\boldsymbol{A}_2 \\ \boldsymbol{A}_4 & -\boldsymbol{A}_3 & \boldsymbol{A}_2 & \boldsymbol{A}_1 \end{bmatrix} \begin{bmatrix} 0 & -\boldsymbol{I}_N & 0 & 0 \\ \boldsymbol{I}_N & 0 & 0 & 0 \\ 0 & 0 & 0 & \boldsymbol{I}_N \\ 0 & 0 & -\boldsymbol{I}_N & 0 \end{bmatrix} \\ &= \widetilde{\boldsymbol{A}}\boldsymbol{L}_N^{(2)\top}. \end{aligned}$$

In the same way, we can verify $\boldsymbol{L}_M^{(\ell)\top}\widetilde{\boldsymbol{A}} = \widetilde{\boldsymbol{A}}\boldsymbol{L}_N^{(\ell)\top}$ in the case where $\ell = 3$ or 4 (details are lengthy and thus omitted). Therefore, (A.9) holds and thus $(\widetilde{\boldsymbol{A}\boldsymbol{B}}) = \widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{B}}$. □

A.4 Proof of Proposition 2

A.4.1 Proof of (1)

Proof. First, we prove the following:

$$\left\langle \boldsymbol{f}, \boldsymbol{L}^{(\ell)\top} \boldsymbol{f} \right\rangle_{\mathcal{H}^m} = 0 \quad \forall \ell \in \{2, \ldots, m\}, \tag{A.10}$$

where $\boldsymbol{f} := [f_1, \ldots, f_m]^\top \in \mathcal{H}^m$ for $f := f_1 \mathbf{i}_1 + \cdots + f_m \mathbf{i}_m \in \mathcal{H}_{\mathbb{A}_m}$ and $f_1, \ldots, f_m \in \mathcal{H}$.

$$\boldsymbol{L}^{(\ell)\top} \boldsymbol{f} = \begin{bmatrix} \delta_{1,1}^{(\ell)} f_1 - \sum_{q=2}^m \delta_{q,1}^{(\ell)} f_q \\ \delta_{1,2}^{(\ell)} f_1 - \sum_{q=2}^m \delta_{q,2}^{(\ell)} f_q \\ \vdots \\ \delta_{1,m}^{(\ell)} f_1 - \sum_{q=2}^m \delta_{q,m}^{(\ell)} f_q \end{bmatrix} = \begin{bmatrix} -\sum_{q=1}^m \delta_{1,q}^{(\ell)} f_q \\ \sum_{q=1}^m \delta_{2,q}^{(\ell)} f_q \\ \vdots \\ \sum_{q=1}^m \delta_{m,q}^{(\ell)} f_q \end{bmatrix}. \tag{A.11}$$

From Lemma 1-3) and Lemma 1-4), for all $\ell \in \{2, \ldots, m\}$,

$$\begin{aligned}
\left\langle \boldsymbol{f}, \boldsymbol{L}^{(\ell)\top} \boldsymbol{f} \right\rangle_{\mathcal{H}^m} &\overset{\text{(A.11)}}{=} -\sum_{q=1}^m \delta_{1,q}^{(\ell)} \langle f_1, f_q \rangle_{\mathcal{H}} + \sum_{p=2}^m \sum_{q=1}^m \delta_{p,q}^{(\ell)} \langle f_p, f_q \rangle_{\mathcal{H}} \\
&\overset{\text{1-3)}}{=} \underbrace{-\langle f_1, f_\ell \rangle_{\mathcal{H}} + \langle f_\ell, f_1 \rangle_{\mathcal{H}}}_{=0} + \sum_{\substack{p=2 \\ p \neq \ell}}^m \sum_{\substack{q=2 \\ q \neq \ell}}^m \delta_{p,q}^{(\ell)} \langle f_p, f_q \rangle_{\mathcal{H}} \\
&= \sum_{\substack{m \geq p > q \geq 2 \\ p,q \neq \ell}} \delta_{p,q}^{(\ell)} \langle f_p, f_q \rangle_{\mathcal{H}} + \sum_{\substack{2 \leq p < q \leq m \\ p,q \neq \ell}} \delta_{p,q}^{(\ell)} \langle f_p, f_q \rangle_{\mathcal{H}} \\
&\overset{\text{1-4)}}{=} \sum_{\substack{m \geq p > q \geq 2 \\ p,q \neq \ell}} \delta_{p,q}^{(\ell)} \langle f_p, f_q \rangle_{\mathcal{H}} - \sum_{\substack{2 \leq p < q \leq m \\ p,q \neq \ell}} \delta_{q,p}^{(\ell)} \langle f_q, f_p \rangle_{\mathcal{H}} \\
&= \sum_{\substack{m \geq p > q \geq 2 \\ p,q \neq \ell}} \delta_{p,q}^{(\ell)} \langle f_p, f_q \rangle_{\mathcal{H}} - \sum_{\substack{m \geq p > q \geq 2 \\ p,q \neq \ell}} \delta_{p,q}^{(\ell)} \langle f_p, f_q \rangle_{\mathcal{H}} \\
&= 0.
\end{aligned}$$

Therefore, (A.10) holds.

By using (A.10), we have

$$\begin{aligned}
\langle f, f \rangle_{\mathcal{H}_{\mathbb{A}_m}} &= \sum_{\ell=1}^m \left\langle \boldsymbol{f}, \boldsymbol{L}^{(\ell)\top} \boldsymbol{f} \right\rangle_{\mathcal{H}^m} \mathbf{i}_\ell = \langle \boldsymbol{f}, \boldsymbol{f} \rangle_{\mathcal{H}^m} \\
&= \sum_{\ell=1}^m \langle f_\ell, f_\ell \rangle_{\mathcal{H}} = \sum_{\ell=1}^m \|f_\ell\|_{\mathcal{H}}^2 \geq 0
\end{aligned}$$

and $\langle f, f \rangle_{\mathcal{H}_{\mathbb{A}_m}} = 0$ iff $f_1 = \cdots = f_m = 0$, i.e., $f = 0$. □

A.4.2 Proof of (2)

Proof. First, we prove the following:

$$\left\langle \boldsymbol{f}, \boldsymbol{L}^{(\ell)\top} \boldsymbol{g} \right\rangle_{\mathcal{H}^m} = -\left\langle \boldsymbol{g}, \boldsymbol{L}^{(\ell)\top} \boldsymbol{f} \right\rangle_{\mathcal{H}^m} \quad \forall \ell \in \{2, \dots, m\}, \tag{A.12}$$

where $\boldsymbol{g} := [g_1, \dots, g_m]^\top \in \mathcal{H}^m$ for $g := g_1 \mathbf{i}_1 + \dots + g_m \mathbf{i}_m \in \mathcal{H}_{\mathbb{A}_m}$ and $g_1, \dots, g_m \in \mathcal{H}$.

From Lemma 1-1), Lemma 1-4) and (A.11), for all $\ell \in \{2, \dots, m\}$ we have

$$\begin{aligned}
\left\langle \boldsymbol{g}, \boldsymbol{L}^{(\ell)\top} \boldsymbol{f} \right\rangle_{\mathcal{H}^m} &\overset{\text{(A.11)}}{=} -\sum_{q=1}^{m} \delta_{1,q}^{(\ell)} \langle g_1, f_q \rangle_{\mathcal{H}} + \sum_{p=2}^{m} \sum_{q=1}^{m} \delta_{p,q}^{(\ell)} \langle g_p, f_q \rangle_{\mathcal{H}} \\
&\overset{\text{1-1)}}{=} -\sum_{q=1}^{m} \delta_{q,1}^{(\ell)} \langle f_q, g_1 \rangle_{\mathcal{H}} + \sum_{q=1}^{m} \sum_{p=2}^{m} \delta_{p,q}^{(\ell)} \langle f_q, g_p \rangle_{\mathcal{H}} \\
&\overset{\text{1-4)}}{=} -\sum_{q=2}^{m} \delta_{q,1}^{(\ell)} \langle f_q, g_1 \rangle_{\mathcal{H}} - \sum_{q=2}^{m} \sum_{p=2}^{m} \delta_{q,p}^{(\ell)} \langle f_q, g_p \rangle_{\mathcal{H}} \\
&\quad + \sum_{p=2}^{m} \delta_{1,p}^{(\ell)} \langle f_1, g_p \rangle_{\mathcal{H}} \\
&= \sum_{p=2}^{m} \delta_{1,p}^{(\ell)} \langle f_1, g_p \rangle_{\mathcal{H}} - \sum_{q=2}^{m} \sum_{p=1}^{m} \delta_{q,p}^{(\ell)} \langle f_q, g_p \rangle_{\mathcal{H}} \\
&= \sum_{q=2}^{m} \delta_{1,q}^{(\ell)} \langle f_1, g_q \rangle_{\mathcal{H}} - \sum_{p=2}^{m} \sum_{q=1}^{m} \delta_{p,q}^{(\ell)} \langle f_p, g_q \rangle_{\mathcal{H}} \\
&= -\left\langle \boldsymbol{f}, \boldsymbol{L}^{(\ell)\top} \boldsymbol{g} \right\rangle_{\mathcal{H}^m}.
\end{aligned}$$

Therefore, (A.12) holds.

