Structured matrices for predictive control of large and multi-dimensional systems

Sinquin, Baptiste

DOI
10.4233/uuid:d784c51d-1ff0-48e7-b187-6f761491bd11

Publication date
2019

Document Version
Final published version

Citation (APA)
Sinquin, B. (2019). *Structured matrices for predictive control of large and multi-dimensional systems*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:d784c51d-1ff0-48e7-b187-6f761491bd11

Structured matrices for predictive control of large and multi-dimensional systems

With application to adaptive optics

Structured matrices for predictive control of large and multi-dimensional systems

With application to adaptive optics

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus Prof.dr.ir. T.H.J.J. van der Hagen,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op 8 mei 2019 om 12.30 uur

door

Baptiste SINQUIN

Ingénieur diplômé de l'École Centrale de Lyon,
geboren te Quimperlé, France.

Dit proefschrift is goedgekeurd door de promotoren.

Samenstelling promotiecommissie bestaat uit:

Rector Magnificus,	voorzitter
Prof. dr. ir. M. Verhaegen,	Technische Universiteit Delft, promotor
Prof. dr. ir. G. Vdovin,	Technische Universiteit Delft, promotor
Onafhankelijke leden:	
Prof. dr. ir. B. Bamieh,	University of California at Santa Barbara
Prof. dr. ir. L. De Lathauwer,	Katholieke Universiteit Leuven
Prof. dr. ir. A. Hansson,	Linköpings Universitet
Prof. dr. ir. G. J. T. Leus,	Technische Universiteit Delft
Dr. ir. P. Massioni,	Institut National des Sciences Appliquées de Lyon
Reservelid:	
Prof. dr. ir. A. J. van der Veen,	Technische Universiteit Delft

The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013). ERC grant agreement 339681.

Keywords: system identification, low Kronecker rank matrices, structure preserving iterations, predictive control, large-scale adaptive optics.

Printed by: Gildeprint Drukkerijen, NL

Front: Puzzle illustrating the separability assumption used in the thesis. Drawn by Maarten Griffioen.

ISBN 978-94-6323-612-6 An electronic version of this dissertation is available at `http://repository.tudelft.nl/`.

Acknowledgements

I would like to express my deep gratitude toward some people with whom I have shared this journey, for helping me, for influencing my thoughts, and for questioning my derivations.

I am especially grateful to Michel for his guidance and the thorough feedback on my notes, which has significantly increased the quality of this dissertation. Thank you for giving me the chance to come to Delft, and for repeating again and again that the best way to open new doors is to start with simple things. I would also like to thank the team members. Many thanks to Reinier, Pieter and Chengpu for the discussions and the crazy expertise you brought in every meeting. I have been impressed by the ideas conveyed in these meetings, and keeping up the pace was both a challenge and a great opportunity. Mario, Guido, Peter, Amaury, and Daniel, your questions, remarks and doubts have fueled my thoughts and your contribution to this thesis is important. I hope you could learn as much from me as I did from you. Maarten and Will played a key role in setting up and automating the laboratory testbed, starting from scratch and in spite of the countless hardware issues. Thank you! I wish we could have collaborated longer to further develop the bench.

The research environment was both stimulating and of great support. Thank you Laurens for the peaceful atmosphere and interesting discussions in the office. Elisabeth, thank you for staying the course and for spicing up life at work! Thank you to the microscopy mafia, and primarily to Paolo, for all the help in the lab. Along with Dean, Gleb, Hai, and Tope, you have opened my eyes on a different view of an adaptive optics system, far away from maths. I would like to thank Oleg, Sander, Tamás and Thao for keeping the door open and for always being eager to think with me on basically any question. Thank you to the colleagues, Barbara, Hans, Iurii, Jacopo, Pascal, Peter, Raf, Sachin, Shrinivas, and Tijmen. Dank je Sjoerd, voor de lunchpauzes in Delft en voor helpen met de taal! Many thanks to Erica for guiding me through the TU paperworks, and to Heleen, Kiran, Kitty and Marieke for the administrative support. This help has been a great relief!

It was a real pleasure assisting for the course on filtering and system identification. Thanks to Edwin who showed me the way, to Pieter for collaborating in improving (and correcting) the exercises and above all, for all the fun times we have had, and to the students for the enthusiasm and also for the Friday-evenings football games. The many interactions we have had made the working atmosphere very lively.

Questions have always been extremely worthy and have helped me to deepen my understanding and broaden the scope of my work. I acknowledge the help of Prof. Anders Hansson who asked questions for which I have so far needed a minimum of a month to provide an answer. Many thanks to Prof. Ivan Markovsky, Mariya Ishteva and Philippe Dreesen for their always renewed interest in this work, and to Prof. Caroline Kulcsár for the constructive comments. I am grateful to the committee

members for reading this thesis and for travelling to Delft for participating in the defence.

Ardalan, thank you for the one-of-a-kind violin sessions, for transmitting so much of your passion (and OCD), and for your high standards and infinite patience. High five to the football and ice-skating teammates from ELSkalatie and the friends in Lyon and Quimperlé for the good times away from work!

For its unwavering support, for having shared with me the ups and downs, my family played the most important role. Merci! Mille fois.

Baptiste Sinquin
Delft, January 2019.

Summary

For controlling systems and more particularly rejecting a disturbance, a classical approach consists of designing an observer either from data or solving a Riccati equation. Although system identification has developed since the sixties and is nowadays a well-established area, identifying from data the spatial and temporal dynamics for large-scale systems with thousands of inputs and outputs remains challenging from the computational point of view. There is a similar curse of dimensionality when solving the discrete algebraic Riccati equation. In order to reduce the memory storage and the computational requirements, prior knowledge on how the sensors are spatially distributed is commonly translated into structural assumptions on the system matrices. When the sensors are regularly spread on a two-dimensional grid, and the underlying function that describes the spatial dynamics is separable in its horizontal and vertical coordinates, a particular matrix representation is studied. This assumption differs from the spatial invariance, Bamieh et al. (2002).

Adaptive Optics (AO) is one example of an application. AO is a control methodology that allows high-resolution imaging of an object emitting little light through an heterogeneous and time-varying medium. The algorithms are implemented in ground-based telescopes to cope with the distorted phase of the light emitted from a reference star and having travelled through a turbulent atmosphere. A deformable mirror is used to reshape the incoming light and reject the atmosphere-induced disturbances flowing over the telescope aperture. The performance of an AO system is improved when the spatial-temporal correlations of the turbulence are used to derive a prediction at the next time instant, thereby reducing the temporal error. A next generation of large telescopes featuring 10^4 actuators and sensors demands scalable predictive algorithms, both for deriving a Kalman filter and for online computations.

We propose in this thesis a dense though data-sparse representation of the matrices for linear time-invariant and so-called multi-dimensional systems, yielding scalable algorithms for identification and paving the way for solving the DARE. The system matrices are parametrized as a sum of Kronecker products of factor matrices. When the number of summands r is small compared to the size of the actuator/sensor array, the matrix belong to the class of low-Kronecker rank matrices. The parameter r allows to make a trade-off between between the accuracy of the representation maximized in an unstructured setting and the scalability of the developed algorithms. This structure is a fortiori competitive with respect to the sparse multi-banded structure when the matrix to be approximated is dense. Such a parametrization is multi-linear, does not require sparsity in the entries and its storage scales linearly with the number of nodes in the array instead of quadratically.

The first contribution of this thesis deals with the identification of large-scale Vector Auto-Regressive models, [**Chapter 2**]. For an array of size $N \times N$, the sensor

data at each time instant is reshuffled into a matrix rather than a vector such that we formulate a bilinear least-squares with rN^2 variables instead of N^4. Regularization to enforce temporal stability or a decay in the factor matrices was incorporated without altering the convergence to the global minimum of the Alternating Least Squares. The computational complexity reduces from $\mathcal{O}(N^6)$ to $\mathcal{O}(N^3 N_t)$, where N_t is the number of time samples used in the identification.

Second, the identification of state-space models is investigated, [**Chapter 3**]. When the state-space matrices are written with a single Kronecker product, a class of matrix state-space models is introduced and a subspace-like algorithm is proposed. The latter consists of three steps, two of which were shown to converge to the global minimum (as observed empirically). Although its computational performances allowed to handle much larger dimensions than the standard algorithms, it nonetheless implies a decrease in accuracy due to the non-globally convergent block-coordinate algorithm used to minimize the rank of a block-Hankel matrix subject to bilinear constraints.

For all standard linear algebra operations, assuming a decomposition of the factor matrices with a single Kronecker product of two terms implies a lower-bound on the achievable computational complexity, equal to $\mathcal{O}(N^3)$ for an array of size $N \times N$. A linear computational complexity with respect to the number of nodes as e.g would be obtained when the nodes are decoupled can not be reached this way, and a parametrization with a product of more Kronecker products than only two was studied. Its close relationship with tensors allowed to derive more efficient algorithms reaching asymptotically with the tensor order $\mathcal{O}(N^2)$ complexity, [**Chapter 4**]. The first tensor orders provide already with most of the computational improvements without losing much accuracy as demonstrated in laboratory experiments dedicated to large-scale AO described in [**Chapter 6**].

In some applications such as AO where the state has a physical meaning and can be estimated - the wavefront-, it is common to derive the system matrices using first principles, i.e without resorting to subspace identification. When these can be decomposed as low-Kronecker rank matrices, natural questions are first, whether the solution of the discrete algebraic Riccati equation can itself be written (or approximated) as a sum of a few Kronecker terms, and second, whether it can be solved efficiently using structure-preserving iterations. As a first step toward answering these two questions, we solve the Kronecker-structured discrete Lyapunov equation with $\mathcal{O}(N^3)$ complexity instead of $\mathcal{O}(N^6)$, [**Chapter 5**].

In addition to the fundamental new contributions, a validation study was proposed based on an optical breadboard in the Smart Optics Lab of TU Delft, [**Chapter 6**]. The use of tensor autoregressive models for modeling the spatial dynamics of open-loop turbulence data and its applicability to closed-loop operation for large-scale AO systems was demonstrated. Especially, we have shown that in spite of losing performance because of structuring the coefficient matrices, it reduces significantly the temporal error for large Greenwood per sample frequency ratio compared to the non-predictive methods.

This PhD thesis draws pros and cons of a multi-linear parametrization of large matrices of LTI systems, especially from an identification perspective. Besides, its

close connection with tensors raised new fundamental questions in the analysis of such structured systems.

Samenvatting

Voor het regelen van systemen en meer specifiek het afwijzen van een stochastische verstoring bestaat een klassieke benadering uit het ontwerpen van een observer, hetzij uit data, hetzij uit het oplossen van een Riccati vergelijking. Hoewel systeemidentificatie sinds de jaren zestig onderzocht wordt en tegenwoordig een goed ontwikkeld gebied is, blijft het vanuit rekenkundig oogpunt uitdagend de ruimtelijke en tijdelijke dynamica van grootschalige systemen met duizenden ingangen en uitgangen te identificeren. Er is een soortgelijke vloek van dimensionaliteit bij het oplossen van de discrete algebraïsche Riccati vergelijking. Teneinde de geheugenopslag en de rekenvereisten te verminderen wordt voorkennis over hoe de actuatoren en sensoren gekoppeld worden gewoonlijk in structurele aannamen op de systeemmatrices vertaald.

Toepassingen van tweedimensionale arrays van actuatoren en sensoren omvatten Adaptive Optics (AO) voor extreem grote telescopen. AO is een besturingsmethode die beeldvorming met hoge resolutie mogelijk maakt van een voorwerp dat weinig licht door een heterogeen en in de tijd variërend medium emitteert. De algoritmen worden geïmplementeerd in telescopen op de grond om de vervormde fase van het licht, dat door een referentie ster uitgestraald wordt, te verwerken en door een turbulente atmosfeer hebben gereisd. Een vervormbare spiegel word gebruikt om het invallende licht opnieuw vorm te geven en door de atmosfeer veroorzaakte verstoringen die over de telescoop opening lopen te verwerpen. De prestaties van een AO systeem worden verbeterd wanneer de ruimtelijk-temporele correlaties van de turbulentie worden gebruikt om een voorspelling af te leiden voor de volgende tijdstap, waardoor de temporele fout wordt verminderd. Deze volgende generatie grote telescopen met 10^4 actuatoren en sensoren werkt op kilohertz snelheden en vereist schaalbare voorspellende algoritmen.

In dit proefschrift, stellen we een dichte maar gegevens-schaarse representatie voor van de matrices voor lineaire tijd invariante en meerdimensionale systemen die schaalbare identificatiealgoritmen opleveren. De systeem matrices worden geparametriseerd als een som van Kronecker producten van factor matrices. Wanneer het aantal summands r klein is in vergelijking met de grootte van de actuatoren/sensoren array, dan behoort de matrix tot de klasse van laag Kronecker rang matrixen. De parameter r maakt het mogelijk om een afweging te maken tussen de nauwkeurigheid van de vertegenwoordiging gemaximaliseerd in een ongestructureerde instelling en de schaalbarheid van de ontwikkelde algoritmen. Deze structuur is a fortiori concurrerend met betrekking tot de schaars multi-bandstructuur wanneer de te benaderen matrix dicht is. Een dergelijke parametrisatie is multi-lineair, vereist geen sparsiteit in de ingangen, en is opslagschalen lineair met het aantal knooppunten in de reeks in plaats van kwadratisch.

De eerste bijdrage van dit proefschrift handelt over de identificatie van groot-

schalige Vector Auto-Regressieve modellen, [**Hoofdstuk 2**]. Voor een array van $N \times N$ worden de sensorgegevens op elk tijdstip in een matrix herschikt in plaats van een vector die een bilineaire least squares oplevert met rN^2 variabelen in plaats van N^4. Regularisatie om temporele stabiliteit of een verval in de factor matrices af te dwingen werd opgenomen zonder de convergentie naar het globale minimum van de afwisselende least squares te veranderen. De rekenkundige complexiteit neemt af van $\mathcal{O}(N^6)$ naar $\mathcal{O}(N^3 N_t)$, waarbij N_t het aantal tijdsamples is dat in de identificatie gebruikt wordt.

Ten tweede wordt de identificatie van toestands modellen onderzocht, [**Hoofdstuk 3**]. Wanneer de toestand matrices met een enkel Kronecker-product geschreven worden, dan wordt een klasse van matrix toestand modellen geïntroduceerd en wordt een subspace-achtig algoritme voorgesteld. Dit laatste bestaat uit drie stappen, waarvan er wordt getoond dat twee convergeren naar het globale minimum (zoals empirisch waargenomen wordt). Hoewel de rekenkost veel grotere dimensies kunnen verwerken dan de standaardalgoritmen, impliceert dit niettemin een afname in nauwkeurigheid vanwege het niet-globaal convergente blokcoördinatenalgoritme dat gebruikt wordt om de rank van een blok-Hankel-matrix die onderhevig aan bilineaire beperkingen is te minimaliseren.

Voor alle standaard lineaire algebra-bewerkingen impliceert een decompositie van de factor-matrices met een enkel Kronecker-product van twee termen een ondergrens voor de bereikbare rekenkundige complexiteit, gelijk aan $\mathcal{O}(N^3)$ voor een array van maat $N \times N$. Een lineaire computationele complexiteit met betrekking tot het aantal knooppunten zoals b.v. zou worden verkregen wanneer de knooppunten worden ontkoppeld, kan niet op deze manier worden bereikt, en een parametrisering met een product van meer Kronecker-producten dan slechts twee werd bestudeerd. Door de nauwe relatie met tensoren konden efficiëntere algoritmen asymptotisch worden afgeleid met de complexiteit van de tensororder $\mathcal{O}(N^2)$, [**Hoofdstuk 4**]. De eerste tensororders bieden al de meeste reken technische verbeteringen zonder veel nauwkeurigheid te verliezen, zoals aangetoond in laboratoriumexperimenten die gewijd zijn aan grootschalige AO beschreven in [**Hoofdstuk 6**].

In sommige toepassingen zoals AO, waar de toestand een fysieke betekenis heeft en kan worden geschat - het wavefront -, is het gebruikelijk om de systeemmatrices af te leiden met behulp van de first principles, d.w.z. zonder toevlucht te nemen tot subspace identificatie. Wanneer deze als laag-Kronecker rang matrices kunnen worden ontbonden, is natuurlijk de eerste vraag of de oplossing van de discrete algebraïsche Riccati-vergelijking zelf kan worden geschreven (of benaderd) als een som van een paar Kronecker-termen, en ten tweede of het efficiënt opgelost kan worden met behulp van structuurbehoudende iteraties. Als een eerste stap naar het beantwoorden van deze twee vragen lossen we de Kronecker-gestructureerde discrete Lyapunov-vergelijking op met $\mathcal{O}(N^3)$ complexiteit in plaats van $\mathcal{O}(N^6)$, [**Hoofdstuk 5**]. Het eerste algoritme dat voorgesteld wordt, is afhankelijk van de Smith's iteratie, terwijl de tweede gebruik maakt van Alternate Direction Implicit methode, waarbij elk van deze laatste een Sylvester vergelijking van veel kleinere omvang oplost.

Naast de fundamentele nieuwe bijdragen, werd een validatiestudie voorge-

steld op basis van een optische breadboard in het Smart Optics Lab van de TU Delft, [**Hoofdstuck 6**]. Het gebruik van tensor autoregressieve modellen voor het modelleren van de ruimtelijke dynamiek van open-loop turbulentiegegevens en de toepasbaarheid ervan in gesloten-lus bedrijf voor grootschalige AO-systemen werd gedemonstreerd. In het bijzonder hebben we aangetoond dat, ondanks het verliezen van prestaties door het structureren van de coëfficiëntmatrices, de temporele fout aanzienlijk wordt gereduceerd voor grote Greenwood per samplefrequentie verhouding vergeleken met de niet-voorspellende methoden.

Dit proefschrift beschrijft de voor- en nadelen van een multilineaire parametrisatie van grote matrices van LTI-systemen, vooral vanuit een identificatieperspectief. Bovendien leverde de nauwe band met tensoren nieuwe fundamentele vragen op bij de analyse van dergelijke gestructureerde systemen.

Contents

Acronyms

ADI	Alternate Direction Implicit
ADMM	Alternating Direction Method of Multipliers
AIC	Akaike Information Criteria
ALS	Alternating Least Squares
AO	Adaptive Optics
BCU	Block-Coordinate Update
CCD	Charged-Coupled Device
CPD	Canonical Polyadic Decomposition
CPU	Central Processing Unit
DARE	Discrete Algebraic Riccati Equation
DM	Deformable Mirror
EE	Encircled Energy
ELT	Extremely Large Telescope
FIR	Finite-Impulse Response
GPU	Graphical Processing Units
IEEE	Institute of Electrical and Electronics Engineers
K4SID	Kronecker-Structured large-Scale SubSpace IDentification
LQG	Linear Quadratic Gaussian
LTI	Linear Time Invariant
MLDS	Multi Linear Dynamical System
MOESP	Multivariable Output Error State sPace
MSSM	Matrix State Space Model
MVM	Matrix Vector Multiplication
N4SID	Numerical algorithms for Subspace State Space System IDentification
PBSID	Predictor Based Subspace IDentification
PSF	Point Spread Function
QUARKS	Kronecker-based Vector AutoRegressive with eXogenous inputs (KVARX)
RMSE	Root Mean Square Error
SCAO	Single Conjugate Adaptive Optics
SH	Shack-Hartmann
SNR	Signal to Noise Ratio
SOK	Sums Of Kronecker
SSARX	SubSpace identification method that uses an ARX estimation
SSS	Sequentially Semi-Separable
SVD	Singular Value Decomposition
TSSM	Tensor State-Space Model
VAF	Variance Accounted For
VARX	Vector AutoRegressive with eXogeneous inputs
WFS	Wavefront sensor

Notations

The notations commonly used throughout the dissertation are introduced here. Other chapter-specific notations are introduced in the respective chapter.

The set of real number is denoted with $\mathbb{R}$. The set of positive integers is denoted with $\mathbb{N}$. For a set Ω, $\text{card}(\Omega)$ denotes the number of elements in Ω.

Scalars are denoted by lower or uppercase letters or symbols. The floor of the real number x denoted with $\lfloor x \rfloor$ and the remainder after division of x by y with $mod(x, y)$.

Vectors are written as boldface lower-case letters such as $\mathbf{x}$. The boldface is used to make a distinction between indexing a set of vectors, such as $\mathbf{x}_1, \mathbf{x}_2$, and referring to the elements of a single vector $\mathbf{x} \in \mathbb{R}^n$, such as $x_1, \ldots, x_n$. The null vector and the vector of ones are denoted by $\mathbf{0}$ and $\mathbf{1}$ respectively, where an index can be used to explicitly show the size e.g $\mathbf{1}_n \in \mathbb{R}^n$. The Euclidean norm of a vector $\mathbf{x}$ is written as $\|\mathbf{x}\|_2 = \sqrt{x_1^2 + \ldots + x_n^2} = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}$. The sum in absolute value for the elements in $\mathbf{x} \in \mathbb{R}^n$ is denoted with $\|\mathbf{x}\|_1 = \sum_{i=1}^n |x_i|$. The covariance for two zero-mean vectors $\mathbf{x}$ and $\mathbf{y}$ is written as $\mathbb{E}\left[\mathbf{x}\mathbf{y}^T\right]$.

Matrices are represented by boldface uppercase letters such has $\mathbf{X}$. The element located at the i-th row and j-th column of the matrix $\mathbf{X}$ is written as $x_{i,j}$, or $x_{\star_{i,j}}$ when the matrix has a subscript $\star$. MATLAB-like notations are used to denote columns and rows of matrices, e.g $\mathbf{X}(:, \mathtt{i})$ refers to the i-th column of $\mathbf{X}$, $\mathbf{X}(\mathtt{i}, :)$ the i-th row. The matrix $\mathbf{X}(\mathtt{a:i:b, c:j:d})$ selects a submatrix from $\mathbf{X}$ composed of all entries with row index $a + ki$ (until it reaches b) and column index $c + kj$ (until d) where k is an integer starting at 0. The trace, transpose and inverse (if it exists) of $\mathbf{X}$ are written respectively with $\text{Trace}(\mathbf{X}), \mathbf{X}^T$ and $\mathbf{X}^{-1}$. The Frobenius norm for a matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$ is denoted with $\|\mathbf{X}\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n x_{i,j}^2} = \sqrt{\text{Trace}(\mathbf{X}^T\mathbf{X})}$. The nuclear norm for a matrix $\mathbf{X}$ is equal to the sum of its singular values and is written with $\|\mathbf{X}\|_\star$. The vectorization operator applied on $\mathbf{X}$ is written with $\text{vec}(\mathbf{X}) = \begin{bmatrix} x_{1,1} & x_{2,1} & \ldots & x_{m,n} \end{bmatrix}^T$. The Kronecker product between two matrices $\mathbf{X}$ and $\mathbf{Y}$ is denoted by $\mathbf{X} \otimes \mathbf{Y}$. The Khatri-Rao product is the column-wise Kronecker product and is denoted with $\odot$. The outer product between two vectors $\mathbf{x}, \mathbf{y}$ of length N is a matrix $\mathbf{x} \circ \mathbf{y}$ of size $N \times N$ with the (i, j)-th element equal to $x_i y_j$.

Tensors are denoted with calligraphic letters. For a tensor $\mathcal{X} \in \mathbb{R}^{J_1 \times \ldots \times J_d}$, the entry at position $j_1, \ldots, j_d$ is denoted with $x_{j_1,\ldots,j_d}$. Matlab-like notations are used to denote parts of a tensor. For example, for a tensor of size $J_1 \times J_2 \times J_3$, $\text{vec}(\mathcal{X}_{i,:,j})$ is equal to $\begin{bmatrix} x_{i,1,j} & \ldots & x_{i,J_2,j} \end{bmatrix}$. A tensor $\mathcal{X}$ is vectorized using the vec operator such that $\text{vec}(\mathcal{X}) = \begin{bmatrix} x_{1,\ldots,1} & x_{2,1,\ldots,} & \ldots & x_{1,2,1,\ldots} & \ldots & x_{J_1,\ldots,J_d} \end{bmatrix}$.

The big-$\mathcal{O}$ notation is used for describing computational complexities and indicates the asymptotic growth rate of the computational cost for a given mathematical operation. For example, an operation costing $\mathcal{O}(n)$ floating-point operations (flops) finishes in at most $c \cdot n$ flops, for some constant c.

1
Introduction

For controlling systems and more particularly rejecting a disturbance, a classical approach consists of designing an observer either from data or solving a Riccati equation. Although system identification has developed since the sixties and is nowadays a well-established area, identifying from data the spatial and temporal dynamics for large-scale systems with thousands of inputs and outputs remains challenging from the computational point of view. There is a similar curse of dimensionality when solving the discrete algebraic Riccati equation. In order to reduce the memory storage and the computational requirements, prior knowledge on how the sensors are spatially distributed and how they communicate with each other is commonly translated into structural assumptions on the system matrices. When the sensors and actuators are regularly distributed on a regular multi-dimensional grid, a particular matrix parametrization may be exploited. The main objective of the thesis is to propose a dense though data-sparse representation of the system matrices in the particular case of a distributed sensing array, and derive scalable algorithms to design an observer.

Applications of large-scale stochastic systems with a multi-dimensional grid of sensors include Adaptive Optics (AO) for extremely large telescopes. AO is a control methodology that allows high-resolution imaging of an object emitting little light through an heterogeneous and time-varying medium. The algorithms are implemented in ground-based telescopes to cope with the distorted phase of the light emitted from a reference star and having travelled through a turbulent atmosphere. A deformable mirror is used to reshape the incoming light and reject the atmosphere-induced disturbances flowing over the telescope aperture. The performance of an AO system is improved when the spatial-temporal correlations of the turbulence are used to derive a prediction at the next time instant, thereby reducing the temporal error.

In this chapter, the dissertation title is analysed and situated in the context of existing literature. System identification is used as a starting point to propose a state-of-the-art for the matrix parametrizations introduced in the control community. The research direction and main contributions are highlighted and the outline of the remaining chapters is presented last.

1

1.1. Large and spatially distributed systems

1.1.1. System identification

Signals, systems and models

A system is an object in which variables of different kind interact and produce observable signals, Ljung (1999). Engineering systems are often equipped with sensors and actuators. The signals measured by the sensors are called outputs, and the signals sent to the actuators are the inputs. Stochastic disturbances may enter the system and their influence be measured on the output only. In adaptive optics, sensor noise is one example of endogenous disturbance while exogenous disturbances are due to the atmospheric turbulence. Both deteriorate the system performance. The system is controlled by applying input commands to the actuators based on the sensor outputs in order to achieve a performance criteria while ensuring that the closed-loop system is stable. For example, the input may be used to stir some user-defined variable toward a desired value, or to minimize the influence of the disturbance in a closed-loop setting. The disturbance evolves while processing the measurements to compute the control inputs, and without prediction, the correction is often outdated. Such temporal error is intrinsic to the control loop. A mathematical model is used to relate the temporal evolutions of the disturbance, the input and output signals, and allows the prediction of the future value of the disturbance in order to apply an updated correction at each time instant. Controlling a system such that it meets the user requirements strongly relies on how accurate the model represents the reality.

The mathematical relation between the inputs and the outputs may be known up to a certain extent using laws from physics such as the conservation of energy and mass, Newton's laws of movement or Partial Differential Equations (PDE). For example, the Euler-Bernoulli beam equation describes the spatial-temporal behaviour of a flexible beam subject to some external excitation. One of the shortcomings of this approach is that some coefficients in the PDE are unknown and estimating them may be either too complex or it may not represent accurately the dynamics due to inhomogeneities in the material or measurement errors. In adaptive optics, prior knowledge on the disturbance flowing over the telescope aperture is either very rough or modelled with non-linear equations. A simplified model consists of assuming that the flow is frozen and propagates at constant windspeed in a known direction whereas the Navier-Stokes equations are highly non-linear and difficult to solve. It illustrates the need for alternative methods to model the disturbance dynamics.

The system identification procedure

System identification is of great interest for expressing mathematically the dynamics of systems with exogenous stochastic disturbances whose temporal behaviour cannot be reliably modelled with first principles. Input and output datasets are collected to identify a model. It has developed since the sixties and is nowadays a well-established area, Ljung (1999). The identification procedure consists of three steps.

First, an input signal is applied to the system to sufficiently excite the system such that its main dynamics are revealed on the output signal. If the system at hand is a beam which is locally distorted by actuators regularly placed beneath the

beam, both the voltage inputs and the beam deformation are measured over time and stored into a dataset.

Second, a set of model candidates is defined according to some assumptions on the system dynamics. For example, assuming that the system is linear, time-invariant, and that prior knowledge on the statistical properties of the noise is available. Standard model representations for linear time-invariant systems are AutoRegressive with eXogenous inputs (ARX), AutoRegressive Moving Average with eXogenous inputs (ARMAX), Finite-Impulse Response (FIR), Output-Error (OE) and Box-Jenkins, Ljung (1999). When all the coefficients in the model set are unknown, the system is a black-box. In grey-box modeling, a matrix parametrization such as tri-diagonal, Toeplitz or sparse may be assumed on the system matrices to shrink the set of model candidates and derive tailored algorithms with reduced computational complexity in the upcoming third step. This second step which consists of choosing the matrix parametrization, also called *matrix structure*, is of prime importance for system identification of large-scale stochastic systems. If the model structure chosen is not included within the set of candidates that accurately represents the true system, the estimates are biased however large the dataset may be.

Third, the best possible approximate model in the set of candidates is searched. A subset of algorithms requires a cost function that balances between maximizing the data fit and reducing the complexity of the model to avoid over-fitting. On the contrary, identifying state-space models with subspace methods as described in Verhaegen and Verdult (2007) does not rely on a quality criteria but exploits mathematical properties to estimate the system matrices in a non-iterative manner. Whether the estimated model is of sufficient quality is assessed by evaluating the least squares fit on a dataset different from the one used for identification. The model may not be validated because either the dataset is not informative enough, the model set does not contain a good parametrization, the cost function (if needed) is not well-chosen, or the algorithm does not converge to a global minimum of the cost function.

Scalability for systems with a large number of inputs and outputs

When the disturbance is two- or three-dimensional in space, a sensor distributed over a regular array is used to provide measurements. A node (equivalently, subsystem) in a regular grid is defined by its location in that grid. A d-dimensional sensor array is a collection of nodes organized on a regular d-dimensional grid and such that each node outputs a signal which is a local information over the quantity of interest, like e.g. the turbulence field. Another terminology used for denoting such collection of nodes is a network.

Definition 1.1. *Let $d \in \mathbb{N}$. A system is said to be d-dimensional when both the sensors and actuators are regularly distributed on a d-dimensional spatial array.*

For a sensor array of size $N \times N$ with N large, system identification might be infeasible even as calibration step without constraints of control bandwidth, let alone an efficient implementation of closed-loop operations such as state updates

necessary for computing a prediction using matrix-vector multiplications. Heavy computations include the number of floating-points operations which scales with N^6 for system identification and the amount of required memory scaling with N^4. The required memory bandwidth to transfer data back and forth from the central unit to the computational units located on the device is rather dealt with by choosing the appropriate computing platform.

The system identification of linear time-invariant dynamic systems in open- and closed-loop with a moderate number of inputs and outputs is well understood, see e.g the textbooks Ljung (1999), Verhaegen and Verdult (2007) and Van den Hof (2018). However, state-of-the-art system identification methods handle in reasonable time and with limited computing resources only a moderate number of inputs and outputs. The algorithms either use a whole data batch of temporal samples to estimate the system matrices as presented in Verhaegen and Verdult (2007), or update recursively the estimates whenever a new measurement becomes available. Starting from an initial guess possibly random, the current estimate for the matrices of an autoregressive model is fused with the new available data using recursive least-squares techniques, Sayed and Kailath (1998). The latter option especially reduces the memory required for identifying autoregressive or state-space models as there is no need to store past data batches which is a real asset to scale to large systems. Chiuso et al. (2008) use a recursive version of the Predictor-Based Subspace IDentification for identifying mirror and turbulence dynamics for large-scale AO systems. For updating unstructured matrices of size $N^2 \times N^2$, the required storage and complexity for matrix-vector multiplications scale with $\mathcal{O}(N^4)$ which may be detrimental for systems with high control bandwidth.

1.1.2. Examples of multi-dimensional sensor grids

Systems with a large-scale multi-dimensional sensor and/or actuator array are used in engineering for various applications ranging from optics to flow control.

Data-driven predictive control for large-scale adaptive optics

Measuring a stochastic disturbance with spatial and temporal evolution and minimizing its effects occurs for example in AO systems. This application is described thoroughly in Section 1.4 and we stick for now to the analysis of the challenges with a mere input-output description of the plant. Figure 1.1 illustrates the turbulence fields flowing over the telescope aperture.

A combination of a sensor and a deformable mirror reshapes the distorted light wavefront to increase the resolving power of the telescope. The sensor is a two-dimensional array of size $N \times N$ as depicted in Figure 1.2 and provides at each sampling time in closed-loop a measure of the residual disturbance, i.e the atmosphere-induced disturbance minus the correction applied with the deformable mirror. A grid of actuators is located beneath the mirror whose shape is modified to minimize the influence of the atmosphere on the image.

Many instruments such as spectrographs and coronographs would benefit from increased scalability of large-scale adaptive optics algorithms. Examples include HARMONI (High Angular Resolution Monolithic Optical and Near-infrared Integral

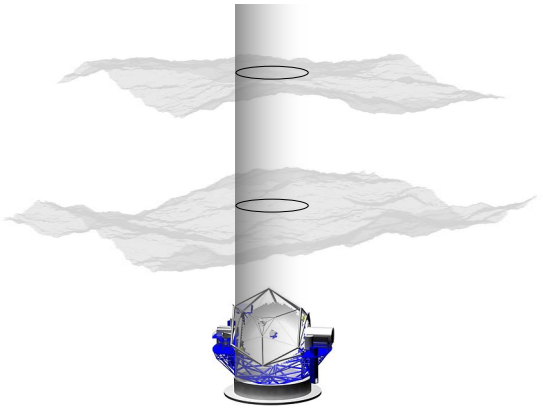


Figure 1.1: Two layers of turbulence flowing over a telescope. With the terminology developed in Section 1.4 of this introduction, each layer represent a wavefront. The isoplanatic angle between the star and the object of interest is not depicted. The disturbance sensed on the ground is the sum of both distortions introduced by each layer. The latter may be flowing at different wind speeds. Courtesy for the telescope schematic: `http://www.mpia.de/`.

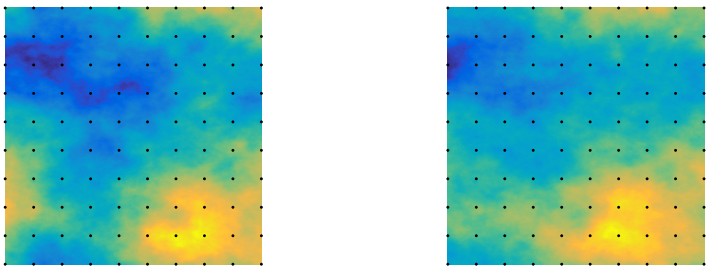


Figure 1.2: Example of a turbulence field consisting of the sum of two frozen layers propagating a different speed at (left) $t = t_0$, (right) $t = t_1 > t_0$. The sensor array is of size 10×10 and the nodes are represented with black dots. The data displayed in colour consists of actual disturbance screens used in the laboratory testbed in Chapter 6 and shifting one layer with respect to the other for simulating a non-zero wind speed. The spatial shift is set such that the rate at which the disturbance evolves between two sampling times corresponds to standard values. The correlations are not only spatial but also temporal.

field spectrograph), Neichel et al. (2016), for the European telescope and NFIRAOS (Narrow Field InfraRed Adaptive Optics System), Ellerbroek (2011), for the Thirty Meter Telescope. The latter system integrates seven 60×60 wavefront sensors to map the turbulence in the volume and over a wide field of view, and operates at 800Hz, Ellerbroek (2011). Predictive control for large-scale AO systems is also required for instruments such as GPI (Gemini Planet Imager), Poyneer et al. (2016), or SPHERE (Spectro-Polarimetric High-contrast Exoplanet REsearch), Petit et al. (2014).

1

System identification in wind farm control using wind speed measurements

Another example shown in Figure 1.3 is in the area of offshore wind farms, Gebraad (2014). Controlling the orientation of all the turbines to direct the downstream wake is essential for maximizing the overall power. It has been shown in Crespo et al. (1999) that orienting each turbine in the farm independently of its neighbours does not yield optimal performances for maximizing the overall power.

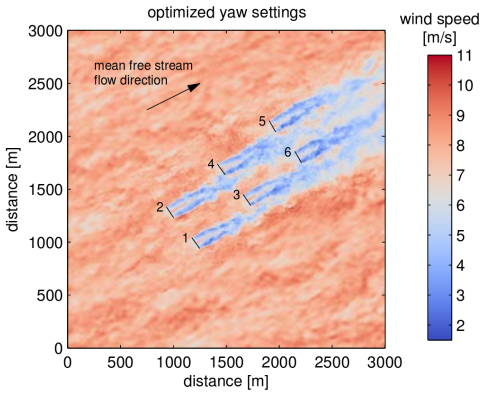


Figure 1.3: 3×2 wind plant rotated 10° w.r.t wind direction. Hub-height wind field at 800s simulated time as calculated by the software SOWFA. The black lines indicate the rotor positions and yaw orientation of each turbine. Courtesy: Gebraad (2014).

Although the control frequency is about one hertz, which is relatively slow compared to the AO application, the model describing the dynamics of the flow should be computationnaly efficient. One approach consists of discretizing the Navier-Stokes equation, Boersma et al. (2018), and deriving e.g ensemble Kalman filtering techniques, Doekemeijer et al. (2018). If we assume that the wind velocity measurements are available at each time sample on a regular three-dimensional grid, a model could be identified from data for predicting how the flow propagates. Although the practicality of this assumption needs to be evaluated, a compact data-driven model which could be handled online by the local processing units in each turbine would provide an alternative to the current state-of-the-art techniques relying on first principles.