By using (A.12), and note that $\mathcal{H}^m$ is a real Hilbert space, we have

$$\begin{aligned}
\langle f, g \rangle_{\mathcal{H}_{\mathbb{A}_m}} &= \langle \boldsymbol{f}, \boldsymbol{g} \rangle_{\mathcal{H}^m} + \sum_{\ell=2}^{m} \left\langle \boldsymbol{f}, \boldsymbol{L}^{(\ell)\top} \boldsymbol{g} \right\rangle_{\mathcal{H}^m} \mathbf{i}_\ell \\
&= \langle \boldsymbol{g}, \boldsymbol{f} \rangle_{\mathcal{H}^m} - \sum_{\ell=2}^{m} \left\langle \boldsymbol{g}, \boldsymbol{L}^{(\ell)\top} \boldsymbol{f} \right\rangle_{\mathcal{H}^m} \mathbf{i}_\ell \\
&= \langle g, f \rangle_{\mathcal{H}_{\mathbb{A}_m}}^{*}.
\end{aligned}$$

□

A.4.3 Proof of (3)

Proof. Let $\boldsymbol{h} := [h_1, \ldots, h_m]^\top \in \mathcal{H}^m$ for $h := h_1\mathbf{i}_1 + \cdots + h_m\mathbf{i}_m \in \mathcal{H}_{\mathbb{A}_m}$ and $h_1, \ldots, h_m \in \mathcal{H}$.

$$
\begin{aligned}
\langle \alpha f + \beta g, h \rangle_{\mathcal{H}_{\mathbb{A}_m}} &= \sum_{\ell=1}^{m} \left(\alpha \left\langle \boldsymbol{f}, \boldsymbol{L}^{(\ell)\top} \boldsymbol{h} \right\rangle_{\mathcal{H}^m} + \beta \left\langle \boldsymbol{g}, \boldsymbol{L}^{(\ell)\top} \boldsymbol{h} \right\rangle_{\mathcal{H}^m} \right) \mathbf{i}_\ell \\
&= \alpha \langle f, h \rangle_{\mathcal{H}_{\mathbb{A}_m}} + \beta \langle g, h \rangle_{\mathcal{H}_{\mathbb{A}_m}}
\end{aligned}
$$

□

A.4.4 Proof of reproducing property (Eq.(5.2))

Proof. Let $f := f_1\mathbf{i}_1 + \cdots + f_m\mathbf{i}_m \in \mathcal{H}_{\mathbb{A}_m}$ $(f_1, \ldots, f_m \in \mathcal{H})$. Since $\mathcal{H}$ is an RKHS, it holds that

$$
f_\ell(\boldsymbol{x}) = f_\ell(\widehat{\boldsymbol{x}}) = \langle f_\ell, K(\widehat{\cdot}, \widehat{\boldsymbol{x}}) \rangle_{\mathcal{H}} \quad \forall \ell \in \{1, \ldots, m\}.
$$

Therefore, from Lemma 1-2), if we set $g := K(\cdot, \widehat{\boldsymbol{x}}) \in \mathcal{H}$ in (5.1),

$$
\begin{aligned}
\langle f, g \rangle_{\mathcal{H}_{\mathbb{A}_m}} &= \sum_{\ell=1}^{m} \left\langle \boldsymbol{f}, \boldsymbol{L}^{(\ell)\top} \boldsymbol{g} \right\rangle_{\mathcal{H}^m} \mathbf{i}_\ell \\
&= \sum_{\ell=1}^{m} \left\langle \begin{bmatrix} f_1 \\ \vdots \\ f_m \end{bmatrix}, \begin{bmatrix} \delta_{1,1}^{(\ell)} K(\widehat{\cdot}, \widehat{\boldsymbol{x}}) \\ \vdots \\ \delta_{1,m}^{(\ell)} K(\widehat{\cdot}, \widehat{\boldsymbol{x}}) \end{bmatrix} \right\rangle_{\mathcal{H}^m} \mathbf{i}_\ell \\
&\overset{1\text{-}2)}{=} \sum_{\ell=1}^{m} \langle f_\ell, K(\widehat{\cdot}, \widehat{\boldsymbol{x}}) \rangle_{\mathcal{H}} \mathbf{i}_\ell = \sum_{\ell=1}^{m} f_\ell(\boldsymbol{x}) \mathbf{i}_\ell.
\end{aligned}
$$

□

A.5 Complex Adjoint Matrices of Quaternion Matrices

Here, we briefly review the *complex adjoint matrices* of quaternion matrices mentioned in Section 6.1. We also discuss it from the view point of the algebraic translations. The complex adjoint matrix $\chi_{\boldsymbol{A}} \in \mathbb{C}^{2M \times 2N}$ of the quaternion matrix $\boldsymbol{A} = \boldsymbol{A}_r + \boldsymbol{A}_i \jmath \in \mathbb{H}^{M \times N}$ $(\boldsymbol{A}_r, \boldsymbol{A}_i \in \mathbb{C}^{M \times N})$ is defined as

$$
\chi_{\boldsymbol{A}} := \begin{bmatrix} \boldsymbol{A}_r & \boldsymbol{A}_i \\ -\boldsymbol{A}_i^* & \boldsymbol{A}_r^* \end{bmatrix}.
$$

The complex adjoint matrices have been utilized in the study of the numerical range of quaternion matrices [So+94]. It has very similar structure to the nontrivial mapping $\widetilde{(\cdot)}$ of complex matrices, so we also denote

$$\xi_{\boldsymbol{A}} = \begin{bmatrix} \boldsymbol{A}_r \\ -\boldsymbol{A}_i^* \end{bmatrix} \in \mathbb{C}^{2M \times N}$$

as the *trivial* adjoint matrix of $\boldsymbol{A}$ to discuss the complex adjoint matrix in terms of the algebraic translation. Similar to Theorem 1, these two adjoints have the following algebraic properties:

Fact A.5.1 (Algebraic properties of complex adjoint matrices [Lee49; Zha97]). *For all quaternion matrices $\boldsymbol{A}, \boldsymbol{A}' \in \mathbb{H}^{M \times N}$ and $\boldsymbol{B} \in \mathbb{H}^{N \times L}$, the complex adjoint matrices have the following algebraic properties:*

1. $\chi_{\boldsymbol{A}+\boldsymbol{A}'} = \chi_{\boldsymbol{A}} + \chi_{\boldsymbol{A}'}$ *and* $\xi_{\boldsymbol{A}+\boldsymbol{A}'} = \xi_{\boldsymbol{A}} + \xi_{\boldsymbol{A}'}$,
2. $\chi_{\boldsymbol{AB}} = \chi_{\boldsymbol{A}}\chi_{\boldsymbol{B}}$,
3. $\xi_{\boldsymbol{AB}} = \chi_{\boldsymbol{A}}\xi_{\boldsymbol{B}}$ *and* $\xi_{\boldsymbol{Ax}} = \chi_{\boldsymbol{A}}\xi_{\boldsymbol{x}}$ *for all* $\boldsymbol{x} \in \mathbb{H}^N$.

Considering the complex adjoint matrices, we have two ways to translate a quaternion matrix $\boldsymbol{A}$ into a real one. The first way is translating $\boldsymbol{A}$ directly into $\widetilde{\boldsymbol{A}}$ or $\widehat{\boldsymbol{A}}$ with $\widetilde{(\cdot)}$ or $\widehat{(\cdot)}$ for quaternion matrices. The other way is translating $\boldsymbol{A}$ into $\widetilde{\chi_{\boldsymbol{A}}}$ or $\widehat{\xi_{\boldsymbol{A}}}$ with $\widetilde{(\cdot)}$ or $\widehat{(\cdot)}$ for complex matrices via complex adjoints $\chi_{\boldsymbol{A}}$ or $\xi_{\boldsymbol{A}}$. The following lemma clarifies the relation between these two ways.