Controlling the boundary layer between fluids and wings

Fluid control aims at reducing the drag and stabilize flows to delay the transition from laminarity to turbulence. In particular, when a flow propagates in a long pipe and is driven by differences of pressure, the Navier-Stokes equation boil down to a linear PDE for small variations around a steady-state. The near-wall flow is measured by pressure sensors and reshaped using blowing/suction distributed over the surface to avoid flow instabilities and sustain a laminated flow, Joshi et al. (1997). We may think of the arrangement of actuators and sensors in a two-dimensional plane as shown in Figure 1.4. The trade-off in fluid control (and similarly for wind

farms in the previous example) is to capture the essential dynamics for the speed and vorticity as required for feedback control without discretising the spatial domain with a high resolution grid as required for accurate numerical simulations. It has stirred off interest to compress the models and derive identification and controller synthesis methods, see for example Kim and Bewley (2007) and Inigo (2015).

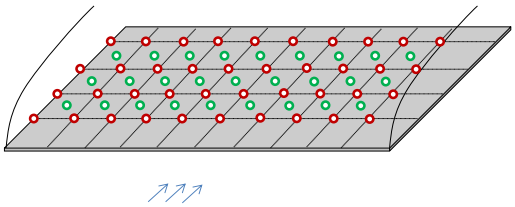


Figure 1.4: Schematic of a possible set of sensors and actuators for controlling a boundary layer flow. Actuators are in green, the sensors are in red and the flow is symbolized with the blue arrows. The actuators and sensors are arranged in a checkerboard pattern which reminds of the Fried geometry in AO applications.

Varied applications ranging from weather prediction to sociology
Even without actuators to control, some datasets can be recast as multi-dimensional, for example in weather prediction, Tsiligkaridis and Hero (2013). The area of statistics is likewise proficient with data that can be recast as multi-dimensional for example in sociology and the study of relational networks, Hoff (2015), although it does not feature a grid with actuators/sensors. The intensity of relations between two countries during a time span is scaled between 0 and 1. The definition of a dimension is less obvious than with sensor grids, but countries may be grouped according to some criteria such as the Gross Domestic Product or geographic proximity.

Adaptive optics systems, wind farms and flow control all rely on large-scale sensor measurements and are potential applications of the system identification and observer design methods that are studied in this thesis. The computational limitations of system identification algorithms to handle large sensor arrays have spurred the analysis of alternatives for deriving algorithms with linear computational complexity with respect to the number of sensor measurements.

1.1.3. The spatio-temporal impulse response

These three examples all relate to a propagation of a fluid whose shape we would like to control. For particular type of waves including the heat conduction or optical waves propagation in an empty medium, the behaviour is governed by a linear PDE featuring both spatial and temporal derivatives. A distributed parameter system is a system whose state-space is infinite-dimensional, Curtain and Zwart (1995). We illustrate with an example on heat conduction. A thin metal plate with homogeneous material density has a known temperature map, $T_0(\xi)$, over the field ξ at $t = t_0$.

1

Let the spatial boundaries be denoted with Ω. The heat propagates in the next time instants $t > t_0$ according to thermal conduction principles and subject to homogeneous Dirichlet boundary conditions,

$$\begin{cases} \frac{\partial}{\partial t}T(t,\xi) &= -c\nabla^2 T(t,\xi) + u(t,\xi) \\ T(t_0,\xi) &= T_0(\xi) \\ T(t,\Omega) &= 0 \end{cases} \tag{1.1}$$

where c is a positive constant, ∇^2 is the Laplacian operator and $u(t,\xi)$ some input applied at position ξ. Although a unique closed-form expression is derived assuming a separable solution in the spatial and time coordinates, an analysis of such systems for engineering applications relies on the discretised PDE obtained with finite-difference, hence giving rise to lumped parameter systems. We assume a uniform two-dimensional spatial grid of size $N \times N$, and discretize the set of equations (1.1) with,

$$\begin{cases} T_{i_1,i_2}(k+1) &= (1+4\alpha)T_{i_1,i_2}(k) - \alpha \sum_{(\bar{i}_1,\bar{i}_2)\in\mathcal{N}_{(i_1,i_2)}} T_{\bar{i}_1,\bar{i}_2}(k) + \Delta t u_{i_1,i_2}(k) \\ T_{i_1,i_2}(0) &= T_0(i_1,i_2) \\ T_\Omega(k) &= 0 \end{cases} \tag{1.2}$$

where $\alpha = c\frac{\Delta t}{\Delta \xi^2}$, $\Delta t, \Delta\xi$ the temporal and spatial discretisation steps and $\mathcal{N}_{(i_1,i_2)}$ is the neighbourhood of node (i_1,i_2) that includes the nodes $\{(i_1-1,i_2),(i_1+1,i_2),(i_1,i_2-1),(i_1,i_2+1)\}$. At time instant $k+1$, each node of the spatial grid acts as a subsystem and updates its own temperature with the knowledge of the temperature from its four closest neighbours. More generally, there is a wider class of systems that can be recast as multi-dimensional: vibrating plates also have a spatial-temporal behaviour governed by a PDE. A significant difference for system identification with the examples mentioned in 1.1.2 is that these lumped parameter systems are deterministic: there is no additional disturbance that perturbs the state of the system, and as a consequence, the knowledge of its neighbourhood is known whereas it is not the case when designing observers for large-scale stochastic systems.

A finite impulse response approximation of the model (1.2) highlights that the influence of the neighbourhood widens with increasing past temporal window. The less spatially damped the system is, the further away a subsystem impacts a given temperature. Both spatial and temporal dimensions are coupled. The coefficients of a spatio-temporal impulse response are depicted in Figure 1.5 (restricting the heat diffusion in (1.2) to a single spatial-dimension for simplifying the illustrations). Let us assume a rod discretized with 100 spatial positions and with non-zero initial condition in the middle. This impulse is propagated in both time and space and it illustrates how fast the information travels from one subsystem to the other, and how much they influence each other. The funnel causality for a spatially-invariant system, as introduced in Bamieh and Voulgaris (2005), is a function defined for every possible distance between two nodes, and equal to the first time at which a node is affected by a change of another one located at a distance x. It is fixed for the discretized PDE (1.2) but is on the contrary unknown for the applications such as adaptive optics, all the more as coupling between nodes can also be expressed with the covariance

matrix of the process noise in a stochastic state-space model. This notion of funnel causality is closely related to the parametrization of the system matrices, some of which are detailed in the next section. Bamieh and Voulgaris (2005) show that if the controller communicates faster than the plant, the optimization problem for designing a stabilizing quadratically optimal feedback gain is convex and a globally optimal solution is achieved. The subsystems of the controller exchange quicker information than the ones of the plant: its funnel causality can be approximated with good accuracy as slightly (depending on the decay rate) larger than the one of the plant and the feedback gain is denser than the system matrices. When the size of the discretization domain N is infinite, and the dynamics are spatially invariant as in (1.2), Bamieh (1997) shows that the optimal controller minimizing a global quadratic objective is also spatially invariant and is inherently localized. The infiniteness assumption does not hold in practice, but is motivated by the fact that the systems dynamics do not exceed the spatial range of the sensor array. The closest neighbours have the largest impact on the controlled input, and their influence when located further away decay at a rate which depends on the system parameters. These first considerations paved the way for further investigations of structured matrices in the context of observer/controller design.

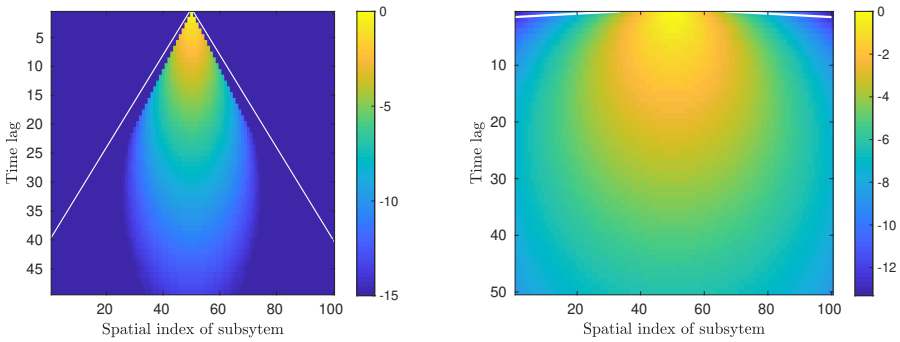


Figure 1.5: Spatio-temporal map for a one-dimensional string of subsystems, for the heat diffusion in (1.2) for $\alpha = -0.22$ (left) and for a case with no spatial delay when propagating the state (right). The entries are in log10. The larger the value in the map, the more the state of the neighbour (possibly in the past) contributes to the temperature of the 50-th subsystem of the heat rod. The furthest away in both time and space, the less it matters. The furthest away in time, the more the neighborhood spreads as more subsystems contributes to the value of the current state. The white lines defining a triangle, and that represent how fast information is shared within the subsystems of the controller, should contain the spatio-temporal impulse response of the plant for the optimization problem of deriving a structured controller in a convex manner.

1.2. Describing the set of model candidates

We step back from the notion of multi-dimensional grid for a moment to analyze the matrix structures that have been studied for alleviating the computations when identifying large LTI systems. The index N now refers to the total number of subsystems in the network. Each subsystem is associated with m inputs and p

1

outputs. Let $(\mathbf{u}(k), \mathbf{y}(k)) \in \mathbb{R}^{mN} \times \mathbb{R}^{pN}$.

Two model structures are investigated in this thesis for data-driven control. Vector AutoRegressive with eXogeneous inputs models with user-chosen temporal orders (n_u, n_y) relate the input $\mathbf{u}(k)$ to the output $\mathbf{y}(k)$,

$$\mathbf{y}(k) = \sum_{i=1}^{n_y} \tilde{\mathbf{A}}_i \mathbf{y}(k-i) + \sum_{i=0}^{n_u} \tilde{\mathbf{B}}_i \mathbf{u}(k-i) + \boldsymbol{\eta}(k) \tag{1.3}$$

for matrices $\tilde{\mathbf{A}}_i, \tilde{\mathbf{B}}_i$ of compatible sizes and where $\boldsymbol{\eta}(k)$ is a zero-mean white Gaussian noise with covariance matrix $\tilde{\mathbf{C}}_\eta$. When there is no input, $\mathbf{u}(k) = 0$ for all time samples, the equation (1.3) is an AutoRegressive model. The second structure we consider is the state-space model, which more generally represents infinite impulse responses via a state information $\mathbf{x}(k)$, and is presented here in the mixed deterministic-stochastic form,

$$\left\{ \begin{array}{rcl} \mathbf{x}(k+1) & = & \bar{\mathbf{A}}\mathbf{x}(k) + \bar{\mathbf{B}}\mathbf{u}(k) + \mathbf{w}(k) \\ \mathbf{y}(k) & = & \bar{\mathbf{C}}\mathbf{x}(k) + \bar{\mathbf{D}}\mathbf{u}(k) + \mathbf{v}(k) \end{array} \right. , \begin{bmatrix} \mathbf{w}(k) \\ \mathbf{v}(k) \end{bmatrix} \sim \mathcal{N}\Big(\mathbf{0}, \begin{bmatrix} \mathbf{Q} & \mathbf{S} \\ \mathbf{S}^T & \sigma_v^2 \mathbf{I}_J \end{bmatrix}\Big) \tag{1.4}$$

for $\bar{\mathbf{A}}$ with spectral radius strictly smaller than one, and where $\mathbf{w}(k), \mathbf{v}(k)$ are respectively process and measurement zero mean white Gaussian noise. The innovation form associated to the model (1.4) may be used in formulating the system identification problem. Introducing the Kalman gain $\bar{\mathbf{K}}$ and the state at time $k+1$ using all the information up to time k with $\mathbf{x}(k+1|k)$, the innovation form reads,

$$\mathbf{x}(k+1|k) = \big(\bar{\mathbf{A}} - \bar{\mathbf{K}}\bar{\mathbf{C}}\big)\mathbf{x}(k|k) + \big(\bar{\mathbf{B}} - \bar{\mathbf{K}}\bar{\mathbf{D}}\big)\mathbf{u}(k) + \bar{\mathbf{K}}\mathbf{y}(k) \tag{1.5}$$

Identifying the matrices in (1.3) or (1.5) scales at least with $\mathcal{O}(N^3)$ which seriously hampers its applicability for large systems. Although dealing with general models without structural assumptions on the matrices $\tilde{\mathbf{A}}_i, \tilde{\mathbf{B}}_i, \tilde{\mathbf{C}}_\eta$, or $\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathbf{K}}, \bar{\mathbf{C}}, \bar{\mathbf{D}}$, yields the most accurate representation, a compact representation of these matrices (in the sense that few parameters are needed to represent them) is key in identification, controller synthesis and real-time implementation.

Because we are mainly interested in the identification of stochastic systems, the innovation form of state-space models allows an identification algorithm unchanged with respect to the deterministic case. Let $\mathcal{S}$ denote an operator (not necessarily linear) mapping some set of parameters $\theta_{\mathbf{X}}$ to the matrix $\mathbf{X}$. It is assumed that, if $\big(\mathcal{S}(\theta_{\bar{\mathbf{A}}}), \mathcal{S}(\theta_{\bar{\mathbf{B}}}), \mathcal{S}(\theta_{\bar{\mathbf{C}}}), \mathcal{S}(\theta_{\bar{\mathbf{D}}}), \mathcal{S}(\theta_{\bar{\mathbf{Q}}}), \mathcal{S}(\theta_{\bar{\mathbf{S}}})\big) \approx (\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathbf{C}}, \bar{\mathbf{D}}, \bar{\mathbf{Q}}, \bar{\mathbf{S}})$, then $\big(\mathcal{S}(\theta_{\bar{\mathbf{A}}-\bar{\mathbf{K}}\bar{\mathbf{C}}}), \mathcal{S}(\theta_{\bar{\mathbf{B}}-\bar{\mathbf{K}}\bar{\mathbf{D}}}), \mathcal{S}(\theta_{\bar{\mathbf{K}}})\big) \approx (\bar{\mathbf{A}} - \bar{\mathbf{K}}\bar{\mathbf{C}}, \bar{\mathbf{B}} - \bar{\mathbf{K}}\bar{\mathbf{D}}, \bar{\mathbf{K}})$. The importance of preserving the structure of the system matrices in the Kalman gain, at least approximately, will be discussed in the subsection 1.2.3.

Deriving scalable algorithms with $\mathcal{O}(N)$ complexity may however be at the expense of performance loss depending on how close to reality the structural assumptions are. We are looking for a trade-off between data-sparsity of the model representation (what is the most concise way of expressing mathematically the behaviour of the system) and the bias between the true and approximated model structure (how far away the model is from the actual dynamics), the latter being responsible for performance loss in the prediction.

The remaining of this section is as follows. We first present different model structures that have been proposed for handling particular large-scale system identification. We then delve into the structure-preserving properties that a model set should have in order to derive efficient and scalable algorithms and which will guide the research in forthcoming chapters. Last, we discuss the Roesser model commonly used in image processing. This model structure stands apart from the rest of the patterns for structured modeling although it has been the standard for multi-dimensional state-space for long.

1.2.1. A local description of the spatial-temporal dynamics

Identifying the matrices in the state-space model (1.4) from standard subspace methods such as N4SID and MOESP, Verhaegen and Verdult (2007), is not feasible as these methods scale at the very least with $\mathcal{O}(N^3)$. The difficulties for the centralized methods to handle large-scale two-dimensional sensor arrays have already been noticed in Hinnen (2007) and stem from both the number of temporal samples to be measured and stored larger than N, and a QR decomposition or an SVD on matrices of size in the order of $N \times N$. A detailed explanation of the computational cost for SSARX is found in Chapter 3. Assumptions have been made in the literature on the system matrices in (1.3) and (1.4) to restrict the set of model candidates and propose efficient algorithms with the underlying idea of establishing a trade-off amongst the compactness of the model, the computational efficiency and the accuracy of the estimated representation for a given application.

A decentralized approach

The simplest state-space model which ignores all coupling between the subsystems assumes that each of them evolves independently from its neighbours. The matrices $\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathbf{K}}, \bar{\mathbf{C}}, \bar{\mathbf{D}}$ are block-diagonal leading to the local representation for the subsystem Σ_i,

$$\{\Sigma_i\}_{i=1..N} : \left\{ \begin{array}{rcl} \mathbf{x}_i(k+1) & = & \mathbf{A}_i\mathbf{x}_i(k) + \mathbf{B}_i\mathbf{u}_i(k) + \mathbf{K}_i\mathbf{e}_i(k) \\ \mathbf{y}_i(k) & = & \mathbf{C}_i\mathbf{x}_i(k) + \mathbf{D}_i\mathbf{u}_i(k) + \mathbf{e}_i(k) \end{array} \right. \tag{1.6}$$

The state $\mathbf{x}_i(k)$ is local and of small size with respect to the total number of subsystems N. This state-space model (1.6) is a linear time-invariant system with a small number of inputs and outputs. Identification and control is performed in a decentralized (equivalently, parallel) manner using standard textbook algorithms and with linear computational complexity with respect to the number of nodes. The set of subsystems is said to be homogeneous if and only if all subsystems are identical, i.e such that $\bar{\mathbf{A}} = \mathbf{I}_N \otimes \mathbf{A}$, and similarly for the other state-space matrices. The set of subsystems is otherwise heterogeneous.

The interconnected string of subsystems

Without parametrizing any structure on the controller, Bamieh (1997) and subsequently Bamieh et al. (2002) show that quadratically optimal controllers for spatially-invariant systems are localized in the sense that state feedback and observer gains decay exponentially with the distance. Motee and Jadbabaie (2008) extend the analysis dealing with possibly heterogeneous systems and introduce spatially decaying

operators to derive structure-preserving results on the solution of the Lyapunov and Riccati equation.

Therefore, instead of ignoring all coupling as in the previous paragraph, a first approach directly related to the lumped systems in 1.1.3 consists of assuming a sparse and localized Kalman gain. It translates into the following state-equation for subsytems interconnected along a string,

$$\{\Sigma_i\}_{i=1..N} : \left\{ \begin{array}{lcl} \mathbf{x}_i(k+1) & = & \sum_{j=i-1}^{i+1} \mathbf{A}_{i,j}\mathbf{x}_j(k) + \mathbf{B}_i\mathbf{u}_i(k) + \sum_{j=i-1}^{i+1} \mathbf{K}_{i,j}\mathbf{e}_j(k) \\ \mathbf{y}_i(k) & = & \mathbf{C}_i\mathbf{x}_i(k) + \mathbf{D}_i\mathbf{u}_i(k) + \mathbf{e}_i(k) \end{array} \right. \tag{1.7}$$

where $\mathbf{x}_0(k)$ and $\mathbf{x}_{N+1}(k)$ are $\mathbf{0}$, for all k. The inputs $\mathbf{u}_i(k)$ enter the subsystem Σ_i which receives unknown state information from its neighbours, $\mathbf{x}_{i-1}(k)$ and $\mathbf{x}_{i+1}(k)$. This representation originates from the discrete one-dimensional PDEs in distributed parameter systems.

The only quantities measured are $\mathbf{u}_i(k)$ and $\mathbf{y}_i(k)$, for all subsystems in the string, $i \in \{1, \ldots, N\}$. A challenge for system identification is that the states leaking into the local subsystems are often not measurable. If these interconnection signals would be known, the identification of each subsystem would be decentralized by recasting $\mathbf{x}_{i-1}(k)$ and $\mathbf{x}_{i+1}(k)$ as inputs. Without knowledge of these signals, the methods that have been proposed rely on approximating the local state $\mathbf{x}_i(k)$ such that estimating the matrices in (1.7) given the state, input and output data for many temporal samples boils down to a least-squares. The state $\mathbf{x}_i(k)$ is written as a linear combination of known signals, the collection of these signals being called a dictionary.

Approximating $\mathbf{x}_{i-1}(k)$ and $\mathbf{x}_{i+1}(k)$ with a well-chosen set of input and output data was initially investigated in Haber (2014). Haber (2014) estimates the state $\mathbf{x}_i(k)$ of a local subsystem as a linear combination of the input-output data of local subsystems, which are in its neighbourhood. It is shown that the size of the neighbourhood is directly related to the condition number of the finite-time observability Gramian, and hence, communication with all subsystems in the string is not necessary for estimating the local state. The exact knowledge of this neighbourhood for state estimation has been derived in Yu and Verhaegen (2018a). Yu et al. (2018a) and Yu and Verhaegen (2018a) propose a local identification of the systems relying on the following observation. Lifting the states of a small set of subsystems (or cluster) and then lifting in time to form the data equation creates a particular data-equation whose low-rank properties are used to isolate the cluster from the rest of the string.

Exploiting sparsity in autoregressive models

Rather than state-space models, Chiuso (2007) addresses the identification of autoregressive models within the Bayesian framework under the assumption that each sensor is related to a localized neighbourhood. Kernel regularization is used for inducing temporal stability and spatial sparsity. When moreover a few latent variables can explain the dynamic behaviour of the global system, the sparse plus low rank structure is studied in Zorzi and Chiuso (2015). One of the drawback for this specific representation is the inability to cope with large datasets and sensor arrays. A related work for the local identification of spatial-temporal dynamics of deterministic only

systems is Ali et al. (2011). It exploits sparsity to derive an Instrumental Variable method for identifying two-dimensional systems modelled with a transfer function having a Box-Jenkins structure.

None of these methods is exclusively limited to the analysis of a one-dimensional string and all extend to systems with more spatial dimensions d by connecting the subsystems to its $2 \times d$ closest neighbours. In this case, the number of neighbours grows linearly with the dimension.

Shortcomings for identification of multi-dimensional stochastic systems
A difficulty that arises for identifying spatial-temporal dynamics of stochastic systems is the knowledge of the neighborhood, and especially its size: which subsystems matter for computing a nearly optimal prediction for the state $\mathbf{x}_i(k+1|k)$ of each subsystem Σ_i? The number of neighbours should remain limited with respect to the total number of subsystems to enable efficient calculations.

Most importantly, these identification methods search for a local model estimation and have difficulties in assuming and/or imposing global properties such as the observability, controllability and stability of the overall system.

1.2.2. A modal analysis of large-scale systems

A modal view of interconnected subsystems stands in opposition to the sparsely interconnected set of subsystems by the global properties such as observability or controllability that can be guaranteed.

Decomposable systems
Massioni and Verhaegen (2009a) introduce decomposable systems to model a set of interconnected subsystems. It assumes that the interconnection pattern between the subsystems is known: to each collection of interconnected subsystems is associated a weighted adjacency matrix $\mathcal{P}$ that describes how the nodes are connected. Let $\mathcal{N}_i$ denote the set of indices associated with the neighbouring nodes of node i. Let α denote the number of neighbours. The entry $p_{i,j}$ is equal to $1/\alpha$ if the nodes i and j are connected, and 0 else.

Lemma 1.1. *(Massioni and Verhaegen (2009a)) Let $n \in \mathbb{N}$. Let $\mathbf{X}$ and $\mathbf{X}_{\mathcal{N}}$ belong to $\mathbb{R}^{n\times n}$. Assume that the matrix $\mathcal{P} \in \mathbb{R}^{N\times N}$ is diagonalizable (i.e there exists an invertible $\mathbf{S}$ such that $\mathbf{S}^{-1}\mathcal{P}\mathbf{S}$ is diagonal). For a matrix $\bar{\mathbf{X}}$ written as $\bar{\mathbf{X}} = \mathbf{I}_N \otimes \mathbf{X} + \mathcal{P} \otimes \mathbf{X}_{\mathcal{N}}$, then the matrix $\boldsymbol{\mathcal{X}} := (\mathbf{S} \otimes \mathbf{I}_n)^{-1}\bar{\mathbf{X}}(\mathbf{S} \otimes \mathbf{I}_n)$ is block-diagonal.*

The reverse implication is however not true in general. The set of all matrices $\bar{\mathbf{X}}$ such that $\bar{\mathbf{X}} = \mathbf{I}_N \otimes \mathbf{X} + \mathcal{P} \otimes \mathbf{X}_{\mathcal{N}}$ where $\mathbf{X}, \mathbf{X}_{\mathcal{N}} \in \mathbb{R}^{p\times q}$ is denoted with $\mathcal{D}_{\mathcal{P},p,q}$.

Definition 1.2. *(Massioni and Verhaegen (2009a)) Assume that the matrix $\mathcal{P} \in \mathbb{R}^{N\times N}$ is diagonalizable. A state-space system (1.4) is said to be decomposable when the matrices $\bar{\mathbf{A}} \in \mathcal{D}_{\mathcal{P},n,n}, \bar{\mathbf{B}} \in \mathcal{D}_{\mathcal{P},n,m}, \bar{\mathbf{C}} \in \mathcal{D}_{\mathcal{P},p,n}, \bar{\mathbf{D}} \in \mathcal{D}_{\mathcal{P},p,m}$.*

1

Such model assumption assumes that the subsystems are homogeneous and that the interconnection defined with $\mathcal{P}$ with all the neighbours is through the same matrix $\mathbf{A}_{\mathcal{N}}$. The state-space model (1.4) is then rewritten with:

$$\left\{ \begin{array}{rcl} \mathbf{x}_S(k) & = & \mathcal{A}\mathbf{x}_S(k) + \mathcal{B}\mathbf{u}_S(k) + \mathbf{w}_S(k) \\ \mathbf{y}_S(k) & = & \mathcal{C}\mathbf{x}_S(k) + \mathcal{D}\mathbf{u}_S(k) + \mathbf{v}_S(k) \end{array} \right. \tag{1.8}$$

where $\mathbf{x}_S(k) = (\mathbf{S} \otimes \mathbf{I}_n)^{-1}\mathbf{x}(k), \mathbf{u}_S(k) = (\mathbf{S} \otimes \mathbf{I}_m)^{-1}\mathbf{u}(k), \mathbf{w}_S(k) = (\mathbf{S} \otimes \mathbf{I}_n)^{-1}\mathbf{w}(k)$, $\mathbf{y}_S(k) = (\mathbf{S} \otimes \mathbf{I}_p)^{-1}\mathbf{y}(k), \mathbf{v}_S(k) = (\mathbf{S} \otimes \mathbf{I}_p)^{-1}\mathbf{v}(k)$. The matrices in calligraphic letters in (1.8) are block-diagonal, hence allowing to rewrite the state-space with decoupled equations each representing a so-called mode of the global system. System identification for handling the deterministic case of (1.8) is proposed in Massioni and Verhaegen (2009b) and Yu and Verhaegen (2017).

Circulant systems

Circulant systems are related to spatially-invariant ones as defined in Bamieh et al. (2002) where the sensor array is assumed infinite. The latter work introduced the decoupled control operations in the frequency domain resulting in parallel and inexpensive computations in the frequency domain. The class of circulant systems was introduced in Massioni and Verhaegen (2008) as a subclass of the decomposable systems. A modification to the string modelling discussed in (1.7) consists of connecting the first subsystem of the string to the last and form a circulant system assuming $\mathbf{x}_0(k) = \mathbf{x}_N(k), \mathbf{x}_{N+1}(k) = \mathbf{x}_1(k)$. Lifting the state equation in the spatial domain yields a global set of equations:

$$\left\{ \begin{array}{rcl} \mathbf{x}(k+1) & = & \mathcal{C}_{n,n}(\{\mathbf{A}_i\}_{i=1..N})\mathbf{x}(k) + \mathcal{C}_{n,m}(\{\mathbf{B}_i\}_{i=1..N})\mathbf{u}(k) + \mathbf{w}(k) \\ \mathbf{y}(k) & = & \mathcal{C}_{p,n}(\{\mathbf{C}_i\}_{i=1..N})\mathbf{x}(k) + \mathcal{C}_{p,m}(\{\mathbf{D}_i\}_{i=1..N})\mathbf{u}(k) + \mathbf{v}(k) \end{array} \right. \tag{1.9}$$

where the operator $\mathcal{C}_{a,b}(\{\mathbf{X}_i\}_{i=1..N})$ is a block circulant matrix defined from blocks equal to $\mathbf{X}_i$ of size $a \times b$. Any circulant matrix is diagonalized using the Fourier matrix and the inverse transformation holds as well. By applying a block-Fourier transformation to the input and output vectors, the system matrices in (1.12) are block-diagonalized which opens the way for decentralized algorithms dealing with a set of N decoupled modal systems of small sizes. More precisely, when the matrix $\mathbf{S}$ in Lemma 1.1 is the Fourier transform, then the reverse implication holds, thus the optimal controller for a circulant system is circulant. This property of circulant matrices is also essential for deriving an identification algorithm for deterministic systems in Massioni and Verhaegen (2008). The key benefit is that algorithms may now be carried out on the modal systems independently using standard methods derived for one-dimensional systems of moderate size. The overall computational cost boils down to the cost of the core operation of interest for one single modal system times the number of systems in the network, hence a linear computational complexity with respect to the number of nodes, N. This approach is very similar to what was proposed in Bamieh et al. (2002) for the class of spatially-invariant distributed parameter systems, and where observability, stability and optimal quadratic control were studied in the frequency domain.

Toward handling heterogeneous systems
Although the decoupling is very appealing for deriving efficient algorithms, the systems need to be homogeneous. Moreover, the network may include many interconnections contrary to the local approach where the neighbourhood shall be limited to avoid any curse of dimensionality when increasing the dimension. Massioni (2014) generalizes to the case where few heterogeneous systems are allowed to connect and exchange information and derives distributed control synthesis methods.

1.2.3. On the importance of preserving the structure in standard linear algebra operations

Stepping back from the identification, a number of matrix structures have shown interesting properties for deriving scalable algorithms e.g for control.

A first option for enforcing structure on a controller is to formulate an optimization problem whose cost function is the norm of a closed-loop transfer function from external disturbance to the regulated output, and with constraints of stabilizing the plant and satisfying information sharing specifications. When the subspace of authorized communication patterns between the subsystems of the controller is a subspace, this minimization problem is convex in the Youla domain if and only if a property called Quadratic Invariance holds. We refer the reader to Rotkowitz and Lall (2006) and Lessard and Lall (2016) for more details. However sparse the system matrices may be, such as multi-banded when lifting the local states in a temperature vector for the whole plate in (1.2), the open-loop transfer matrices are nonetheless dense. Thus, a set of admissible transfer functions for computing a sparse controller is not quadratic invariant. Wang et al. (2018) circumvent this limitation introducing the framework of System Level Approach, and assuming both localizability and separability of the cost function.

An alternative consists of deriving structured solutions to the DARE using iterative algorithms.

Examples of algorithms where it matters
Performing even standard linear algebra operations such as addition, multiplication or inversion on certain type of matrices may destroy the original matrix structure. For example, the product of a Toeplitz matrix with another Toeplitz matrix (with finite sizes) is not necessarily Toeplitz. Similarly, the product of a sparse matrix with another sparse matrix having a potentially different non-zero pattern may be full, or is at least denser. In general, the inverse of sparse matrices destroys sparsity. Such structure-preserving properties matter in deriving scalable algorithms, and these considerations are not limited to system identification but have further reaching implications for solving controller synthesis problems such as based on large-scale Lyapunov and Riccati equations.

Preserving the structure was already illustrated at the beginning of Section 1.2: the Kalman gain as computed from the Riccati equation should be approximated with good accuracy with the same structure as the matrices $\bar{\mathbf{A}}, \bar{\mathbf{C}}, \bar{\mathbf{Q}}, \bar{\mathbf{R}}$. For example, Yu et al. (2018a) propose a method for identifying deterministic state-space systems interconnected along a string. Using the algorithm for stochastic systems requires

1

that the matrix $\bar{\mathbf{A}} - \bar{\mathbf{K}}\bar{\mathbf{C}}$ is block tri-diagonal. It is not specific to this particular structure and similar observations will be made in Chapter 3.

The matrix sign provides another motivation as this operator allows to check stability and solve Lyapunov and Riccati equations, Kenney and Laub (1995). One way for calculating the matrix sign of a square matrix having no eigenvalue on the imaginary axis is the Newton sign iteration. It is an iterative algorithm that, starting from an initial matrix $\mathbf{Z}_0$ proceeds with $\mathbf{Z}_{\kappa+1} = \frac{1}{2}(\mathbf{Z}_\kappa + \mathbf{Z}_\kappa^{-1})$, where κ is the iteration counter. It is all the more relevant when these iterations can be performed efficiently, and consequently, when the addition and the inversion do not ruin the structure that $\mathbf{Z}_0$ may initially possess such that $\mathbf{Z}_\kappa$ is approximated with $\mathcal{S}(\theta_{\mathbf{Z}_\kappa})$.

Sequentially Semi-Separable matrices

This algorithm was successfully used in Rice (2010) when modelling the matrices in (1.4) as Sequentially Semi-Separable (SSS). To understand how such matrices help to carry out sums, multiplications and inversions efficiently, the SSS matrices are related to a set of state-space models that represent the spatial-temporal dynamics of an interconnected string of subsystems. Each of the subsystem is modelled with a mixed causal anti-causal linear time-invariant model and shares unknown interconnections with all the neighbours. It does not rely on a short-range interaction of the subsystems with its environment but rather on a limited set of matrix generators that model the spatial-temporal dynamics. Denoting the input disturbance with $\mathbf{w}_i(k)$, the performance measure $\mathbf{z}_i(k)$ and the left and right interconnection signals with $\mathbf{v}_i^\ell(k), \mathbf{v}_i^r(k)$, the dynamics of a subsystem Σ_i are written with,

$$\{\Sigma_i\}_{i=1..N} : \begin{bmatrix} \mathbf{x}_i(k+1) \\ \mathbf{v}_{i-1}^r(k) \\ \mathbf{v}_{i+1}^\ell(k) \\ \mathbf{z}_i(k) \\ \mathbf{y}_i(k) \end{bmatrix} = \begin{bmatrix} \mathbf{A}_i & \mathbf{B}_i^r & \mathbf{B}_i^\ell & \mathbf{B}_i^1 & \mathbf{B}_i^2 \\ \mathbf{C}_i^r & \mathbf{W}_i^r & \mathbf{0} & \mathbf{L}_i^r & \mathbf{V}_i^r \\ \mathbf{C}_i^\ell & \mathbf{0} & \mathbf{W}_i^\ell & \mathbf{L}_i^\ell & \mathbf{V}_i^\ell \\ \mathbf{C}_i^1 & \mathbf{J}_i^r & \mathbf{J}_i^\ell & \mathbf{D}_i^{11} & \mathbf{D}_i^{12} \\ \mathbf{C}_i^2 & \mathbf{H}_i^r & \mathbf{H}_i^\ell & \mathbf{D}_i^{21} & \mathbf{D}_i^{22} \end{bmatrix} \begin{bmatrix} \mathbf{x}_i(k) \\ \mathbf{v}_i^r(k) \\ \mathbf{v}_i^\ell(k) \\ \mathbf{w}_i(k) \\ \mathbf{u}_i(k) \end{bmatrix} \tag{1.10}$$

When concatenating column-wise the local state $\bar{\mathbf{x}}(k) = \begin{bmatrix} \mathbf{x}_1(k)^T & \dots & \mathbf{x}_N(k) \end{bmatrix}^T$, the global model is obtained,

$$\begin{bmatrix} \bar{\mathbf{x}}(k+1) \\ \bar{\mathbf{z}}(k) \\ \bar{\mathbf{y}}(k) \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{A}} & \bar{\mathbf{B}}_1 & \bar{\mathbf{B}}_2 \\ \bar{\mathbf{C}}_1 & \bar{\mathbf{D}}_{11} & \bar{\mathbf{D}}_{12} \\ \bar{\mathbf{C}}_2 & \bar{\mathbf{D}}_{21} & \bar{\mathbf{D}}_{22} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{x}}(k) \\ \bar{\mathbf{w}}(k) \\ \bar{\mathbf{u}}(k) \end{bmatrix} \tag{1.11}$$

where all matrices $\bar{\mathbf{A}}, \bar{\mathbf{B}}_1, \bar{\mathbf{B}}_2, \bar{\mathbf{C}}_1, \bar{\mathbf{D}}_{11}, \bar{\mathbf{D}}_{12}, \bar{\mathbf{C}}_2, \bar{\mathbf{D}}_{21}, \bar{\mathbf{D}}_{22}$ have a SSS structure. A SSS matrix is defined from a linear number of generators with respect to the string's size. For example, $\bar{\mathbf{A}}$ is defined with,

$$\bar{\mathbf{A}} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{C}_1^r\mathbf{B}_2^r & \mathbf{C}_1^r\mathbf{W}_2^r\mathbf{B}_3^r & \dots & \\ \mathbf{C}_2^\ell\mathbf{B}_1^\ell & \mathbf{A}_2 & \mathbf{C}_2^r\mathbf{B}_3^r & \dots & \\ \mathbf{C}_3^\ell\mathbf{W}_2^\ell\mathbf{B}_1^\ell & \mathbf{C}_3^\ell\mathbf{B}_2^\ell & \mathbf{A}_3 & \dots & \\ \vdots & \ddots & \ddots & \ddots & \\ & & & & \mathbf{A}_N \end{bmatrix} \tag{1.12}$$

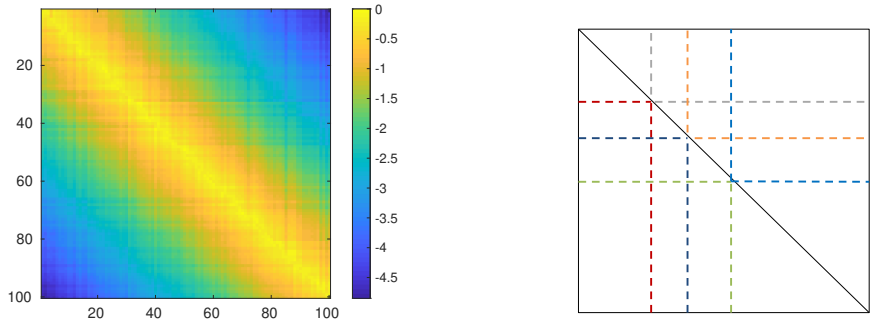


Figure 1.6: Left: 100×100 SSS matrix, $\bar{\mathbf{A}}$. The absolute value of the entries are represented in log10. The decay of the values away from the main diagonal can be more or less quicker depending on the spectral radius of $\mathbf{W}_i^\ell, \mathbf{W}_i^r$. Right: Schematic representation of the low-rank off-diagonal blocks. Each rectangle formed by two dashed lines of the same color and the two borders of the matrix is low-rank.