Lemma A.5.1 (Relations between complex adjoint matrices and the algebraic translations). *For all quaternion matrix $\boldsymbol{A} \in \mathbb{H}^{M \times N}$, it holds that*

$$\widehat{\xi_{\boldsymbol{A}}} = \boldsymbol{T}_M \widehat{\boldsymbol{A}} \text{ and } \widetilde{\chi_{\boldsymbol{A}}} = \boldsymbol{T}_M \widetilde{\boldsymbol{A}} \boldsymbol{T}_N^{-1},$$

where $\boldsymbol{T}_M = \begin{bmatrix} \boldsymbol{I}_M & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{I}_M & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_M & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_M \end{bmatrix} \in \mathbb{R}^{4M \times 4M}$.

Especially if $\boldsymbol{A}$ is a square matrix, $\widetilde{\chi_{\boldsymbol{A}}}$ and $\widetilde{\boldsymbol{A}}$ are similar.

Proof. Let $\boldsymbol{A} := \boldsymbol{A}_1 + \boldsymbol{A}_2\imath + \boldsymbol{A}_3\jmath + \boldsymbol{A}_4\kappa = \boldsymbol{A}_r + \boldsymbol{A}_i\jmath$. We first prove the left equation. If we notice that $\boldsymbol{A}_r = \boldsymbol{A}_1 + \boldsymbol{A}_2\imath$ and $\boldsymbol{A}_i = \boldsymbol{A}_3 + \boldsymbol{A}_4\imath$, we have

$$\boldsymbol{T}_M\widehat{\boldsymbol{A}} = \begin{bmatrix} \boldsymbol{I}_M & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{I}_M & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_M & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_M \end{bmatrix} \begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{A}_2 \\ \boldsymbol{A}_3 \\ \boldsymbol{A}_4 \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1 \\ -\boldsymbol{A}_3 \\ \boldsymbol{A}_2 \\ \boldsymbol{A}_4 \end{bmatrix} = \widehat{\xi}_{\boldsymbol{A}}. \tag{A.13}$$

We next prove the right equation. For all quaternion vector $\boldsymbol{x} \in \mathbb{H}^N$, from Theorem 1-4), Fact A.5.1-3) and (A.13) we have

$$\begin{aligned} \boldsymbol{T}_M\widetilde{\boldsymbol{A}}\widehat{\boldsymbol{x}} &\overset{1\text{-}4)}{=} \boldsymbol{T}_M(\widehat{\boldsymbol{A}\boldsymbol{x}}) \overset{(A.13)}{=} \widehat{\xi}_{\boldsymbol{A}\boldsymbol{x}} \overset{A.5.1\text{-}3)}{=} (\widehat{\chi_{\boldsymbol{A}}\xi_{\boldsymbol{x}}}) \\ &\overset{1\text{-}4)}{=} \widetilde{\chi}_{\boldsymbol{A}}\widehat{\xi}_{\boldsymbol{x}} \overset{(A.13)}{=} \widetilde{\chi}_{\boldsymbol{A}}\boldsymbol{T}_N\widehat{\boldsymbol{x}}. \end{aligned}$$

Hence, $\boldsymbol{T}_M\widetilde{\boldsymbol{A}} = \widetilde{\chi}_{\boldsymbol{A}}\boldsymbol{T}_N$ and thus $\widetilde{\chi}_{\boldsymbol{A}} = \boldsymbol{T}_M\widetilde{\boldsymbol{A}}\boldsymbol{T}_N^{-1}$.

If $\boldsymbol{A}$ is a square matrix, $M = N$ holds, so $\widetilde{\chi}_{\boldsymbol{A}} = \boldsymbol{T}_N\widetilde{\boldsymbol{A}}\boldsymbol{T}_N^{-1}$. It implies $\widetilde{\chi}_{\boldsymbol{A}}$ and $\widetilde{\boldsymbol{A}}$ are similar. □

A.6 Proof of Theorems on Singular Value Decomposition of Quaternion Matrices

The following lemma will be used in the proofs of Theorem 4 and Theorem 5.

Lemma A.6.1. *For a quaternion matrix $\boldsymbol{A} \in \mathbb{H}^{M\times N}$, there always exists a quaternion matrix $\boldsymbol{A}' \in \mathbb{H}^{M\times N}$ such that $\widetilde{\boldsymbol{A}}' = \sigma' \sum_{i=1}^{p} \boldsymbol{u}_i'\boldsymbol{v}_i'^{\top} \in \mathbb{R}^{4M\times 4N}$, where $\boldsymbol{u}_i' \in \mathbb{R}^{4M}, \boldsymbol{v}_i' \in \mathbb{R}^{4N}$ are the singular vectors of $\widetilde{\boldsymbol{A}}$ corresponding to the common singular value σ', and p is the multiplicity of σ'.*

Proof. Let $\boldsymbol{A} = \sum_{i=1}^{r} \sigma_i\boldsymbol{u}_i\boldsymbol{v}_i^{\mathsf{H}}$ be the quaternion singular value decomposition based on [BM04], where r is the rank of $\boldsymbol{A}$, $\sigma_i \geq \cdots \geq \sigma_r$ are the singular values and $\boldsymbol{u}_i \in \mathbb{H}^M$ and $\boldsymbol{v}_i \in \mathbb{H}^N$ $(i = 1, \ldots, r)$ are respectively left and right singular vectors corresponding to σ_i. For simplicity, we suppose that all singular values σ_i are distinct in this proof but we can easily prove the claim even if $\boldsymbol{A}$ has overlapped singular values. In this case, from Theorem 3, $p = 4$ and from Theorem 1-1) and 1-8), we have

$$\widetilde{\boldsymbol{A}} = \sum_{i=1}^{r} \sigma_i\widetilde{\boldsymbol{u}}_i\widetilde{\boldsymbol{v}}_i^{\top} = \sum_{i=1}^{r} \sigma_i \sum_{\ell=1}^{4} \boldsymbol{u}_{i,\ell}\boldsymbol{v}_{i,\ell}^{\top},$$

where $[\boldsymbol{u}_{i,1}, \ldots, \boldsymbol{u}_{i,4}] := \widetilde{\boldsymbol{u}}_i \in \mathbb{R}^{4M\times 4}$ and $[\boldsymbol{v}_{i,1}, \ldots, \boldsymbol{v}_{i,4}] := \widetilde{\boldsymbol{u}}_i \in \mathbb{R}^{4N\times 4}$. If we focus only on a singular value σ_j $(1 \le j \le r)$ such that $\sigma_j = \sigma'$, $\widetilde{\boldsymbol{A}}'$ also can be expressed as

$$\widetilde{\boldsymbol{A}}' = \sigma_j \sum_{\ell=1}^{4} \boldsymbol{u}_{j,\ell} \boldsymbol{v}_{j,\ell}^\top = \sigma_j \widetilde{\boldsymbol{u}}_j \widetilde{\boldsymbol{v}}_j^\top .$$

Hence $\boldsymbol{A}' = \sigma_j \boldsymbol{u}_j \boldsymbol{v}_j^{\mathsf{H}}$ and thus always exists. □