An example of such a matrix with strictly stable $\mathbf{W}_i^\ell, \mathbf{W}_i^r$ is shown in Figure 1.7, which also highlights the essential low-rank off-diagonal blocks of $\bar{\mathbf{A}}$. Summing, multiplying, or inverting efficiently such matrices can be done if the sizes of the matrices $\mathbf{W}_i^\ell, \mathbf{W}_i^r$ are much smaller than the number of interconnected subsystems along the string, N. As an illustration that the SSS structure is preserved while inverting, Figure 1.7 displays the singular values of one off-diagonal matrix and the one of the same submatrix in the inverted matrix. When adding or multiplying SSS matrices, the structure is kept as new generators are formed although the size of $\mathbf{W}_i^\ell, \mathbf{W}_i^r$ increases. The SSS structure is maintained throughout the Newton iterations provided the order of the matrices $\mathbf{W}_i^\ell, \mathbf{W}_i^r$ are truncated at each iteration, Rice (2010). Most importantly, the Kalman gain derived has a SSS structure which guarantees efficient state-feedback control.

This framework deals in a scalable manner with large strings of subsystems: both linear algebra operations and control to achieve global $\mathcal{H}_2$ performance were shown to be achievable within linear computational complexity in the string's size, Rice (2010). Identification algorithms are found in Rice and Verhaegen (2011) using the extended Kalman filter and Torres et al. (2015) with output-error methods. It is well-known that such type of algorithm requires a good initial guess to provide accurate estimates.

SSS matrices are closely related to the 1D string of interconnected systems that has been considered in (1.7). When setting the matrices $\mathbf{B}_i^1$, $\mathbf{W}_i^r$, $\mathbf{W}_i^\ell$, $\mathbf{L}_i^r$, $\mathbf{L}_i^\ell$, $\mathbf{V}_i^r$, $\mathbf{V}_i^\ell$, $\mathbf{H}_i^r$, $\mathbf{H}_i^\ell$ to zero, and $\mathbf{C}_i^r$,$\mathbf{C}_i^\ell$ to the identity, the state-space (1.12) is identical to the simplified model (1.7). When the subsystems are interconnected in two dimensions, the matrix $\bar{\mathbf{A}}$ in (1.12) is similarly built with the only difference that now $\mathbf{A}_1, \mathbf{C}_2^\ell, \mathbf{B}_1^\ell$, etc. are also SSS. The product of two SSS being SSS, the matrix $\bar{\mathbf{A}}$ is then a block-matrix with SSS structure with blocks themselves SSS. The number of generators is however much larger as it scales exponentially with the dimension, i.e 7^d generators are required to form a d-level SSS matrix.

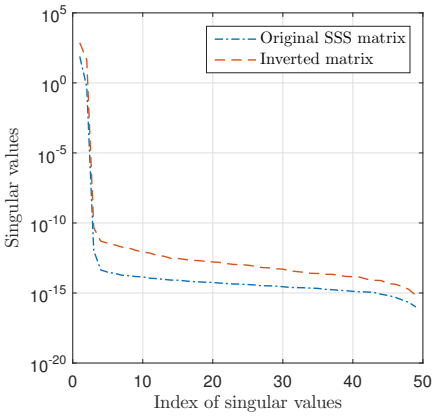


Figure 1.7: Singular values for one low-rank off-diagonal matrix, $\bar{\mathbf{A}}(50:100, 1:49)$ for the SSS matrix and its inverse. An example of such submatrix is shown in Figure 1.7. The rank of these submatrices is equal to two. Although it can not be deduced from this plot only, the rank of the off-diagonal blocks are not increased when inverting a SSS matrix.

The extension of the 1D SSS methods to higher spatial dimensions gives rise to multi-level SSS problems, for which up till now no efficient solution for identification and control exist. The Hierarchical Semi-Separable structure represents a large matrix with a set of generators following a pattern of low-rank matrices as shown in Figure 1.8. It presents structure-preserving properties for standard matrix operations although it has not been studied in the context of LTI dynamical systems and it is not clear which network structure it would model.

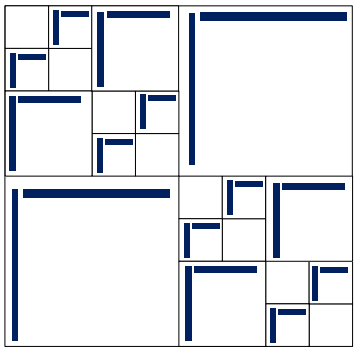


Figure 1.8: Pattern of low-rank matrices (in blue) within a Hierarchical Semi-Separable matrix.

Localized systems

Still dealing with the string of interconnected systems, the review in Benzi et al. (2017) has put forward another class of matrices maintaining the structure for matrix

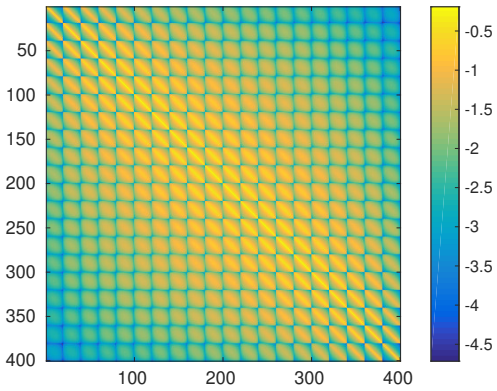


Figure 1.9: Inverse of $\mathbf{I} \otimes \mathbf{M} + \mathbf{M} \otimes \mathbf{I}$ where $\mathbf{M}$ is tri-diagonal, symmetric and positive definite ($m_{i,i} = 2, m_{i,i-1} = 1, m_{i-1,i} = 1$). The absolute values of the entries are plotted in a logarithmic scale.

multiplication and inverses.

A key property in the derivations in Haber (2014) is the decaying pattern of the observability and controllability Gramians, which are positive-definite matrices whose inverse is also decaying away from the main diagonal. This work relies on the property that the inverse of banded positive-definite matrices belong to the class of off-diagonally decaying matrices, that are such that their elements in absolute value decay as moving away from the main diagonal, Benzi et al. (2017). Computing an approximate inverse of a banded positive-definite matrix is achieved with a complexity that grows linearly with the string's size. Moreover, Canuto et al. (2014) study the decay rate of multi-level matrices of the form $\mathbf{I} \otimes \mathbf{M} + \mathbf{M} \otimes \mathbf{I}$, where $\mathbf{M}$ is tri-diagonal symmetric and positive definite, as seen in Figure 1.9.

Using the approximate inverse is not limited to identification but is also relevant for solving large-scale Lyapunov equations, Haber and Verhaegen (2016), and optimal control problems, Haber and Verhaegen (2018). In these works, the target is to derive a sparse feedback matrix for systems whose state update matrix $\bar{\mathbf{A}}$ is symmetric negative definite with a banded pattern (as obtained from the discretisation of PDEs). The Newton iterations consist of solving the Riccati equation through a sequence of Lyapunov equations and inverses, Benner et al. (2008). As a first step, Haber and Verhaegen (2016) show that the solution of the Lyapunov equation is spatially localized and computes efficiently an approximation with a sparse banded matrix.

Maintaining the same structures for standard linear algebra operations such as addition, multiplication and inversion is crucial for deriving efficient algorithms suited to the large dimensions of the problem at hand. The SSS matrix and the banded positive-definite matrix are two examples of parametrizations with favourable properties.

1

1.2.4. Roesser models in image processing

This paragraph stands very much apart from the previous ones particularly because of the manner the time dimension is being integrated into the state equations. The class of multi-dimensional systems was originally introduced in Givone and Roesser (1972). It is used for modelling the dynamics of an array of identical cells that are connected in a regular pattern. Each cell acts as a subsystem (or node) with its own state, input and output. In a two-dimensional setting, a cell is influenced by the states from the left and upper cells: the state information flows horizontally and vertically in a single direction. Roesser (1975) focuses on image processing and introduces a partition of the global state into horizontal and vertical states, respectively denoted with $\mathbf{x}_1$ and $\mathbf{x}_2$. These states are coupled together with the discrete state-space equation:

$$\begin{cases} \begin{bmatrix} \mathbf{x}_1(i_1+1,i_2) \\ \mathbf{x}_2(i_1,i_2+1) \end{bmatrix} &= \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1(i_1,i_2) \\ \mathbf{x}_2(i_1,i_2) \end{bmatrix} + \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \end{bmatrix} \mathbf{u}(i_1,i_2) + \begin{bmatrix} \mathbf{K}_1 \\ \mathbf{K}_2 \end{bmatrix} \mathbf{e}(i_1,i_2) \\ \mathbf{y}(i_1,i_2) &= \begin{bmatrix} \mathbf{C}_1 & \mathbf{C}_2 \end{bmatrix} \begin{bmatrix} \mathbf{x}_1(i_1,i_2) \\ \mathbf{x}_2(i_1,i_2) \end{bmatrix} + \mathbf{D}\mathbf{u}(i_1,i_2) + \mathbf{e}(i_1,i_2) \end{cases} \tag{1.13}$$

The state-space model (1.13) is a quarter-plane causal filter: computing the output at position $\bar{i}_1, \bar{i}_2$ uses the input data for $i_1 \leq \bar{i}_1, i_2 \leq \bar{i}_2$. Therefore, i_1 or i_2 may represent time. If the time dimension is left out as commonly done in image processing, the state-space representation (1.13) models the spatial dynamics by introducing horizontal and vertical states that are coupled together through some unknown interconnection.

The applications of the quarter-plane causal Roesser model (1.13) include mainly image processing, and especially what concerns image restoration and deblurring. When an image is corrupted by additive white Gaussian noise, its resolution decreases. The general idea is to consider the stochastic form of the model (1.13), associate a state to each RGB pixel and estimate the system matrices including the Kalman gain from intensity values in order to retrieve an image with as little blur as possible. Research on deriving a Kalman filter with the Roesser modelling in its stochastic version includes Wu (1985) and the recent contribution Ramos and Mercère (2018) who derive a subspace identification algorithm.

However, the Roesser model suffers from at least two drawbacks. The first one is that it handles specifically homogeneous systems although images may be spatially non-stationary. The second is related to causality. As seen in (1.2), the temperature of a single node at time instant $k+1$ depends on the neighbourhood and spatial causality is not required. Such assumption appears to be destructive for e.g modelling the spatial dynamics of deformable mirrors as pointed out in Voorsluys (2015). Lele and Mendel (1987) describe a full plane non-causal filter that is the linear combination of four quarter-plane RSD models (1.13) although the extension to larger dimension does not result in compact models. In essence, a finite-impulse response for the output $\mathbf{y}(i_1,i_2)$ is decomposed into four non-overlapping quadrants (for uniqueness purposes).

1.3. Research question

As can be seen from the literature review in Section 1.2, many structures have been assumed on the system matrices of state-space or auto-regressive models to overcome the computational issues that arise when the number of inputs and outputs is large. When a disturbance enters a large-scale system, the underlying question to derive tailored algorithms for prediction is: what is the best approximate structure of the Kalman gain $\bar{\mathbf{K}}$ and thereafter, of the matrix $\bar{\mathbf{A}} - \bar{\mathbf{K}}\bar{\mathbf{C}}$, for multi-dimensional systems?

The spatial dynamics are embedded within the structure of the matrices. For example, the spatial invariance is represented with a block-Toeplitz pattern, the spatial invariance and infinitely large dimensions with a circulant matrix (of finite size). Because of the finite size of the sensor array and the edges, the optimal controller is not spatially-invariant in spite of the regular grid. Deriving a Kalman gain assuming spatial invariance in the middle of the aperture only would require to invert a particular structured matrix when solving the DARE. Such an hybrid approach was proposed in Rice (2010) with the Almost-Toeplitz SSS structure: only the local dynamics at the edges are spatially-varying. However much the SSS structure efficiently deals with dense though data-sparse matrices stemming from an heterogeneous string of subsystems, its extension for block-matrices is unclear. Parametrizing multi-dimensional systems with SSS matrices embedded into SSS matrices as explained in Section 1.2.3 is such that the number of matrices needed for writing the state-space model scales exponentially with the dimension of the grid. In general, the results in Rice (2010) are limited to the systems in one spatial dimension.

The only current alternative to the assumption of spatial invariance is sparsity. In adaptive optics, although the matrix $\mathbf{A}$ is sparse (multi-)banded, the noise covariance matrix is $\mathbf{R}$ diagonal, and the sensor measures local information about the disturbance, it is not true in general that the Kalman gain is sparse. Similarly, the identification of stochastic models following the guidelines in Yu and Verhaegen (2018a) laid for deterministic systems requires to assume that the Kalman gain is block tri-diagonal (or, with eight non-zero block-diagonals when dealing with two spatial dimensions) and is therefore too restrictive.

In this thesis, the main research question is formulated as follows:

What is a dense, data-sparse and structure-preserving representation for identifying from data and in a scalable manner the spatial and temporal dynamics of multi-dimensional stochastic systems?

The capabilities of the model structure will be nuanced: the data-sparse or structure-preserving properties might be valid up to a certain extent that we will evaluate.

1

Analysis of research question We have so far discussed the multi-dimensional systems as systems that feature a sensor (and actuator) array but there is no adequate literature dealing with a tailored representation of LTI models. Let $(i, d) \in \mathbb{N}^2$ such that $1 \leq i \leq d$ and a tuple of integers $(J_1, \ldots, J_d)$. Let J be the total number of output signals in the array such that $J = \prod_{i=1}^{d} J_i$. Each node provides p outputs where $p \ll \min(\{J_i\}_{i=1..d})$.

Definition 1.3. *Let $d \in \mathbb{N}$ and $(J_1, \ldots, J_d) \in \mathbb{N}^d$. A real-valued tensor of order d is defined as belonging to $\mathbb{R}^{J_1 \times \ldots \times J_d}$.*

For example, a tensor of order one is a vector, and of order two is a matrix. Importantly, the integer d may be equal -but not necessarily- to the dimension the sensor array. In Definition 1.4, we precise the class of multi-dimensional models.

We remind the reader of the definition of a system as introduced in the beginning of this chapter: it relates to the physical objects like a sensor array and it differs from a model which is the mathematical relation which relates the input and output. Some examples of models were introduced when describing the second step of the identification procedure in Section 1.2. The model is typically a state-space representation, while the system is the physical object.

Definition 1.4. *Let $d \in \mathbb{N}$. Let $(I_1, \ldots, I_d)$ and $(J_1, \ldots, J_d)$ two sequences of integers and $(I, J) \in \mathbb{N}^2$ such that $I = \prod_{i=1}^{d} I_i$ and $J = \prod_{i=1}^{d} J_i$. Let $\big(\mathbf{u}(k), \mathbf{y}(k)\big) \in \mathbb{R}^I \times \mathbb{R}^J$ represent respectively the input and output data at time instant k.*
A model for a system is said to be d-dimensional when writing the input-output relationship involves the tensor representation of order d for $\mathbf{u}(k), \mathbf{y}(k)$, i.e the tensors $\boldsymbol{\mathcal{U}}(k) \in \mathbb{R}^{I_1 \times \ldots \times I_d}$ and $\boldsymbol{\mathcal{Y}}(k) \in \mathbb{R}^{J_1 \times \ldots \times J_d}$ are such that $vec\big(\boldsymbol{\mathcal{U}}(k)\big) = \mathbf{u}(k)$ and $vec\big(\boldsymbol{\mathcal{Y}}(k)\big) = \mathbf{y}(k)$.

This definition allows a lot of freedom in choosing the dimension of the model based only on how the data is stored: there are many ways to reshuffle the input and output vectors into tensors. For example, a system with 4 inputs and 4 outputs may be written with a one or two-dimensional model. This non-uniqueness will be discussed in Chapter 4 and 6.

To summarize, the dimensions of a tensor are the J_i whereas the order is d. The dimension of a system is the number of spatial dimensions in the sensor array, and the dimension of a model is the order of the associated tensor representation. The dimension of a model representation is not necessarily equal to the dimension of the sensor array.

The definition 1.4 writes the input and output with tensors rather than with vectors. How to write a compact state-space model when the input-output data are tensors? A way to derive closed-form solutions for a multi-dimensional PDE is to assume that the set of candidate functions are separable in the spatial and temporal dimensions, i.e for a function f, we have $f(x, y, t) = f_x(x) f_y(y) f_t(t)$. Figure 1.10 illustrates. Is there a relationship between the separability of an underlying function and a compact representation of multi-dimensional dynamical systems? If we assume in the simplest case that f_x, f_y and f_t are linear in the parameters, the function f is multi-linear in the sense that fixing all variables but one yields a linear function.

Are multi-linear parametrization of the system matrices related to multi-dimensional dynamical systems? If we assume that such a multi-linear parametrization of the system matrices compresses the system data, can we formulate optimization problems with fewer variables maybe at the expense of convexity rather than large-scale convex ones? Particularly, how would the trade-off between model accuracy, computational complexity, and memory storage be modified when instead parametrizing the system matrices with multi-linear operators?

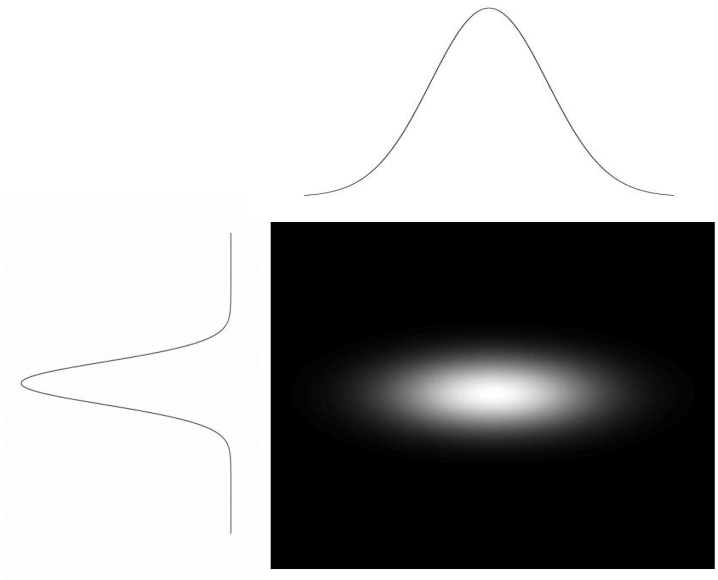


Figure 1.10: Schematic illustrating the separability of a two-dimensional function into its horizontal and vertical coordinates. The one-dimensional functions need not to be Gaussian as shown above.

These questions are answered in the Chapters 2 to 5. The framework we introduce in this thesis does neither rely on spatial-invariance nor on sparsity, and we will shed new light in the analysis of spatial-temporal systems.

1.4. Controlling large-scale adaptive optics systems

Adaptive optics has been briefly mentioned in 1.1.2 as an illustration of multi-dimensional stochastic systems and is used as a case study to validate the new methods developed in this thesis in simulations and in a laboratory experiment. This section is organized as follows. The propagation of light to the telescope is first presented, then the AO closed-loop is discussed. A third part introduces first principles and data-driven models for the spatial-temporal dynamics of the disturbance. Last, we review the scalability of these methods as a function of the size of the sensor.