A.6.1 Proof of Theorem 4

Proof. We only prove only for the quaternion case where $m = 4$ but we can easily prove the claim for the complex case in the same way. Let $\boldsymbol{A}_{4p} \in \mathbb{R}^{4M\times 4N}$ be the best $4p$ rank approximation of $\widetilde{\boldsymbol{A}} \in \mathbb{R}^{4M\times 4N}$ for $\boldsymbol{A} \in \mathbb{H}^{M\times N}$ of rank r (i.e., $\operatorname{rank}(\widetilde{\boldsymbol{A}}) = 4r > 4p$), i.e., $\boldsymbol{A}_{4p} = \sum_{i=1}^{4p} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^\top$ for $\widetilde{\boldsymbol{A}} = \sum_{i=1}^{4r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^\top$, where $\boldsymbol{u}_i \in \mathbb{R}^{4M}$ and $\boldsymbol{v}_i \in \mathbb{R}^{4N}$ are respectively left and right singular vectors of $\widetilde{\boldsymbol{A}} \in \mathbb{R}^{4M\times 4N}$ corresponding its ith largest singular value σ_i. From Theorem 3, all singular values at least have multiplicity 4, so we can rewrite

$$\boldsymbol{A}_{4p} = \sum_{q=1}^{p} \sigma'_q \sum_{\ell=1}^{4} \boldsymbol{u}_{4q-\ell+1} \boldsymbol{v}_{4q-\ell+1}^\top,$$

where $\sigma'_q := \sigma_{4q-3} = \sigma_{4q-2} = \sigma_{4q-1} = \sigma_{4q}$ $(q = 1, \ldots, p)$. Moreover, from Theorem 3, σ'_q is identical to the qth largest singular value of $\boldsymbol{A}$. From Lemma A.6.1, we have

$$\boldsymbol{A}_{4p} = \sum_{q=1}^{p} \sigma'_q \sum_{\ell=1}^{4} \boldsymbol{u}_{4q-\ell+1} \boldsymbol{v}_{4q-\ell+1}^\top = \sum_{q=1}^{p} \sigma'_q \widetilde{\boldsymbol{u}}'_q \widetilde{\boldsymbol{v}}'^\top_q ,$$

where $\boldsymbol{u}'_q \in \mathbb{H}^M, \boldsymbol{v}'_q \in \mathbb{H}^N$ are respectively left and right singular vectors of $\boldsymbol{A}$ corresponding to σ'_q. Hence $\boldsymbol{A}_{4p} \in \mathfrak{S}_{\mathbb{H}}(M, N)$ and $\underset{\sim}{\boldsymbol{A}}_{4p} = \sum_{q=1}^{p} \sigma'_q \boldsymbol{u}_q \boldsymbol{v}_q^{\mathsf{H}}$ achieving the best p rank approximation of $\boldsymbol{A}$. □

A.6.2 Proof of Theorem 5

Proof. We prove only the most general case whare $m = 4$. From Theorem 3 and Lemma 4, all singular values at least have multiplicity 4 and $\widetilde{\boldsymbol{A}}$ is of rank $4r$ if we suppose $\boldsymbol{A}$ is of rank r. In this case, $\operatorname{shrink}(\widetilde{\boldsymbol{A}}, \tau)$ can be expressed as

$$\operatorname{shrink}(\widetilde{\boldsymbol{A}}, \tau) = \sum_{q=1}^{r} \max\{\sigma'_q - \tau\} \sum_{\ell=1}^{4} \boldsymbol{u}_{4q-\ell+1} \boldsymbol{v}_{4q-\ell+1}^\top,$$

where $\sigma'_q := \sigma_{4q-3} = \sigma_{4q-2} = \sigma_{4q-1} = \sigma_{4q}$ $(q = 1, \ldots, p)$, $\boldsymbol{u}_i \in \mathbb{R}^{4M}$ and $\boldsymbol{v}_i \in \mathbb{R}^{4N}$ are respectively left and right singular vectors of $\widetilde{\boldsymbol{A}}$ corresponding to its ith largest singular value σ_i $(i = 1, \ldots, 4r)$. From Lemma A.6.1, we have

$$\text{shrink}(\widetilde{\boldsymbol{A}}, \tau) = \sum_{q=1}^{r} \max\{\sigma'_q - \tau\} \widetilde{\boldsymbol{u}}'_q \widetilde{\boldsymbol{v}}'^{\top}_q, \tag{A.14}$$

where $\boldsymbol{u}'_q \in \mathbb{H}^M$ and $\boldsymbol{v}'_q \in \mathbb{H}^N$ are respectively left and right singular vectors of $\boldsymbol{A}$ corresponding to its singular value σ'_q. Similar to the proof of Theorem 4, (A.14) implies that there exists unique quaternion matrix $\boldsymbol{B} = \sum_{q=1}^{r} \max\{\sigma'_q - \tau\} \boldsymbol{u}'_q \boldsymbol{v}'^{\top}_q \in \mathbb{H}^{M \times N}$ such that $\widetilde{\boldsymbol{B}} = \text{shrink}(\widetilde{\boldsymbol{A}}, \tau)$. Hence $\text{shrink}(\widetilde{\boldsymbol{A}}, \tau) \in \mathfrak{S}_{\mathbb{H}}(M, N)$. □

A.7 Tensors and Multidimensional Data

A.7.1 Brief Review of Tensor Basics

Here, we briefly review the basic tensor notations and algebra. A detailed survey of tensor algebra in real domain for applications data science can be found in e.g., [Cic+09; KB09]. For simplicity, here, we focus on tensor in real domain (Note: tensor in hypercomplex domain is defined naturally by extending its entries to hypercomplex number). A *tensor* is a multi-dimensional array and generalizations of matrices. The visualization of 3-order tensor is shown in the rightmost of Fig. A.1. As shown in these figures, a 0-order tensor is a scalar, a 1-order tensor is a vector, 2-order tensor is a matrix, and tensors of order three or higher are called (higher-order) tensors. Three or higher order tensors are customary denoted by calligraphic

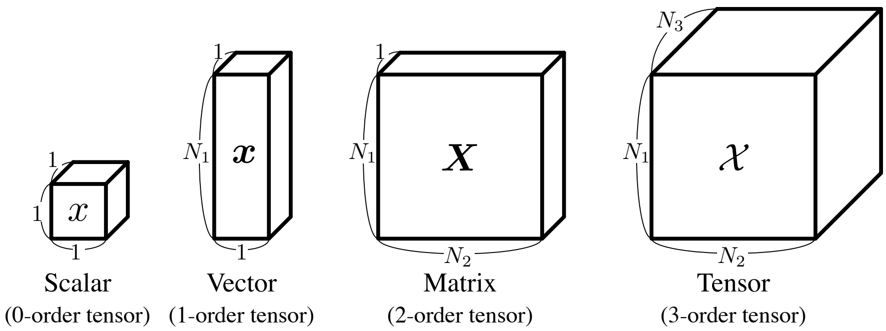


Figure A.1.: Visualization of tensors

letters, e.g., $\mathcal{X}$ as in Fig. A.1. The entry at $(i_1, \ldots, i_n)$ of a tensor $\mathcal{X}$ is denoted by a corresponding lower case letter with the position, e.g., $x_{i_1 \ldots i_n}$.

Fibers are the higher-order analogue of matrix rows and columns. A fiber is defined by fixing every index but one. A matrix column is a mode-1 fiber and a matrix row is a mode-2 fiber. For general n-order tensor $\mathcal{X} \in \mathbb{R}^{N_1 \times \cdots \times N_n}$, the mode-$k$ ($k = 1, \ldots, n$) fibers are all vectors $\boldsymbol{x}_{i_1 \ldots i_{k-1} : i_{k+1} \ldots i_n}$ which are obtained by fixing the value of $\{i_1, \ldots, i_n\} \setminus i_k$. The simplest example of a fiber for 3-order tensor is illustrated in the left of Fig. A.2.

Slices are two-dimensional sections of a tensor, defined by fixing every index but two. For $\mathcal{X}$, the mode-(k, ℓ) ($k, \ell = 1, \ldots, n$) slices are all matrices $\boldsymbol{X}_{i_1 \ldots i_{k-1} : i_{k+1} \ldots i_{\ell-1} : i_{\ell+1} \ldots i_n}$ which are obtained by fixing the value of $\{i_1, \ldots, i_n\} \setminus \{i_k, i_\ell\}$. The simplest example of a slice for 3-order tensor is illustrated in the right of Fig. A.2.

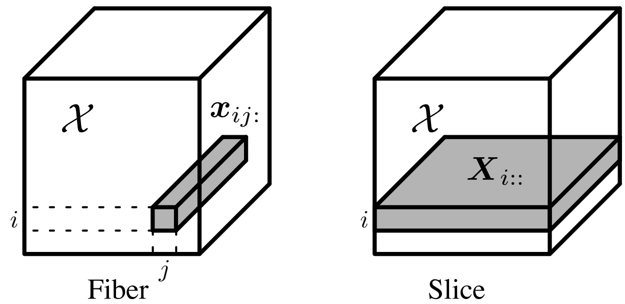


Figure A.2.: Visualization of a fiber and a slice for a 3-order tensor

The *mode-k unfolding* (also called *matricization* or *flattening*) of a tensor $\mathcal{X}$ is denoted by the corresponding bold upper case $\boldsymbol{X}_{(k)} \in \mathbb{R}^{N_k \times I_k}$. It is an $N_k \times I_k$ ($I_k = \prod_{\ell=1, \ell \neq k}^{n} N_\ell$) matrix and obtained by concatenating all mode-k fibers along columns.

The *norm* of a tensor $\mathcal{X}$ is the square root of the sum of squares of all its elements, i.e., $\|\mathcal{X}\| := \sqrt{\sum_{i_1=1}^{N_1} \cdots \sum_{i_n=1}^{N_n} x_{i_1 \ldots i_n}^2}$.

The *inner product* of two same sized tensor $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{N_1 \times \cdots \times N_n}$ is the sum of the products of their same positioned entries, i.e., $\langle \mathcal{X}, \mathcal{Y} \rangle := \sum_{i_1=1}^{N_1} \cdots \sum_{i_n=1}^{N_n} x_{i_1 \ldots i_n} y_{i_1 \ldots i_n}$. It follows immediately that $\|\mathcal{X}\| = \sqrt{\langle \mathcal{X}, \mathcal{X} \rangle}$.

There are several notions of *tensor rank* [KB09] but in this paper we mainly focus on *n-rank* defined as the tuple of the ranks of unfolding for all modes, i.e., $n\text{-rank}(\mathcal{X}) := \left[\text{rank}(\boldsymbol{X}_{(1)}), \ldots, \text{rank}(\boldsymbol{X}_{(n)})\right]$.

A.7.2 Representing Multidimensional Data with Hypercomplex Numbers

Here, we give a brief guide to representing general multidimensional data with C-D numbers.

The most popular case is to represent RGB color images by quaternion matrices as presented in [SE99; BS03; ES07; GA18]. In this case, $M \times N$ color images have 3-dimensional data $(x_{ij}^R, x_{ij}^G, x_{ij}^B) \in \mathbb{R}^3$ in each pixel which can also be regarded as a quaternion with zero real part[1] $x_{ij}^R \imath + x_{ij}^G \jmath + x_{ij}^B \kappa \in \mathbb{H}$ as shown in Fig. A.3. Therefore, $M \times N$ color images can be regarded as $M \times N$ quaternion matrices.

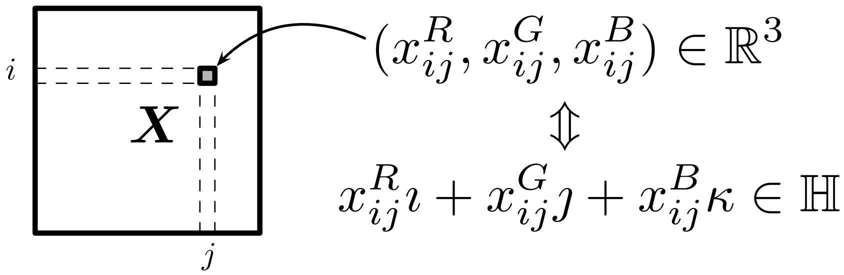


Figure A.3.: Representing color pixels by quaternions

In general case, each data point of multidimensional data is observed as d-dimensional vector. To represent it by a hypercomplex number, the simplest method is the following:

1. Fix the dimension of C-D number $m = 2^n$ such that $2^n \geq d$,
2. Assign all entries of d-dimensional vector to a d of 2^n imaginary parts,
3. Fill all other imaginary parts by zero.

Then, we obtain C-D numbers which represent multidimensional data. Note that basic arithmetic operations are available for them unlikely to multidimensional vectors.

[1] Quaternions whose real part is zero are especially called *pure* quaternions

Bibliography

[Ada+09] T. Adalı, H. Li and R. Aloysius. “On properties of the widely linear MSE filter and its LMS implementation”. In: *Proc. CISS*. 2009 (cit. on p. 19).

[Ada+11] T. Adalı, P. J. Schreier and L. L. Scharf. “Complex-Valued Signal Processing: The Proper Way to Deal With Impropriety”. In: *IEEE Trans. Signal Process* 59.11 (2011), pp. 5101–5125 (cit. on pp. 1, 2, 19, 33).

[AH09] T. Adalı and S. Haykin, eds. *Adaptive Signal Processing: Next Generation Solutions*. Wiley, 2009 (cit. on pp. 2, 19, 25, 34, 43).

[Alf06] D. Alfsmann. “On families of 2^N-dimenisonal hypercomplex algebras suitable for digital signal processing”. In: *Proc. EUSIPCO*. Florence, Italy, 2006 (cit. on pp. 1, 11).

[Aro50] N. Aronszajn. “Theory of reproducing kernels”. In: *Trans. Amer. Math. Soc.* 68.3 (1950), pp. 337–404 (cit. on p. 37).

[Bae01] John C. Baez. “The Octonions”. In: *Bulletin of the American Mathematical Society* 39.2 (2001), pp. 145–205 (cit. on pp. 1, 10–12, 24).

[BB06] N. L. Bihan and S. Buchholz. “Quaternionic independent component analysis using hypercomplex nonlinearities”. In: *Proc. IMA Conf. Math. Signal Process.* Cirenceister, U.K., 2006 (cit. on p. 1).

[Ben+10] J. Benesty, J. Chen and Y. Huang. “A Widely Linear Distortionless Filter for Single-Channel Noise Reduction”. In: *IEEE Signal Process. Lett.* 17.5 (2010), pp. 469–472 (cit. on pp. 1, 19).

[BIG03] A. B.-Israel and T. N. E. Greville. *Generalized Inverses - Theory and Applications*. Springer-Verlag, New York, 2003 (cit. on p. 50).

[BM04] N. L. Bihan and J. Mars. “Singular value decomposition of quaternion matrices: a new tool for vector-sensor signal processing”. In: *Signal Processing* 84.7 (2004), pp. 1177–1199 (cit. on pp. 4, 51, 88).

[Bos94] A. van den Bos. “Complex gradient and Hessian”. In: *Proc. IEE Vision, Image, Signal Process.* 141.6 (1994), pp. 380–383 (cit. on p. 25).

[Bou+12] P. Bouboulis, K. Slavakis and S. Theodoridis. “Adaptive Learning in Complex Reproducing Kernel Hilbert Spaces Employing Wirtinger’s Subgradients”. In: *IEEE Trans. on Neural Net. and Learning Sys.* 23.3 (2012), pp. 425–438 (cit. on pp. 25, 26).

[Boy+11] S. Boyd, N. Parikh, E. Chu, B. Peleato and J. Eckstein. “Distributed optimization and statistical learning via the alternating direction method of multipliers”. In: *Foundations and Trends in Machine Learning* 3.2 (2011), pp. 1–122 (cit. on pp. 57, 63, 70).

[Bra83] D. H. Brandwood. “A complex gradient operator and its application in adaptive array theory”. In: *Proc. Inst. Elect. Eng.* 130 (1983), pp. 11–16 (cit. on p. 25).

[Bre51] J. L. Brenner. “Matrices of quaternions”. In: *Pacific Journal of Mathematics* 1.3 (1951), pp. 329–335 (cit. on pp. 3, 13, 48).

[BS03] N. L. Bihan and S. J. Sangwine. “QUATERNION PRINCIPAL COMPONENT ANALYSIS OF COLOR IMAGES”. In: *Proc. IEEE ICIP*. 2003, pp. 809–812 (cit. on pp. 1, 92).

[BT11] P. Bouboulis and S. Theodoridis. “Extension of Wirtinger’s calculus to reproducing kernel Hilbert spaces and the complex kernel LMS”. In: *IEEE Trans. Signal Process.* 59.3 (2011), pp. 964–978 (cit. on p. 37).

[Can+11] E. J. Candès, X. Li, Y. Ma and J. Wright. “Robust principal component analysis?” In: *Journal of the ACM* 58.3 (2011), pp. 1–37 (cit. on pp. 4, 53, 54).

[CF04] S. B. Choe and J. J. Faraway. “Modeling head and hand orientation during motion using quaternions”. In: *Journal of Aerospace* 113.1 (2004), pp. 186–192 (cit. on p. 1).

[Che10] X. Chen. “Location for the Left Eigenvalues of Quaternionic Matrix”. In: *Journal of Mathematics Research* 2.4 (2010), pp. 141–146 (cit. on p. 13).