1

1.4.1. Seeing-limited and diffraction-limited imaging systems

Let $\boldsymbol{\rho} \in \mathbb{R}^2$ represent the spatial coordinates (x, y) in a two-dimensional plane orthogonal to the direction of propagation of the light, and j the complex number satisfying $j^2 = -1$. The monochromatic light can be modelled by the electromagnetic field,

$$u(\boldsymbol{\rho}, t) = a(\boldsymbol{\rho})\mathrm{Re}\big(e^{j(wt-\phi(\boldsymbol{\rho}))}\big) \tag{1.14}$$

where $a(\boldsymbol{\rho})$ is the (real) amplitude, $w = 2\pi\nu$ the temporal pulsation associated to the frequency ν, and $\phi(\boldsymbol{\rho})$ is the phase expressed in radians. While propagating through atmospheric turbulence, it is generally assumed that the amplitude is constant, $a(\boldsymbol{\rho}) = a$. The term $U(\boldsymbol{\rho}) = ae^{-j\phi(\boldsymbol{\rho})}$ is the complex amplitude of the wave. Assuming a propagation according to the Fraunhofer diffraction, the intensity I in the focal plane of the telescope is derived from the squared modulus of the Fourier transform of the complex amplitude $U(\boldsymbol{\rho})$,

$$I(x, y) \propto |\iint_{x_0,y_0\in\Omega} e^{-j\phi(x_0,y_0)}e^{-j\frac{2\pi}{\lambda f}(xx_0+yy_0)}dx_0dy_0|^2 \tag{1.15}$$

where Ω is a circular aperture of diameter D, and f is the focal length of the imaging lens. If the phase $\phi(x_0, y_0)$ is independent of the spatial coordinates, (1.15) reduces to,

$$I(x, y) \propto |\iint_{x_0,y_0\in\Omega} e^{-j\frac{2\pi}{\lambda f}(xx_0+yy_0)}dx_0dy_0|^2 \tag{1.16}$$

After integrating (1.16), and introducing θ the angular coordinate in the focal plane equal to $\sqrt{x^2+y^2}/f$ for small angles, the image of a point source is expressed with the Point Spread Function, or Airy pattern,

$$p_0(\theta) = \frac{\pi D^2}{4\lambda^2}\Big(\frac{2J_1(\pi D|\theta|/\lambda)}{\pi D|\theta|/\lambda}\Big)^2 \tag{1.17}$$

where J_1 is the Bessel function of the first kind. The first dark ring occurs at,

$$\sin(\theta) \approx 1.22\frac{\lambda}{D} \tag{1.18}$$

The light is altered by the telescope aperture if D is finite and the image of a star through an optical system even if there would be no atmosphere is a bright spot surrounded with alternating dark and bright rings. An imaging system with pupil size D cannot distinguish objects separated with angular distances smaller than θ. Widening the pupil increases the resolving power of the telescope as much as it increases the signal-to-noise ratio by collecting more photons within the pupil.

The wavefront is defined as a surface of equal phase. For example, imagine a parabola with constant phase value ϕ_0. The wavefront is the surface defined from the set of coordinates $(x, y, \frac{2\pi}{\lambda}\delta)$ where $\delta = \sqrt{x^2+y^2}$. The distance δ is denoted in optics as the difference of path length.

In astronomy, the star is assumed infinitely far away and therefore the wavefront is approximated as plane before crossing the atmosphere layer, i.e it is independent

of the spatial coordinates $\boldsymbol{\rho}$. When the wave propagates through an heterogeneous and possibly time-varying medium, the wavefront is no longer planar and the PSF does not resemble the Airy disk anymore: the intensity is rather spread out. The imaging system is then said to be seeing-limited rather than diffraction-limited.

1.4.2. Adaptive optics systems

When a single star is used as a reference, the optical system is said to be single conjugated. Multiple reference stars may be used to widen the field of view but are not explored further in this thesis. Figure 1.11 illustrates a single-conjugate adaptive optics system, and Figure 1.12 the difference it makes for imaging through turbulence.

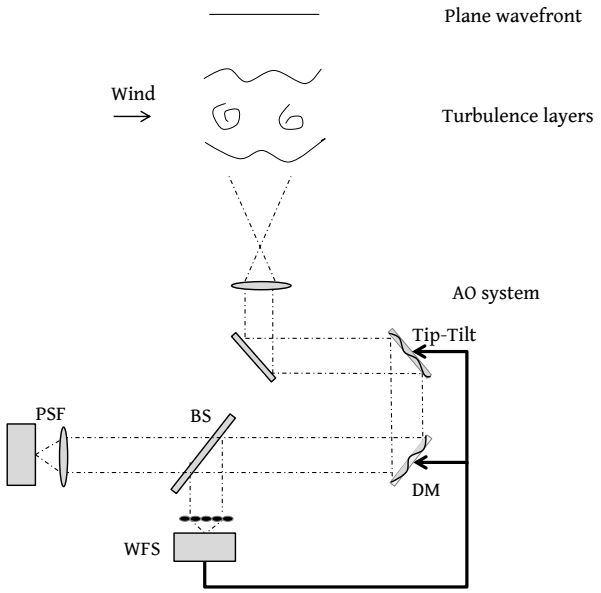


Figure 1.11: Schematic of an AO system. The plane wavefront is distorted by the atmospheric turbulence whose temporal dynamics are partially wind-driven. The telescope's largest mirrors (not represented) fold the light beam that is directed toward the AO system. It is then reshaped using the tip-tilt mirror and the deformable mirror (DM) which both reshape the wavefront based on the sensor signals fed back by the wavefront sensor (WFS). The reference star is imaged on the Point Spread Function (PSF) camera.

Atmospheric turbulence

The wavefront aberrations on the ground are caused by inhomogeneities in the refraction index caused by local variations of temperature, densities and water vapor content and are essentially driven by the wind. Kolmogorov (1991) proposed to

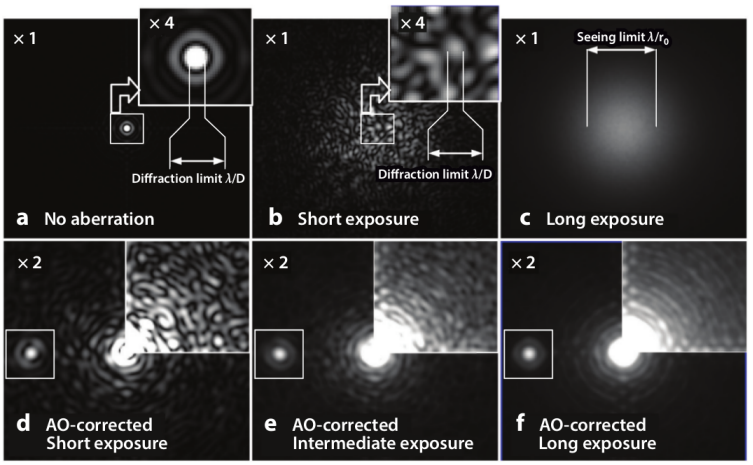


Figure 1.12: Simulated PSFs for an 8 meter telescope with a wavelength in near-infrared. (a) Diffraction-limited PSF in the absence of wavefront aberrations. (b) Short-exposure image showing diffraction-limited speckles. (c) Long-exposure seeing-limited image. (d-f) AO-corrected images. Courtesy: Guyon (2018).

model the dynamics of the turbulence as large eddies that collapse one onto another into smaller structures which do not sustain but rather dissipate by viscous friction.

Travelling through a medium with spatially heterogeneous refraction index creates optical path differences. Let z denote the height coordinate along the line of sight (which corresponds to the vertical altitude for sources located azimuthal at $z = h$). The phase difference is a linear function of the refraction index integrated over the line of sight,

$$\phi(\boldsymbol{\rho}) - \Phi = \frac{2\pi}{\lambda} \underbrace{\int_0^h n(\boldsymbol{\rho}, z) dz}_{\delta(\boldsymbol{\rho})} \tag{1.19}$$

for Φ a constant. Even though the path difference $\delta(\boldsymbol{\rho})$ is assumed independent of the wavelength, this is not the case for the phase difference incurred and as a consequence, for the image quality. The larger the wavelength, the smaller the wavefront distortions in Euclidean norm, and the larger the diffraction-limited angular resolution as seen in 1.4.1. AO operates for light within a narrow range of wavelength, and a good correction is more easily achieved in the infra-red spectrum.

The volume crossed by the lightwave is generally modelled mathematically with a linear combination of infinitely thin layers, independent, stationary, each driven by a wind blowing at a specific speed and direction. Each of these layers, or phase screens, is assumed to be a zero-mean Gaussian signal with a covariance function depending on few physical parameters related to the size of the eddies,

$$C_\phi(r) = \Big(\frac{L_0}{r_0}\Big)^{5/3} \frac{\alpha}{2} \Big(\frac{2\pi r}{L_0}\Big)^{5/6} K_{5/6}\Big(\frac{2\pi r}{L_0}\Big) \tag{1.20}$$

where r is the distance between two phase points $\boldsymbol{\rho}_1 = (x_1, y_1)$ and $\boldsymbol{\rho}_2 = (x_2, y_2)$, $K_{5/6}$ is the modified Bessel function of the third type and α a constant such that:

$$\alpha = \frac{2^{1/6}\Gamma(11/6)}{\pi^{8/3}} \Big(\frac{24}{5}\Gamma(6/5)\Big)^{5/6} \tag{1.21}$$

The parameter L_0 is known as the outer scale: the larger L_0, the more coupling between far away wavefront values, i.e the denser the wavefront covariance matrix. The Fried parameter r_0 is defined as the diameter of a circular area over which the root-mean-square of the wavefront is equal to 1 radian. The smaller r_0, the stronger the turbulence. Realistic values for L_0 and r_0 are 20m and $5 - 20$cm respectively. The equation (1.20) reflects the isotropic property of the turbulence as the spatial statistical properties reduce to a one-dimensional function. For a wavefront discretized on a regular square grid and lifted into a vector, its covariance matrix $\mathbf{C}_{\boldsymbol{\phi},0}$ is Toeplitz with Toeplitz blocks and cannot be approximated with good accuracy as sparse.

Noll (1976) quantifies the wavefront root mean square error for Kolmogorov turbulence associated with each of the wavefront modes. Without AO, the residual phase variance is equal to $1.029\big(D/r_0\big)^{5/3}$ compared to $0.134\big(D/r_0\big)^{5/3}$ when correcting the tip and tilt modes only. Therefore, instead of representing the wavefront on a zonal basis as in the previous paragraph, it may be decomposed into a modal basis to reduce the dimensionality for control to a given number of modes. A modal approach also allows to reconstruct and predict only modes that can be corrected with the DM. The lowest order modes such as tip and tilt contribute the most to the turbulence, let alone when considering vibrations and wind-shake on the mechanical structure of the telescope which increase all the more the discrepancy with the other modes. A drawback of modal representations of the phase however, is that they are not suited to exploit in a distributed or sparse manner the localized properties of the sensor and deformable mirror.

The deformable mirror

The control is usually applied with two deformable mirrors, a first one with large stroke and few actuators to correct the tip-tilt modes and a second one with many degrees of freedom although lower stroke to correct higher spatial frequencies.

Membrane mirrors with continuous face sheets are a common choice for integrating a large number of actuators. Each actuator is coupled to its closest neighbours: the interaction is limited to its close neighbourhood and is modelled with a two-dimensional Gaussian influence function identical for each actuator. Typical configurations are such that the values for the four closest neighbours of the i-th actuator reach $15 - 20\%$ of the value set on the i-th actuator. Mirrors relying on the micro-electromechanical technology have their first resonant frequency much larger than usual sampling frequencies of the sensor, and hence settle sufficiently fast to its steady state to neglect its temporal dynamics.

Between two consecutive sampling times, the input is maintained to the same value using a zero-th order hold. A one-step delay is assumed between the time at which the control inputs are applied and the time at which the wavefront is actually

induced. When applying $\mathbf{u}(k)$ at time instant kT_s, the relationship between the control inputs and the wavefront induced by the mirror only $\boldsymbol{\phi}_t^m$ is,

$$\boldsymbol{\phi}_t^m = \mathbf{H}\mathbf{u}(k), \text{for all } t \in [kT_s, (k+1)T_s] \tag{1.22}$$

The matrix $\mathbf{H}$ is sparse and two-level Toeplitz with an adequate choice of the phase sampling points and without failing actuators.

The mirror cannot take any arbitrary shape, and therefore there remains a residual error between the reconstructed wavefront on a zonal basis and the applied correction. The variance of the fitting error is evaluated as follows, Hardy (1998),

$$\sigma_{\text{fit}}^2 = \kappa_f \Big(\frac{d_t}{r_0}\Big)^{5/3} \tag{1.23}$$

where d_t is the inter-actuator spacing projected on the primary aperture, and $\kappa_f = 0.28$ for membrane mirrors.

The wavefront sensor

A commonly used sensor is the Shack-Hartmann sensor. It is composed of a two-dimensional array of micro-lenses, each of which focuses the wavefront located over its aperture on a camera placed at the back focal point. When the pitch of the array is large enough, the pattern observed on the camera is a set of independent Airy patterns that are located on a regular grid if the wavefront is flat. The center of gravity of each Airy pattern is first computed and the position of these centroids is used for reference. Local tilts of the wavefront deviate the point where light rays focus on the CCD plane. A schematic is presented in Figure 1.13.

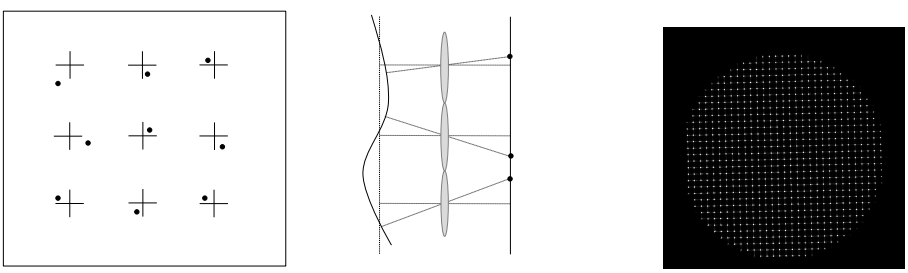


Figure 1.13: Left: schematic representation of a SH two-dimensional array projecting the wavefront onto a CCD plane. The crosses indicate the reference location of centroids. The dots correspond to the measured centroids when the wavefront is not flat. Middle: one-dimensional view of an abberrated wavefront with lenslets focusing the local wavefront on their back focal plane. The local displacements are measured with respect to the reference obtained when the wavefront is plane. Right: reading of a medium-size Shack-Hartmann sensor used in the laboratory testbed. Each white dot represents a reference position for the centroids.

The SH sensor measures the local gradients averaged over the respective subaperture which is defined by corner coordinates $\{(x_i, y_i), (x_{i+1}, y_i), (x_i, y_{i+1}), (x_{i+1}, y_{i+1})\}$ in

the horizontal and vertical directions,

$$\left\{\begin{array}{rcl} s_{x_{i,j}}(k) & = & \frac{1}{T_s}\int_{kT_s}^{(k+1)T_s}\alpha_x\Big(\int_{y_i}^{y_{i+1}}\phi_{t_{x_i,y}}dy-\int_{y_i}^{y_{i+1}}\phi_{t_{x_{i+1},y}}dy\Big)dt+\eta_{x_{i,j}}(k)\\ s_{y_{i,j}}(k) & = & \frac{1}{T_s}\int_{kT_s}^{(k+1)T_s}\alpha_y\Big(\int_{x_i}^{x_{i+1}}\phi_{t_{x,y_i}}dx-\int_{x_i}^{x_{i+1}}\phi_{t_{x,y_{i+1}}}dx\Big)dt+\eta_{y_{i,j}}(k)\end{array}\right. \tag{1.24}$$

where α_x, α_y are geometrical properties and $\boldsymbol{\eta}_x, \boldsymbol{\eta}_y$ such that $\boldsymbol{\eta}(k) = \begin{bmatrix}\boldsymbol{\eta}_x(k)^T & \boldsymbol{\eta}_y(k)^T\end{bmatrix}^T$ is a zero-mean white noise with covariance matrix $\sigma_\eta^2\mathbf{I}$. The measurements are integrated over the sampling period T_s (assumed equal to the exposure time). The additive noise $\boldsymbol{\eta}$ on the wavefront sensor accounts for Poisson noise due to the arrival of photons on the camera, the read-out and thermal noise, non-linearities in the sensor among which the spatial discretization of the CCD with pixels and discretization of CCD intensity values. It is however approximated as zero-mean white, Gaussian and stationnary. All channels are uncorrelated.

The equation (1.24) is rewritten between the discretized wavefront $\boldsymbol{\phi}(k)$ and $\mathbf{s}(k)$ as follows,

$$\mathbf{s}(k+1) = \mathbf{G}\boldsymbol{\phi}(k) + \boldsymbol{\eta}(k) \tag{1.25}$$

A delay of one step is assumed in (1.25) to account for the time required for collecting photons and reading out the frames from the camera. The measurement matrix $\mathbf{G}$ is rank-deficient due to the presence of unseen modes by the sensor such as the piston and waffle modes. The piston mode does not affect the image quality on the PSF and the waffle mode corresponds to a very large spatial frequency, Roddier (2004). The Shack-Hartmann is blind to the frequencies above the Nyquist limit, although the power spectral density of the wavefront decreases as a function of the spatial frequency.

Performance measures

AO systems strive to recover an image of a faint star as close to diffraction limit as possible which occurs when the wavefront is constant over the telescope aperture as shown in (1.15). In closed-loop, the residual wavefront $\boldsymbol{\epsilon}_t$ is,

$$\boldsymbol{\epsilon}_t(\boldsymbol{\rho}) = \boldsymbol{\phi}_t^{tur}(\boldsymbol{\rho}) + \boldsymbol{\phi}_t^{m}(\boldsymbol{\rho}) \tag{1.26}$$

where the wavefront induced by the turbulence only is denoted with $\boldsymbol{\phi}_t^{tur}$. The variance of the residual wavefront over the pupil aperture and averaged over infinitely long exposures is then,

$$\sigma_\epsilon^2 = \lim_{T_s\to\infty}\frac{1}{T_s}\int_0^{T_s}\Big(\iint_{\boldsymbol{\rho}\in\Omega}\boldsymbol{\epsilon}_t(\boldsymbol{\rho})^2 d\boldsymbol{\rho} - \Big(\iint_{\boldsymbol{\rho}\in\Omega}\boldsymbol{\epsilon}_t(\boldsymbol{\rho})d\boldsymbol{\rho}\Big)^2\Big)dt \tag{1.27}$$

The quality of the imaging system is also evaluated in terms of sharpness of the long-exposure PSF. As the optical flux that reaches the PSF camera is identical whatever the aberration, the more concentrated over a central core the intensities are, the better. The Strehl ratio S is defined as the maximum intensity of the PSF divided by the one of the diffraction-limited PSF. It is shown in Herrmann (1992) that maximizing S is equivalent to minimizing the Euclidean norm of the residual

wavefront. When the total variance of the wavefront σ_ϵ^2 is smaller than 1 rad^2, the Maréchal approximation relates both quantities with $S = e^{-\sigma_\epsilon^2}$.

The normalized encircled energy measures the flux that enters within a circle of radius r centered around the position of the maximum value, $\mathbf{p}_0$, in the PSF image I,

$$EE(r) = \frac{\lim_{T_s\to\infty} \frac{1}{T_s} \int_0^{T_s} \iint_{\boldsymbol{\rho}\in\mathcal{B}_r(\mathbf{p}_0)} I(\boldsymbol{\rho},t) d\boldsymbol{\rho} dt}{\lim_{T_s\to\infty} \frac{1}{T_s} \int_0^{T_s} \iint_{\boldsymbol{\rho}\in\mathcal{I}} I(\boldsymbol{\rho},t) d\boldsymbol{\rho} dt} \tag{1.28}$$

where $\mathcal{B}_r(\mathbf{p}_0) = \{\mathbf{p} \in \mathcal{I} : \|\mathbf{p} - \mathbf{p}_0\|_2 < r\}$ and $\mathcal{I}$ is the set of coordinates in the image.