[Cho+13] S. Chouvardas, K. Slavakis, S. Theodoridis and I. Yamada. “Stochastic Analysis of Hyperslab-Based Adaptive Projected Subgradient Method Under Bounded Noise”. In: *IEEE Signal Process. Lett.* 20.7 (2013), pp. 729–732 (cit. on p. 32).

[Cho92] J. C. K. Chou. “Quaternion kinematic and dynamic differential equations”. In: *IEEE Trans. Robot. Autom.* 8.1 (1992), pp. 53–64 (cit. on p. 1).

[Cic+09] A. Cichocki, R. Zdunek, A. H. Phan and S. Amari. *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-Way Data Analysis and Blind Source Separation*. Wiley Hoboken, NJ, USA, 2009 (cit. on pp. 6, 90).

[Cic+15] A. Cichocki, D. Mandic, A-H. Phan, C. Caiafa, G. Zhou, Q. Zhao and L. De Lathauwer. “Tensor Decompositions for Signal Processing Applications: From two-way to multiway component analysis”. In: *IEEE Signal Processing Magazine* 32.2 (2015), pp. 145–163 (cit. on p. 6).

[Cic+17] A. Cichocki, A.-H. Phan, Q. Zhao, N. Lee, I. V. Oseledets, M. Sugiyama and D. P. Mandic. “Tensor Networks for Dimensionality Reduction and Large-Scale Optimization: Part 2 Applications and Future Perspectives”. In: *Foundations and Trends® in Machine Learning* 9.6 (2017), pp. 431–673 (cit. on p. 6).

[Cli78] W. K. Clifford. “Applications of Grassmann’s extensive algebra”. In: *Amer. J. Math.* 1.4 (1878), pp. 350–358 (cit. on p. 11).

[Con13] L. Condat. “A primal-dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms”. In: *Journal of Optimization Theory and Applications* 158.2 (2013), pp. 460–479 (cit. on pp. 57, 63, 70).

[CP07] P. L. Combettes and J. C. Pesquet. “A Douglas-Rachford Splitting Approach to Nonsmooth Convex Variational Signal Recovery”. In: *IEEE J. Sel. Topics Signal Process.* 1.4 (2007), pp. 564–574 (cit. on pp. 4–6, 16, 55, 60, 66).

[CY16] T.-S. T. Chan and Y.-H. Yang. “Complex and Quaternionic Principal Component Pursuit and Its Application to Audio Separation”. In: *IEEE Signal Processing Letters* 23.2 (2016), pp. 287–291 (cit. on pp. 4, 54).

[Deu01] F. R. Deutsch. *Best Approximation in Inner Product Spaces*. Springer-Verlag, New York, 2001 (cit. on pp. 56, 61, 68).

[Din08] P. S. R. Diniz. *Adaptive Filtering: Algorithms and Practical Implementation*. 3rd. Berlin: Springer, 2008 (cit. on pp. 2, 32, 34).

[DL14] I. Domanov and L. D. Lathauwer. “Canonical polyadic decomposition of third-order tensors: Reduction to generalized eigenvalue decomposition”. In: *SIAM J. Matrix Anal. & Appl.* 35.2 (2014), pp. 636–660 (cit. on p. 6).

[DM98] T. Dray and C. A. Manogue. “THE OCTONIONIC EIGENVALUE PROBLEM”. In: *Advances in Applied Clifford Algebras* 8.2 (1998), pp. 341–364 (cit. on pp. 1, 13).

[DR56] J. Douglas and H. Rachford. “On the numerical solution of heat conduction problems in two and three space variables”. In: *Trans. of the American Mathematical Society* 82 (1956), pp. 421–439 (cit. on p. 16).

[EB92] J. Eckstein and D. P. Bertsekas. “On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators”. In: *Math. Programming* 55 (1992), pp. 293–318 (cit. on pp. 16, 66).

[Eng+04] Y. Engel, S. Mannor and R. Meir. “Kernel recursive least squares”. In: *IEEE Trans. Signal Process.* 52.8 (2004), pp. 2275–2285 (cit. on p. 37).

[ES07] T. A. Ell and S. J. Sangwine. “Hypercomplex Fourier Transforms of Color Images”. In: *IEEE Trans. Image Process.* 16.1 (2007), pp. 22–35 (cit. on pp. 1, 92).

[Faz+01] M. Fazel, H. Hindi and S. Boyd. “A rank minimization heuristic with application to minimum order system approximation”. In: *Proc. the American Control Conference*. Vol. 6. 2001, pp. 4734–4739 (cit. on p. 51).

[Fra14] W. E. Frazier. “Metal Additive Manufacturing: A Review”. In: *Journal of Materials Engineering and Performance* 23.6 (2014), pp. 1917–1928 (cit. on p. 1).

[GA18] A. M. Grigoryan and S. S. Agaian. *Quaternion and Octonion Color Image Processing with MATLAB*. SPIE Press Monographs, 2018 (cit. on pp. 1, 92).

[Gan+11] S. Gandy, B. Recht and I. Yamada. “Tensor completion and low-n-rank tensor recovery via convex optimization”. In: *Inverse Problems* 27.2 (2011) (cit. on pp. 6, 66, 67, 69, 70).

[GB00] S. L. Gay and J. Benesty. *Acoustic Signal Processing for Telecommunication*. Kluwer Academic Publishers, 2000 (cit. on p. 35).

[Goh+06] S. L. Goh, M. Chen, D. H. Popovi, K. Aihara, D. Obradovic and D. P. Mandic. “Complex-valued forecasting of wind profile”. In: *Renewable Energy* 31 (2006), pp. 1733–1750 (cit. on p. 1).

[GY10] S. Gandy and I. Yamada. “Convex optimization techniques for the efficient recovery of a sparsely corrupted low-rank matrix”. In: *Journal of Math-for-Industry* 2 (2010), pp. 147–156 (cit. on pp. 4, 55, 57, 63).

[Hen88] P. Henrici. *Applied and Computational Complex Analysis Volume I*. Wiley, 1988 (cit. on p. 2).

[HJ12] R. A. Horn and C. R. Johnson. *Matrix Analysis*. 2nd edition. Cambridge Univ. Press, 2012 (cit. on p. 6).

[HJ85] R. A. Horn and C. R. Johnson. *Matrix analysis*. Cambridge University Press, 1985 (cit. on p. 23).

[Hjø11] A. Hjørungnes. *Complex-Valued Matrix Derivatives: With Applications in Signal Processing and Communications*. Cambridge University Press, 2011 (cit. on p. 25).

[Hjø66] A. Hjørungnes. *An Introduction to Nonassociative Algebras*. Academic Press, 1966 (cit. on p. 12).

[HM75] T. Hinamoto and S. Maekawa. “Extended theory of learning identification”. In: *Trans. IEE Japan* 95-C.10 (1975), pp. 227–234 (cit. on p. 32).

[HS01] L. Huang and W. So. “On left eigenvalues of a quaternionic matrix”. In: *Linear Algebra and its Applications* 323 (2001), pp. 105–116 (cit. on pp. 3, 13).

[Hua+12] P.-S. Huang, S. D. Chen, P. Smaragdis and M. Hasegawa-Johnson. “Singing-voice separation from monaural recordings using robust principal component analysis”. In: *Proc. IEEE ICASSP*. 2012, pp. 57–60 (cit. on p. 4).

[Jah+12] C. Jahanchahi, C. C. Took and D. P. Mandic. “On Gradient Calculation in Quaternion Adaptive Filtering”. In: *Proc. IEEE ICASSP*. 2012 (cit. on pp. 2, 33, 34).

[Jah+13] C. Jahanchahi, C. C. Took and D. P. Mandic. “A class of quaternion valued affine projection algorithms”. In: *Signal Processing* 93 (2013), pp. 1712–1723 (cit. on pp. 2, 34).

[KB09] T. Kolda and B. Bader. “Tensor decompositions and applications”. In: *SIAM Review* 51.3 (2009), pp. 455–500 (cit. on pp. 6, 65, 90, 91).

[KD09] K. Kreutz-Delgado. *The complex gradient operator and the* $\mathbb{CR}$*-calculus*. [Online]. Available: http://dsp.ucsd.edu/~kreutz/PEI05.html, Course ECE275A Lecture Supplement on Complex Vector Calculus. 2009 (cit. on pp. 2, 25).

[KS89] I. Kantor and A. Solodovnikov. *Hypercomplex numbers, An Elementary Introduction to Algebras*. Springer-Verlag, New York, 1989 (cit. on pp. 1, 9, 10).

[KW71] G. Kimeldorf and G. Wahba. “Some results on Tchebycheffian spline functions”. In: *Journal of Math. Anal. Appl* 33.1 (1971), pp. 82–95 (cit. on p. 40).

[Lat06] L. D. Lathauwer. “Quaternion ICA From Second-Order Statistics”. In: *SIAM J. Matrix Analysis Applications* 28.3 (2006), pp. 642–666 (cit. on p. 6).

[Lee49] H. C. Lee. “Eigenvalues and Canonical Forms of Matrices with Quaternion Coefficients”. In: *Proc. of the Royal Irish Academy. Section A: Mathematical and Physical Sciences* 52 (1949), pp. 253–260 (cit. on pp. 3, 4, 13, 48, 87).

[Lin+09] Z. Lin, A. Ganesh, J. Wright, L. Wu, M. Chen and Y. Ma. *Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix*. Tech. rep. UIUC Technical Report UILU-ENG-09-2214, 2009 (cit. on pp. 57, 63).

[Liu+08] W. Liu, P. Pokharel and J. C. Principe. “The kernel least-mean-square algorithm”. In: *IEEE Trans. Signal Process.* 56.2 (2008), pp. 543–554 (cit. on pp. 37, 43).

[Lue69] D. G. Luenberger. *Optimization by Vector Space Methods*. Wiley, 1969 (cit. on p. 82).

[Man+11] D. P. Mandic, C. Jahanchahi and C. C. Took. “A quaternion gradient operator and its applications”. In: *IEEE Signal Processing Letters* 18.1 (2011), pp. 47–50 (cit. on pp. 2, 33).

[MB10] G. M. Menanno and N. L. Bihan. “Quaternion polynomial matrix diagonalization for the separation of polarized convolutive mixture”. In: *Signal Processing* 90.7 (2010), pp. 2219–2231 (cit. on p. 1).

[MG09] D. P. Mandic and S. L. Goh. *Complex Valued Nonlinear Adaptive Filters: Noncircularity, Widely Linear and Neural Models*. John Wiley and Sons Ltd, 2009 (cit. on pp. 1, 19, 25, 43).

[Mik+99] S. Mika, G. Rätsch, J. Weston, B. Schölkopf and K.-R. Müller. *Fisher discriminant analysis with kernels*. (in Proc. Advances in Neural Networks for Signal Processing), Y. H. Hu, J. Larsen, E. Wilson, and S. Douglas, Eds. San Mateo, CA: Morgan Kaufmann, 1999, pp. 41–48 (cit. on p. 37).

[Mir+06] S. Miron, N. L. Bihan and J. I. Mars. “Quaternion-MUSIC for Vector-Sensor Array Processing”. In: *IEEE Trans. Signal Process.* 54.4 (2006), pp. 1218–1229 (cit. on p. 1).

[Mor62] J. J. Moreau. “Fonctions convexes duales et points proximaux dans unespace hilbertien”. In: *Comptes Rendus de l’Académie des Sciences (Paris), Série A* 255 (1962), pp. 2897–2899 (cit. on p. 16).

[MY14] T. Mizoguchi and I. Yamada. “An algebraic translation of Cayley-Dickson linear systems and its applications to online learning”. In: *IEEE Trans. Signal Process.* 62.6 (2014), pp. 1438–1453 (cit. on pp. 3, 5).

[MY18a] T. Mizoguchi and I. Yamada. “Hypercomplex Principal Component Pursuit via Convex Optimization”. In: *Proc. APSIPA ASC*. Honolulu, HI, USA, 2018 (cit. on p. 62).

[MY18b] T. Mizoguchi and I. Yamada. “HYPERCOMPLEX TENSOR COMPLETION WITH CAYLEY-DICKSON SINGULAR VALUE DECOMPOSITION”. In: *Proc. IEEE ICASSP*. Calgary, AB, Canada, 2018 (cit. on pp. 4, 63).

[NN11] F. G. A. Neto and V. H. Nascimento. “A NOVEL REDUCED-COMPLEXITY WIDELY LINEAR QLMS ALGORITHM”. In: *Proc. SSP*. 2011 (cit. on p. 3).

[NN12] F. G. A. Neto and V. H. Nascimento. “Second-order analysis of the RC-WL-QLMS algorithm”. In: *Proc. ISWCS*. 2012 (cit. on p. 3).

[NN67] J. Nagumo and J. Noda. “A learning method for system identification”. In: *IEEE Trans. Autom. Control* 12.3 (1967), pp. 282–287 (cit. on pp. 16, 32).

[OU84] K. Ozeki and T. Umeda. “An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties”. In: *IEICE Trans.* 67-A.5 (1984), pp. 126–132 (cit. on pp. 16, 32).

[Pau06] V. I. Paulsen. *An introduction to the theory of reproducing kernel Hilbert spaces*. [Online]. Available: http://www.math.uh.edu/~vern/rkhs.pdf, Lecture Notes on reproducing kernel Hilbert spaces. 2006 (cit. on p. 37).

[PC95] B. Picinbono and P. Chevalier. "Widely linear estimation with complex data". In: *IEEE Trans. Signal Process.* 43.8 (1995), pp. 2030–2033 (cit. on pp. 19, 33).

[Pen+12] Y. Peng, A. Ganesh, J. Wright, W. Xu and Y. Ma. "RASL: Robust Alignment by Sparse and Low-Rank Decomposition for Linearly Correlated Images". In: *IEEE Trans. Pattern Anal. Mach. Intell.* (2012), pp. 2233–2246 (cit. on p. 4).

[Ric+09] C. Richard, J. C. M. Bermudez and P. Honeine. "Online Prediction of Time Series Data With Kernels". In: *IEEE Trans. Signal Process.* 57.3 (2009), pp. 1058–1067 (cit. on pp. 41, 42).

[SB00] N. Sidiropoulos and R. Bro. "On the uniqueness of multilinear decomposition of N-way arrays". In: *J. Chemometrics* 14.3 (2000), pp. 229–239 (cit. on p. 6).

[SB13] S. J. Sangwine and N. Le Bihan. *Quaternion Toolbox for* MATLAB®*, Version 2 with support for Octonions*. [Online]. Available: http://qtfm.sourceforge.net/ (online software library). 2013 (cit. on pp. 24, 34, 44, 71).

[Sch+01] B. Schölkopf, R. Herbrich and A. J. Smola. "A Generalized Representer Theorem". In: *Proc. COLT*. 2001, pp. 416–426 (cit. on p. 40).

[Sch+98] B. Schölkopf, A. J. Smola and K.-R. Müller. "Nonlinear component analysis as a kernel eigenvalue problem". In: *Neural Computation* 10.5 (1998), pp. 1299–1319 (cit. on p. 37).

[SE99] S. J. Sangwine and T. A. Ell. "Hypercomplex auto- and cross-correlation of color images". In: *Proc. IEEE ICIP*. 1999, pp. 319–322 (cit. on pp. 1, 92).

[SM18] D. L. Sun and R. Mazumder. "Non-negative matrix completion for bandwidth extension: A convex optimization approach". In: *Proc. IEEE MLSP*. Southampton, UK, 2018 (cit. on pp. 5, 63).

[So+94] W. So, R. C. Thompson and F. Zhang. "Numerical ranges of matrices with quaternion entries". In: *Linear and Multilinear Algebra* 37 (1994), pp. 175–195 (cit. on p. 87).

[SS01] B. Schölkopf and A. J. Smola. *Learning With Kernels*. MIT press, Cambridge, MA, 2001 (cit. on p. 37).

[SS10] 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 (cit. on p. 25).

[Ste93] G. W. Stewart. "On the Early History of the Singular Value Decomposition". In: *SIAM Review* 35.4 (1993), pp. 551–566 (cit. on p. 50).

[SY13] K. Slavakis and I. Yamada. "The Adaptive Projected Subgradient Method Constrained by Families of Quasi-nonexpansive Mappings and Its Application to Online Learning". In: *SIAM Journal on Optimization* 23.1 (2013), pp. 123–152 (cit. on p. 3).