Other measures include full-width at half-maximum and also the Power Spectrum Density of the residual PSF (and its Euclidean norm).

1.4.3. Control for large-scale AO

The wavefronts reaching both the PSF and SH cameras are theoretically identical. In practice, however, optical alignment errors are such that there exists a time-invariant difference between both of them and that are called non-common path aberrations. These are assumed negligible. Removing the spatial mean of the residual wavefront, we isolate the contribution that does not depend on the control input in σ_ϵ^2,

$$\sigma_\epsilon^2 = \lim_{T_s\to\infty} \frac{1}{T_s} \int_0^{T_s} \|\boldsymbol{\phi}_t^{tur}\|_2^2 + \lim_{K\to\infty} \frac{1}{K} \sum_{k=0}^{K-1} \left(\|\mathbf{H}\mathbf{u}(k)\|_2^2 + \boldsymbol{\phi}^{tur}(k+1)^T \mathbf{H}\mathbf{u}(k) \right) \tag{1.29}$$

where $\boldsymbol{\phi}^{tur}(k+1) = \frac{1}{T_s} \int_{kT_s}^{(k+1)T_s} \boldsymbol{\phi}_t^{tur} dt$. The performance criterion in continuous time is rewritten into a discrete version by discarding the first term on the right hand side of (1.29) which does not depend on the input and adding the discrete counterpart, Kulcsár et al. (2012),

$$\sigma_{\epsilon,d}^2 = \lim_{K\to\infty} \frac{1}{K} \sum_{k=0}^{K-1} \|\boldsymbol{\phi}^{tur}(k+1) + \mathbf{H}\mathbf{u}(k)\|_2^2 \tag{1.30}$$

An additional error appears because of the zero-th order hold for sending the mirror inputs: the commands are set to a fixed voltage while the turbulence keeps on evolving during a sampling period. This error is called inter-sample variance. Increasing the sampling frequency to infinity sets the inter-sample variance to 0.

Minimum variance control in AO boils down to first, deriving a prediction for the future wavefront and second, projecting the latter onto the actuator space, as written first in Roux et al. (2004), then in Kulcsár et al. (2006). The separation principle states that the two stages for a disturbance rejection problem can be performed independently. We first focus on the Linear-Quadratic Regulator (LQR) problem for which solving a Riccati equation is not necessary thanks to the linear static model in (1.22). The cost function makes a trade-off between minimizing the residual wavefront and the control effort,

$$\min_{\mathbf{u}(k)} \quad \|\widehat{\boldsymbol{\phi}^{tur}}(k+1|k) + \mathbf{H}\mathbf{u}(k)\|_2^2 + \mathbf{u}(k)^T \mathbf{Q}\mathbf{u}(k) \tag{1.31}$$

where $\mathbf{Q}$ is a semi-positive definite matrix, and $\widehat{\boldsymbol{\phi}^{tur}}(k+1|k)$ is the estimate of $\boldsymbol{\phi}^{tur}(k+1)$ which is derived using all the data up to time k. The least-squares may include bound inequalities to constraint the input within the linear range of the DM such that all the actuators are not penalized with the quadratic term in (1.31). When the coupling between actuators is localized such that the matrix $\mathbf{H}$ is sparse and the matrix $\mathbf{Q}$ is diagonal, solving (1.31) is a large though sparse least-squares.

The temporal error and the frozen flow assumption

The temporal error is related to the time delay between measuring the slopes and actually applying the control inputs. The turbulence layers above the telescope are driven by the wind and therefore, the control commands become quickly outdated if the wavefront aberrations have significantly evolved during a sampling period. The mean-square wavefront distortion due to the time delay evolves with,

$$\sigma^2_{\text{temp}} = \Big(\frac{\bar{v}}{r_0 f_S}\Big)^{5/3} \tag{1.32}$$

where the control frequency is f_S, and the overall windspeed of the screen is defined as,

$$\bar{v} = \Big(\frac{\int C_n^2(z)|v(z)|^{5/3}dz}{\int C_n^2(z)dz}\Big)^{3/5} \tag{1.33}$$

$C_n(z)$ is the refractive index structure coefficient characterizing the turbulence strength at height z, Hardy (1998). The faster the turbulence moves over the telescope aperture, the larger the temporal error with a standard Proportional-Integral controller. Fried (1990) introduces the Greenwood frequency to quantify the wavefront errors as a function of key characteristics of the turbulence,

$$f_G = 0.427\frac{\bar{v}}{r_0} \tag{1.34}$$

Plugging (1.34) into (1.32) is such that the mean-square wavefront distorition, σ^2_{temp}, is proportional to $(f_G/f_S)^{5/3}$. More accurate models for estimating $\phi^{tur}(k+1)$ are needed for large Greenwood per sample frequency ratio and are motivated by this relation. These are discussed in a next paragraph.

As a summary, the errors in the AO system are due to the sensor noise (including photon noise), the temporal dynamics of the turbulence, the aliasing error related to the WFS, the non-common path aberrations, the fitting error and the intersampling error. The latter is considered negligible for systems running at frequencies larger than 100Hz, Kulcsár et al. (2012).

1.4.4. Turbulence prediction

Motivation and challenges

The temporal error has two main sources, namely the dynamics of the turbulence and the vibrations of the telescope. This thesis studies more particularly the former.

When there are few actuators and sensors and the deformable mirror is such that the first resonance frequencies appears well above the kilohertz, practitioners

usually increase the control bandwidth as much as the WFS camera allows. The drawback is nonetheless that less photons are collected and the signal to noise ratio decreases especially if the guide star is not bright enough. Moreover, the design of the mirror M4 in the ELT with about 8000 actuators and a first resonant frequency at 600Hz does not permit to lessen the nefarious effect of the temporal error this way and predictive control algorithms should be used in combination with an efficient implementation.

We denote $\mathbf{s}^{tur}(k)$ the so-called pseudo open-loop measurements obtained by subtracting the influence of the previous input on the residual disturbance, $\mathbf{s}^{tur}(k) = \mathbf{s}(k) - \mathbf{B}\mathbf{u}(k-1)$. The most general form for modelling the open-loop temporal dynamics of the disturbance is a stochastic state-space model,

$$\left\{ \begin{array}{lcl} \mathbf{x}(k+1) & = & \bar{\mathbf{A}}\mathbf{x}(k) + \mathbf{w}(k) \\ \boldsymbol{\phi}^{tur}(k) & = & \mathbf{C}_d\mathbf{x}(k) + \mathbf{v}(k) \\ \mathbf{s}^{tur}(k+1) & = & \bar{\mathbf{C}}_s\mathbf{x}(k) + \boldsymbol{\eta}(k) \end{array} \right. \tag{1.35}$$

where $\mathbf{w}(k), \mathbf{v}(k), \boldsymbol{\eta}(k)$ are zero-mean white Gaussian noises, and with covariance matrix respectively $\mathbf{C}_w, \mathbf{C}_v, \mathbf{C}_\eta$. The latter is assumed diagonal.

When the state is assumed equal to the open-loop wavefront, prior knowledge of the system matrices such as the measurement matrix $\mathbf{G}$ can be used. Equation (1.35) is then rewritten into,

$$\left\{ \begin{array}{lcl} \boldsymbol{\phi}^{tur}(k+1) & = & \mathbf{A}\boldsymbol{\phi}^{tur}(k) + \mathbf{w}(k) \\ \mathbf{y}^{tur}(k) & = & \mathbf{G}\boldsymbol{\phi}^{tur}(k) + \boldsymbol{\eta}(k) \end{array} \right. \tag{1.36}$$

where the shifted output $\mathbf{y}^{tur}(k) = \mathbf{s}^{tur}(k+1)$ is introduced to take into account the time delay in (1.25). Once the matrices $\mathbf{A}$ and $\mathbf{C}_w$ are estimated, there remains to solve in a scalable manner the Riccati equation.

The main challenges are twofold. First, deriving offline (i.e not being constrained by the control frequency imposed for real-time operation of the system) though efficiently the Kalman gain $\bar{\mathbf{K}}$ either from data or solving the Riccati equation *and*, second, in a structured manner to pave the way for efficient online computations. The methods derived in the literature mainly depend on the temporal order chosen for the wavefront update and the parametrization chosen for the matrix $\mathbf{A}$.

Unstructured approaches for medium-size AO

When the sampling time of the AO loop is much smaller than the lifetime of the wind-blown inhomogeneities, it is common to assume a frozen flow propagation over the telescope such that the future wavefront is merely a shifted version of the current one, Roddier (2004),

$$\boldsymbol{\phi}^{tur}_{t+T_s}(\boldsymbol{\rho}) = \boldsymbol{\phi}^{tur}_{t}(\boldsymbol{\rho} - \bar{v}T_s) \tag{1.37}$$

Under the frozen-flow assumption, Gavel and Wiberg (2003) introduced the near-Markov approximation to carry out the time update, the conditional expectation for the wavefront at a time instant requires only the knowledge of the previous instance when only a small fraction of the telescope aperture is crossed during a sampling period. The matrix $\mathbf{A}$ is estimated using the phase covariance matrix

$C_{\phi,0} = \mathbb{E}\left[\phi(k)\phi(k)^T\right]$ and $C_{\phi,1} = \mathbb{E}\left[\phi(k)\phi(k-1)^T\right]$, computed from the piston-removed Kolomogorov turbulence in Gavel and Wiberg (2003) and Piatrou and Roggemann (2007) in (1.20) via $\mathbf{A} = \mathbf{C}_{\phi,1}\mathbf{C}_{\phi,0}^{-1}$. The near-Markov assumption was used in Piatrou and Roggemann (2007) to highlight the increased performances of a Kalman filter-based control algorithm with respect to the static reconstruction.

Data-driven methods were studied in Beghi et al. (2008), Hinnen et al. (2007) and Guyon and Males (2017). In Hinnen et al. (2007), a $\mathcal{H}_2$-optimal control was proposed and the full spatial-temporal dynamics in (1.36) were identified from open-loop data in the most general state-space form using a subspace algorithm. This approach moreover removes the unobservable wavefront modes in the sensor measurements in order to decrease the output dimension although it is not sufficient to scale to larger arrays. While the temporal error follows the trend in (1.32), a data-driven $\mathcal{H}_2$ optimal control improves the performances all the more as the Greenwood per sample frequency ratio is large, see Figure 1.14.

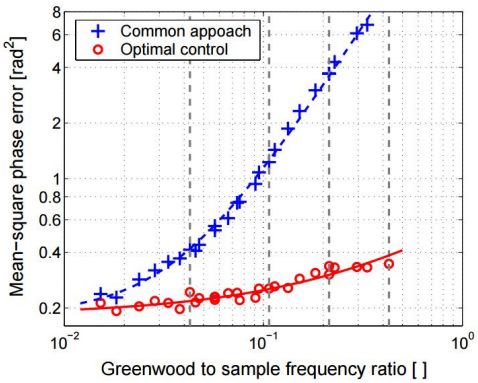


Figure 1.14: The plot corresponds to laboratory experiments presented in Hinnen (2007) and represents the mean-square phase error as a function of the Greenwood per sample frequency ratio. The blue curve assumes $\widehat{\phi^{tur}}(k+1|k) = \phi^{tur}(k)$ whereas the optimal control in red predicts the wavefront using a data-driven approach identifying a stochastic state-space model.

Instead of coping with the full generality of an infinite impulse response as in Hinnen et al. (2007), Guyon and Males (2017) assume an autoregressive model for modeling the temporal dynamics of the wavefront data and estimate the coefficient matrices with a truncated Singular Value Decomposition.

Although reaching promising performances in terms of disturbance rejection, neither of these methods is able to handle the large number of actuators and sensors in the next generation of extremely large telescopes. With a sampling grid of size $N \times N$, the identification algorithms in Hinnen et al. (2007) and Guyon and Males (2017) scale with an order of N^6 when estimating for the former the state space matrices including the Kalman gain, and for the latter, the coefficient matrices of the AR model. This restriction stems from the high computational cost associated with system identification methods for large N when no a priori information on the

spatial-temporal behaviour of the disturbance is available (or postulated).

Toward prediction for large-scale AO

Solving the discrete algebraic Riccati equation Although dealing with general unstructured coefficient-matrices in the state-space model (1.36) yields the most accurate representation, it does not enable to derive efficient control algorithms when the sensor grid is large and was therefore simplified for scalability purposes in a number of works that we now revisit. The simplest model for describing the temporal dynamics of the turbulence is the random-walk,

$$\boldsymbol{\phi}^{tur}(k+1) = \boldsymbol{\phi}^{tur}(k) + \mathbf{w}(k) \tag{1.38}$$

The predicted wavefront at time instant $k+1$ is then equal to the wavefront at time instant k, the latter being efficiently reconstructed from sensor measurements solving a stochastic least squares and exploiting the sparse pattern of the inverse covariance matrix of the wavefront, Ellerbroek (2002). Diagonal autoregressive model of temporal order one have been widely used, either in a zonal or Zernike basis,

$$\boldsymbol{\phi}^{tur}(k+1) = \mathbf{A}\boldsymbol{\phi}^{tur}(k) + \mathbf{w}(k) \tag{1.39}$$

When the matrix $\mathbf{A}$ is diagonal, it is through the full covariances of the wavefront that the spatial correlations are taken into account. The assumption $A = aI$ has been used in few works as a first step to solve in a scalable manner the DARE. The coefficient a, usually determined from the wind speed and the control frequency, plays the role of a forgetting factor, a coefficient equal to one reflects a frozen turbulence whereas the closer to zero, the more temporally varying the wavefront is between two time samples.

Such model for apprehending the temporal correlations of large systems is considered in Correia et al. (2010) which exploits sparsity and in Massioni et al. (2011) where a distributed control approach is investigated. The latter approximates the phase screen over the aperture as a cropped version of an infinitely long screen, so that the state equation is diagonal in the Fourier domain. The DARE is then solved in parallel for all frequencies, and transformed back into the real domain using the inverse Fourier transform. The online control does thus not involve any Fourier transform. Following Bamieh et al. (2002), the influence of the neighbours decays with the distance and the Kalman filter is localized, however to an extent that has not been quantified. This limitation is essentially revealed with the finite-size of the sensor, and the edges in the telescope aperture. This method is reminiscent of the developments on decomposable and circulant systems mentioned in the paragraph 1.2.2. The optimal Kalman filter is nonetheless spatially varying because of the finite size of the sensor. Gilles et al. (2013) compare the distributed Kalman filtering approach in Massioni et al. (2011) with a Fourier-based tomographic reconstruction in Multi-Conjugate AO and show its improved performances when the wind speed and direction are known. The latter are estimated in Massioni et al. (2015) and used for updating the Kalman filter.

This Fourier-based approach differs from the one in Poyneer et al. (2007) where the control relies on a wavefront reconstruction from a Fourier-transformed vector of

measurements (or more precisely, the extension of the latter on a square periodic grid rather than circular). The restrictive assumption in modelling the VAR model with a diagonal matrix has been released, e.g in Gilles et al. (2013) where a shift matrix is used, and a VAR with a second order in time is used in Sivo et al. (2014) along with a representation of the wavefront in a Zernike basis.

As a summary, the shortcomings of the autoregressive models that have been postulated for predicting the future wavefront in large-scale AO are first, the restricted spatio-temporal coupling assumed, and second, the use of first principles to determine the coefficients. Deriving the models from data would allow to overcome the problem of estimating prior information such as wind speed, Fried parameter, etc, which are likely to evolve during an observing run, and to strive for more accuracy in expressing the spatio-temporal dynamics.

When the control frequency is fast enough with respect to the frequency at which the turbulence evolves, assuming a sparse structure on the coefficient matrices of the AR model is relevant and has been investigated in the works we now mention. Yu and Verhaegen (2018b) bridge a gap by parametrizing $\mathbf{A}$ in (1.36) with a Kronecker product of banded matrices, which is then used for deriving a sparse dynamical controller. The SSS structure, discussed in the paragraph 1.2.3, is used in Fraanje et al. (2010) and uses a model derived from Beghi et al. (2008) which strongly relies on the frozen flow assumption. The phase screen is decomposed into columns that are shifted in subsequent time samples. The two-dimensional disturbance is recast as one-dimensional when the wind speed and direction is known.

Data-driven approaches One data-driven approach for large-scale AO which does not rely on solving a Riccati equation is Piscaer (2016). This method identifies a sparse banded VARX model from open-loop slopes data and random mirror inputs. It is however still unknown how simulatenously identifying in open-loop the turbulence and the mirror dynamics would translate in closed-loop on real systems. The approach in Yu and Verhaegen (2018a) has not been applied to AO but is a potential candidate for deriving a sparse Kalman gain from data (the grid of lenslets is partitioned into a set of subsystems). The state observer then only receive the output information from its four closest neighbours.

Online updates
The turbulence dynamics are stationary over relatively short time period. It holds as an approximation only for longer exposures and are very much likely to evolve during the observation run, and the control law (and in this case, the model used for prediction) should adapt to the different turbulence scenarios flowing over the telescope aperture. Ellerbroek and Rhoadarmer (2001) derive a recursive least-squares algorithm for medium-size AO. Updating the prediction model from data has not been studied in the context of large-scale adaptive optics.

1.4.5. Research question

The only data-driven and scalable algorithm that has been proposed so far is proposed in Piscaer (2016). One shortcoming is that the coefficient matrices are not necessarily

banded for various values of the Greenwood per sample frequency ratio, especially for varying wind conditions for which the fixed bandwidth pattern for the non-zero entries should vary to adapt. This makes such sparse pattern unpractical for updating online the model from data. Moreover, solving the Riccati equation corresponding to the stochastic model (1.36) when $\mathbf{A}$ is unstructured is untractable and the alternatives have assumed a diagonal parametrization. Is it possible to handle more general patterns of $\mathbf{A}$ with minor impact on the computational cost? What is a dense data-sparse parametrization $\mathbf{A}$ and such that the structure is preserved when e.g solving the DARE with the Newton's iterations?

We propose the following question,

To what extent do the identification algorithms proposed in addressing the research question of Section 1.3 handles the balance between computational complexity and data storage, and minimizing the temporal error?

We validate the algorithms derived in this thesis by applying them on the basis of numerical simulations and laboratory experiments. In other words, we would like to place another point in the plot accuracy versus scalability by studying another matrix structure suited to the two-dimensional sensor array.

1.5. Research direction and main contributions

In this dissertation, we introduce the class of low-Kronecker rank matrices $\mathcal{K}_{d,r}$ defined as the set containing all the matrices $\mathbf{X} \in \mathbb{R}^{J \times I}$, such that

$$\mathbf{X} = \sum_{j=1}^{r} \mathbf{X}_{d,j} \otimes \ldots \otimes \mathbf{X}_{1,j}, \quad r \ll \min(\{J_i, I_i\}_{i=1..d}) \tag{1.40}$$

where $\mathbf{X}_{i,j} \in \mathbb{R}^{J_i \times I_i}$ and $J = \prod_{i=1}^{d} J_i, I = \prod_{i=1}^{d} I_i$. The parametrization is not affine in the matrices $\{\mathbf{X}_{i,j}\}_{i=1..d,j=1..r}$ also known as factor parameters. Such data-sparse representation for possibly dense large matrices relies on a certain separability approximation for the underlying multi-dimensional function describing the spatial dynamics. A main idea that is carried out throughout the thesis is that a large-scale convex problem is transformed into a multi-convex optimization with a reduced number of variables. An optimization is said to be multi-convex when the variables can be partitioned into sets, each of them such that the cost function is convex when all other variables are fixed. Three methods for computing a prediction are studied under the light of this new parametrization: identifying Vector AutoRegressive models, identifying stochastic state-space models, and computing a Kalman gain from the DARE.