[The+11] S. Theodoridis, K. Slavakis and I. Yamada. “Adaptive Learning in A World of Projections: A Unifying Framework for Linear and Nonlinear Classification and Regression Tasks”. In: *IEEE Signal Processing Magazine* 28.1 (2011), pp. 97–123 (cit. on p. 3).

[TM10] C. C. Took and D. P. Mandic. “A quaternion widely linear adaptive filter”. In: *IEEE Trans. Signal Process.* 58.8 (2010), pp. 4427–4431 (cit. on p. 19).

[Too+10] C. C. Took, D. P. Mandic and K. Aihara. “Quaternion-valued short term forecasting of wind profile”. In: *Proc. IJCNN*. 2010 (cit. on p. 1).

[Uja+11] B. C. Ujang, C. C. Took and D. P. Mandic. “Quaternion-Valued Nonlinear Adaptive Filtering”. In: *IEEE Trans. on Neural Net.* 22.8 (2011), pp. 1193–1206 (cit. on p. 44).

[V+10] J. Vía, D. Ramírez and I. Santamaría. “Properness and widely linear processing of quaternion random vectors”. In: *IEEE Trans. Inf. Theory* 56.7 (2010), pp. 3502–3515 (cit. on p. 1).

[V+11] J. Vía, D. P. Palomar, D. Ramírez, L. Vielva and I. Santamaría. “Quaternion ICA From Second-Order Statistics”. In: *IEEE Trans. Signal Process.* 59.4 (2011), pp. 1586–1600 (cit. on p. 1).

[Vap95] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 1995 (cit. on p. 37).

[Vũ13] B. C. Vũ. “A splitting algorithm for dual monotone inclusions involving cocoercive operators”. In: *Advances in Computational Mathematics* 38.3 (2013), pp. 667–681 (cit. on pp. 57, 63, 70).

[Wid75] B. Widrow. “The complex LMS algorithm”. In: *Proc. of IEEE* 63.4 (1975) (cit. on pp. 33, 34).

[Wie55] N. A. Wiegmann. “Some theorems on matrices with real quaternion elements”. In: *Canadian Journal of Mathematics* 7 (1955), pp. 191–201 (cit. on p. 1).

[Wir27] W. Wirtinger. “Zur formalen theorie der funktionen von mehr komplexen veranderlichen”. In: *Math. Ann.* 97 (1927), pp. 357–375 (cit. on pp. 2, 25).

[Wol36] L. A. Wolf. “Similarity of matrices in which the elements are real quaternions”. In: *Bulletin of the American Mathematical Society* 42.10 (1936), pp. 737–743 (cit. on p. 1).

[Woo85] R. M. W. Wood. “Quaternionic eigenvalues”. In: *Bulletin of the London Mathematical Society* 17 (1985), pp. 137–138 (cit. on p. 13).

[WR07] D. Wipf and B. Rao. “An empirical Bayesian strategy for solving the simultaneous sparse approximation problem”. In: *IEEE Trans. Signal Process.* 55.7 (2007), pp. 3704–3716 (cit. on p. 54).

[Xia+10] Y. Xia, C. C. Took and D. P. Mandic. “An augmented affine projection algorithm for the filtering of noncircular complex signals”. In: *Signal Processing* 90 (2010), pp. 1788–1799 (cit. on pp. 2, 32).

[Yam+02] I. Yamada, K. Slavakis and K. Yamada. “An Efficient Robust Adaptive Filtering Algorithm Based on Parallel Subgradient Projection Techniques”. In: *IEEE Trans. Signal Process.* 50.5 (2002), pp. 1091–1101 (cit. on pp. 3, 16, 32).

[Yam+11] I. Yamada, M. Yukawa and M. Yamagishi. “Minimizing the Moreau Envelope of Nonsmooth Convex Functions over the Fixed Point Set of Certain Quasi-Nonexpansive Mappings”. In: *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*. Ed. by H. H. Baischke, R. S. Burachik, P. L. Combettes, V. Elser, D. R. Luke and H. Wolkowicz. Springer, 2011. Chap. 17, pp. 345–390 (cit. on p. 16).

[Yam03] I. Yamada. “Adaptive projected subgradient method: a unified view for projection based adaptive algorithms”. In: *The Journal of IEICE (in japanese)* 86.8 (2003), pp. 654–658 (cit. on pp. 3, 15, 30).

[YO04] I. Yamada and N. Ogura. “Adaptive projected subgradient method for asymptotic minimization of sequence of nonnegative convex functions”. In: *Numer. Funct. Anal. Optim.* 25.7-8 (2004), pp. 593–617 (cit. on pp. 3, 15, 30).

[Yua88] J. S. Yuan. “Closed-loop manipulator control using quaternion feedback”. In: *IEEE Trans. Robot. Autom.* 4.4 (1988), pp. 434–440 (cit. on p. 1).

[Yuk+13] M. Yukawa, Y. Sung and G. Lee. “Dual-domain adaptive beamformer under linearly and quadratically constrained minimum variance”. In: *IEEE Trans. Signal Process.* 61.11 (2013), pp. 2874–2886 (cit. on p. 26).

[Zha07] F. Zhang. “Geršgorin type theorems for quaternionic matrices”. In: *Linear Algebra and its Applications* 424 (2007), pp. 139–155 (cit. on p. 13).

[Zha97] F. Zhang. “Quaternions and matrices of quaternions”. In: *Linear Algebra and Its Applications* 251 (1997), pp. 21–57 (cit. on pp. 1, 3, 13, 14, 48–50, 87).

[Zhe01] K.A. Zhevlakov. “Alternative rings and algebras”. In: *Encyclopedia of Mathematics*. Ed. by Hazewinkel and Michiel. Splinger, 2001 (cit. on p. 11).

List of Publications

Articles in Journal Papers

[J1] T. Mizoguchi and I. Yamada. "An algebraic translation of Cayley-Dickson linear systems and its applications to online learning". In: *IEEE Trans. Signal Process.*, 62.6 (2014), pp. 1438–1453.

[J2] T. Mizoguchi and I. Yamada. "Hypercomplex tensor completion via convex optimization". In: *IEEE Trans. Signal Process.*, 67 (2019), pp. 4078–4092.

Peer-Reviewed Articles in International Conferences

[C1] T. Mizoguchi and I. Yamada. "AN ALGEBRAIC REAL TRANSLATION OF HYPERCOMPLEX LINEAR SYSTEMS AND ITS APPLICATIONS TO ADAPTIVE FILTERING". In *Proc. IEEE ICASSP*, Vancouver, BC, Canada, 2013.

[C2] T. Mizoguchi and I. Yamada. "HYPERCOMPLEX TENSOR COMPLETION WITH CAYLEY-DICKSON SINGULAR VALUE DECOMPOSITION". In: *Proc. IEEE ICASSP*, Calgary, AB, Canada, 2018.

[C3] T. Mizoguchi and I. Yamada. "Hypercomplex principal component pursuit via convex optimization". In: *Proc. APSIPA ASC*, Honolulu, HI, USA, 2018.

[C4] T. Mizoguchi and I. Yamada. "HYPERCOMPLEX LOW RANK MATRIX COMPLETION WITH NON-NEGATIVE CONSTRAINTS VIA CONVEX OPTIMIZATION". In: *Proc. IEEE ICASSP*, Brighton, England, UK, 2019.

Domestic Conferences

[D1] T. Mizoguchi, M. Yamagishi and I. Yamada. "Adaptive projected subgradient method for expanded real systems". In: *Proc. IEICE SIP Symposium*, Okinawa, Japan, 2012.

[D2] 溝口毅彦, 山田功. Cayley-dickson 線形システムのための実同型変換と超複素オンライン学習への応用. 第 5 回コンピューテーショナル・インテリジェンス研究会 (計測自動制御学会), 神奈川, 2014.

[D3] T. Mizoguchi and I. Yamada. "A real domain computation for hypercomplex singular value decompositions". In: *Proc. IEICE SIP Symposium*, Morioka, Japan, 2017.

Other Publications

[O1] H. Chen, T. Mizoguchi, T. Yan, K. Zhang and G. Jiang. "A quality control engine for complex physical systems". In: *Proc. IEEE/IFIP DSN*, Rio de Janeiro, Brazil, 2015.

[O2] M. Sato, T. Hashimoto, T. Soma and T. Mizoguchi. "Failure detection of liquid propellant rocket engine using system invariant analysis". In: *Proc. 68th International Astronautical Congress (IAC)*, Adelaide, Australia, 2017.