For sensor arrays distributed on a grid of size $N \times N$, the coefficient matrices of AutoRegressive models are parametrized with low-Kronecker rank matrices, and are identified with $\mathcal{O}(N^3 N_t)$ computational complexity, where N_t is the number of temporal samples, instead of $\mathcal{O}(N^6)$ in the unstructured case. An Alternating Least Squares is proposed which was empirically shown to converge to a global minimum. The global convergence is essential in the thesis as it guarantees that the

residual error is implied by a discrepancy between the actual model structure and the Kronecker decomposition, and not by the optimization itself. Such data-sparse representation is moreover not only useful for more scalable linear algebra but also for reducing the length of the data batch used for identification as the number of unknowns now scales linearly with the number of sensor outputs. Regularization on the factor matrices is proposed to handle noisy and short data-batches. A recursive algorithm is derived to update the factor matrices in order to first, account for possibly time-varying disturbances and second, to further decrease the memory storage compared to the case when batches of data are stored.

We introduce state-space models with matrices in $\mathcal{K}_{2,1}$ and formulate a matrix state-space model where the states, input and output are matrices rather than vectors. Results on stability, observability and controllability are derived to characterize this class of system. This formulation allows the derivation of an identification algorithm of state-space matrices with $\mathcal{O}(N^3 N_t)$. The non-uniqueness of the estimates obtained from ALS on a Finite-Impulse Response model hampers the direct use of standard realization theory, and as a consequence, we have formulated a low-rank optimization subject to bilinear constraints as a step toward the identification of the state-sequence. This method has shown significant improvements in the computational complexity at the expense of lower accuracy due to a non-globally convergent behaviour of the proposed Block-Coordinate Update algorithm. The algorithm is derived for deterministic only state-space systems although a numerical experiment carried out on a stochastic system highlights the applicability to models in innovation form under certain conditions. Whereas most of these results are first presented with a product of two matrices with a single Kronecker product, the framework is extended to account for more spatial dimensions or to further compress the data (for fixed I, J, less parameters are stored in the factor matrices for increasing d). State-space models where the input, state and output are tensors are introduced. The algorithm previously proposed is simplified and importantly, a subclass of $\mathcal{K}_{d,1}$ is introduced. This subclass is composed of systems whose factored Markov parameters are strictly positive element-wise. Such assumption allows to significantly improve the performances both in terms of accuracy and computational cost.

Inspired by the structure-preserving iterations for deriving efficient (and structured) solutions for the Lyapunov and Riccati equation in Rice (2010), we show that standard linear algebra operations can be computed with $\mathcal{O}(N^3)$ complexity instead of $\mathcal{O}(N^6)$ for matrices written as sums of Kronecker products between two matrices. This paves the way for e.g doubling algorithms used for computing the solution of discrete-time Lyapunov equations while maintaining the low-Kronecker rank structure throughout the iterations hence allowing efficient computations. An alternative is derived in close connection with the well-studied case where the semi-positive definite matrix in the right-hand side is low-rank implying (as observed empirically) a low-rank solution. We adapt a state-of-the-art factored Alternating Direction Implicit method to the case when the matrices are Kronecker-structured.

The tensor models are used in the context of adaptive optics to derive a minimum-variance unbiased estimate of the turbulence-induced slopes. We formulate a tensor-based autoregressive model on the slopes data which are identified in open-

loop and with an algorithm that approaches linear complexity with respect to the number of Shack-Hartmann lenslets. It is shown on a validation dataset collected in open-loop as well as in closed-loop the improved performances with respect to e.g the diagonal or sparse banded assumptions especially for large Greenwood per sample frequency ratio. Although the coefficient matrices are allowed to be dense, computational rules relative to tensors enable to compute the prediction with $\mathcal{O}(N^{\frac{2(d+1)}{d}})$. The approach is validated on a laboratory testbed and it demonstrates the decrease of the temporal error over standard non-predictive methods.

1.6. Outline of the thesis

The remaining chapters of the thesis are structured as follows. Recommendations for future work are written at the end of each chapter when specific and presented in the concluding Chapter 7 for broader aspects.

Part I. Chapter 2 - Identifying Kronecker-structured AutoRegressive models
In this chapter, we introduce autoregressive models with low-Kronecker rank coefficient matrices and propose scalable identification algorithms, one for dealing with batches of stationnary data, and a recursive variant for handling non-stationarities. I acknowledge Guido Monchen for his contribution on the recursive algorithm. The material in this chapter stems from,

B. Sinquin, M. Verhaegen, "QUARKS, Identification of Kronecker Vector AutoRegressive models", in *IEEE Transactions on Automatic Control*, vol. 64, no. 2, pp.448-463, 2019.

G. Monchen, B. Sinquin, M. Verhaegen, "Recursive Kronecker-Based Vector Autoregressive Identification for Large-Scale Adaptive Optics", in *IEEE Transactions on Control Systems Technology*, 2018.

Part I. Chapter 3 - Identifying Kronecker-structured state-space models
In this chapter is considered the identification of deterministic state-space models when the state-space matrices are parametrized with a single Kronecker product. It is published in,

B. Sinquin, M. Verhaegen, "K4SID, Large-Scale Subspace Identification with Kronecker modeling", in *IEEE Transactions on Automatic Control*, vol. 64, no. 3, pp. 960-975, 2019.

Part I. Chapter 4 - Scaling up
The results derived in the chapters 2 and 3 are generalized to more spatial dimensions and tensor state-space models are introduced. The state is no longer a vector but a tensor. We focus on the particular case where the systems have a strictly externally positive impulse response for improved performances. This assumption was suggested by Prof. Hansson. This work has not been previously published and appears for the first time in this thesis.

Part I. Chapter 5 - Kronecker-structured discrete Lyapunov equation
Whether the low-Kronecker rank structure is preserved when adding, multiplying and truncating low-Kronecker rank matrices is of prime importance for deriving scalable iterative algorithms (e.g for solving the DARE). It is investigated in this chapter. Moreover, we propose to compute the discrete-time Lyapunov equation using iterative algorithms without forming the large-scale matrices but rather exploiting the low-Kronecker rank structure, which alleviates the computational load. This work appears for the first time in this thesis.

Part II. Chapter 6 - Tensor-based predictive control for large-scale AO and experimental validation
An auto-regressive model with a low-Kronecker rank parametrization is used for deriving a prediction of the disturbance. The approach is validated on a laboratory testbed dedicated for large-scale AO.

I acknowledge Maarten Griffioen and Will van Geest for writing the software to drive from C the turbulence disks, the cameras, and deformable mirror; and a GPU code for computing the sensor outputs from the raw images, and Dr. Oleg Soloviev for the interesting discussions.
Parts of this chapter were published in,

B. Sinquin, M. Verhaegen, "Tensor-based predictive control for extremely large-scale adaptive optics", in *J. Opt. Soc. Am. A* 35, 1612-1626 (2018).

Chapter 7 - Conclusion and recommendations
In this concluding chapter are summarized the main findings along with further research questions to deepen the understanding of the class of low-Kronecker rank structures for controlling multi-dimensional linear dynamical systems.

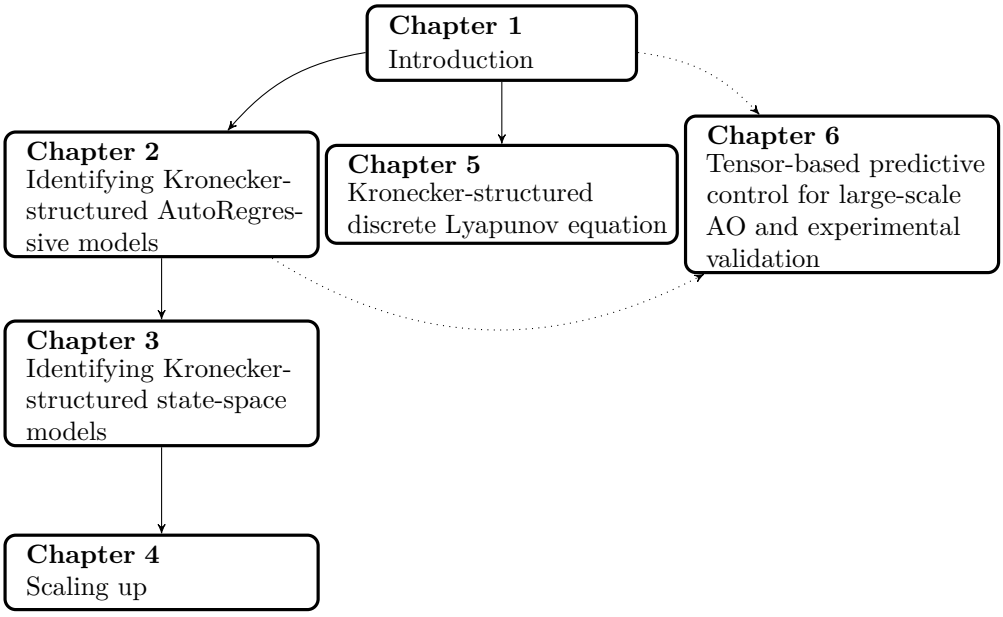
Appendix
We review the most important properties of the Kronecker product in Appendix A and the fundaments on tensors in Appendix B.

Matlab toolbox
The Matlab codes related to the algorithms presented in the chapters 2, 3, 4 and 5 are available in a toolbox on the Bitbucket repository,

`https://bitbucket.org/csi-dcsc/t4sid.git`

It contains moreover Matlab code written by Guido Monchen and Peter Varnai while studying their MSc thesis. I acknowledge Guido Monchen for writing the code related to the Recursive Least Squares using a QUARKS modeling and the CUDA code for solving the QUARKS with a sum-of-Kronecker parametrization. Peter Varnai contributed by writing the Matlab code for approximating the inverse of low-Kronecker rank matrices and solving large box-constrained least squares exploiting this same structure.

The layout of the thesis is presented in the flow chart below. The reader most interested in the theoretical developments should follow the paths with solid lines whereas the dashed part focuses on the application to AO.



2
Identifying Kronecker-structured auto-regressive models

In this chapter, we address the identification of two-dimensional spatial-temporal dynamical systems described by the Vector Auto-Regressive (VAR) form. The coefficient matrices of the VAR model are parametrized as sums of Kronecker products. When the number of terms in the sum is small compared to the size of the matrices, such a Kronecker representation efficiently models large-scale VAR models.

Estimating the coefficient matrices in least-squares sense gives rise to a bilinear estimation problem which is tackled using an Alternating Least Squares (ALS) algorithm. Regularization or parameter constraints on the coefficient matrices allows inducing temporal properties such as stability as well as spatial ones such as sparsity or e.g Toeplitz structure. The estimates of a particular formulation of ALS which features some normalization converge to a fixed point. A numerical example demonstrates the advantages of the new modeling paradigm. It leads to comparable variance of the prediction error with the unstructured least-squares estimation of VAR models. However, the number of parameters grows only linearly with respect to the number of nodes in the 2D sensor network instead of quadratically in the case of fully unstructured coefficient matrices.

A recursive variant of the Kronecker-structured autoregressive models is then proposed. A validation on non-stationary atmospheric turbulence data, both synthetic and experimental, is shown for an adaptive optics application. Significant improvements in accuracy over batch identification methods that assume stationarity are observed while both the computational complexity and the required storage are reduced.

Section 2.1 to 2.5 and 2.7 have been published in:
B. Sinquin and M. Verhaegen, "QUARKS: Identification of Large-Scale Kronecker Vector-AutoRegressive Models," in *IEEE Transactions on Automatic Control*, vol. 64, no. 2, pp.448-463, 2019.
Sections 2.6 and 2.8 have previously appeared in:
G. Monchen, B. Sinquin, M. Verhaegen, "Recursive Kronecker-Based Vector Autoregressive Identification for Large-Scale Adaptive Optics", as a brief paper in *IEEE Transactions on Control Systems Technology*, 2018.

2.1. Introduction

A novel modeling paradigm is introduced to identify 2D spatial systems with temporal dynamics. As a fundament of this new approach, we restrict to temporal Vector Auto-Regressive models of temporal order p with the spatial structure imposed on the coefficient matrices. Let the sensor measurements be distributed on a grid of size $N \times N$. The spatial structure is embedded into the coefficient matrices $\{\mathbf{A}_i\}_{i=1..p}$, parametrized with a finite sum of a Kronecker product between low dimensional matrices:

$$\mathbf{A}_i = \sum_{i=1}^{r} \mathbf{U}_i \otimes \mathbf{V}_i \tag{2.1}$$

where r is called the Kronecker rank and $\mathbf{U}_i, \mathbf{V}_i \in \mathbb{R}^{N \times N}$ are the factor matrices. Such representation of large matrices was studied in van Loan and Pitsianis (1993) in which the equivalence between expressing a matrix as a sum of r Kronecker products and a rank-r approximation of a reshuffled matrix was established. Therefore, any matrix admits such a decomposition by just fixing r to the adequate value. We are especially interested in the case where r is much smaller than N.

More than only enjoying the storage of $2rN^2$ entries instead of N^4, such a structure enables fast computations thanks to the very pleasant algebra of the Kronecker product, see e.g van Loan (2000). Using Kronecker structures for solving Partial Differential Equations stemming from multi-dimensional problems is well-known, Grasedyck et al. (2013). Besides, Kronecker structures have been applied efficiently for computing second moments in multi-dimensional processes, Tsiligkaridis and Hero (2013), for analyzing EEG signals, Bijma et al. (2005), and for image deblurring, Hansen et al. (2006). The latter example enables to relate the Kronecker rank-one modeling with physical properties of the system. Denoting an object $\mathbf{O}$ imaged with a static optical system, the resulting blurred image $\mathbf{B}$ undergoes the linear blurring operation,

$$\text{vec}(\mathbf{B}) = \mathbf{A}\text{vec}(\mathbf{O}) \tag{2.2}$$

The equation (2.2) represents the 2D convolution operation between the PSF and the object $\mathbf{O}$. The structure in $\mathbf{A}$ is related to the separability of the PSF in both horizontal and vertical directions which implies the following Kronecker structure for the coefficient matrix $\mathbf{A}$:

$$\mathbf{A} = \mathbf{A}_r \otimes \mathbf{A}_c \tag{2.3}$$

where $\mathbf{A}_r$ and $\mathbf{A}_c$ represent respectively the 1D convolution with the rows and columns. A large-scale static input-output map in (2.2) is represented by a matrix parametrized with a Kronecker product, (2.3). In a more general context, separation-of-variable techniques have been applied in Doostan and Iaccarino (2009) and the references therein to break down the curse of dimensionality when modeling high-dimensional partial differential equations.

Although tensor-based algorithms for handling large datasets receive a growing interest, system identification of multi-dimensional systems is however in its infancy. An overview of data-driven algorithms that handle large datasets using the tensor representation is provided in Cichocki et al. (2017) and includes a multi-linear tensor

regression for relational longitudinal data, Hoff (2015). The approach proposed in Hoff (2015) handles the estimation of factor matrices from an input-output tensor model and using Alternating Least Squares. However, it embeds temporal dynamics in a higher-order tensor whereas the parametrization we propose follows the control engineering approach to combine the temporal dynamics linearly while modeling independently each coefficient matrix with a sum of Kronecker matrices. Besides, we allow the Kronecker rank to be strictly larger than one for more generality and applicability for identification and control of systems such as adaptive optics. These two points are crucial to achieve good accuracy estimations in e.g a laboratory environment and hence, enable its effective use for control. Third, regularization to estimate stable and sparse models is proposed.

Another work related to the framework we propose deals with blind source separation using tensor representations, Boussé et al. (2017a) and Boussé et al. (2017b). The approach consists in estimating two matrices $\mathbf{M}$ and $\mathbf{S}$ from the measurements stored in $\mathbf{X}$ given the relationship:

$$\mathbf{X} = \mathbf{MS} \tag{2.4}$$

where $\mathbf{M}$ represents the mixing matrix and $\mathbf{S} \in \mathbb{R}^{n \times K}$ the n source signals for K time samples. Boussé et al. (2017a) rely on a low-rank decomposition of a certain reshaping for the rows of the mixing matrix and the source channels in order to achieve a trade-off between data compression and accuracy of the data fit.

The present chapter and Boussé et al. (2017a) reshuffle respectively the coefficient matrices and the rows of the mixing matrix both in order to exhibit a low-rank matrix and subsequently, reduce the number of modeling parameters. Nonetheless, our modeling assumptions differ in three ways. We do not make restrictive assumptions on the signals rather than being obtained from a regular grid and being persistently exciting. We focus on the specific case where the sources signals $\mathbf{S}$ are known which allows getting rid of the ambiguity transformation inherent to blind source separation and to formulate spatial and temporal stability constraints on the coefficient matrices $\mathbf{A}_i$. Last, we exploit the 2D structure of the network and separability of the modeled functions in order to reduce the number of parameters. This point is detailed in Section 2.5. The different modeling assumptions lead to distinct optimization procedures.

In the following, the class of *low-Kronecker rank* matrices is studied with a focus on modeling 2D spatial-temporal dynamical systems of the VAR form. The Kronecker tool as presented in this chapter is meant to break down the curse of dimensionality when working with arrays of higher dimensions and without necessarily enforcing a priori a sparsity pattern in the network, hence allowing to discover both spatially varying dynamics and an unknown topology from the data. It also serves as the basis for other identification approaches such as subspace identification, as we will see in Chapter 3. As such, it will establish the fundamentals of a new modeling framework for the identification and analysis of large-scale 2D dynamical systems.

The challenge lies in deriving algorithms that are, on the one hand, scalable in terms of data storage as well as in terms of computational complexity in identifying and using these models, e.g in subsequent control design, and on the other hand,

that still ensures similar prediction performances compared to the unstructured least-squares estimates. The main contributions of this chapter are the definition of a new class of dynamical systems -of low Kronecker rank-, the formulation of a regularized cost function for identification and the formulation of an Alternating Least Squares algorithm with $\mathcal{O}(N^3 N_t)$ computational complexity, where N_t is the number of temporal samples.

The chapter has the following outline. Section 2.2 describes the class of low-Kronecker rank matrices, whereas Section 2.3 associates a VAR model associated with the sensor data. In Section 2.4, we describe regularization methods to emphasize the identification of stable models both in time and space. We study in Section 2.5 the Alternating Least Squares algorithm with a focus on the conditions to ensure global convergence. The methods are then illustrated in Section 2.6 on a randomly generated VARX model (with coefficient matrices sums of Kronecker) and a practical scenario dealing with open-loop identification of the atmospheric turbulence for adaptive optics purposes. Recursive updates to allow accurate identification of non-stationary data is presented in Section 2.7.

Notations. Let $i \in \{0, \dots, N-1\}$. For a matrix $\mathbf{X} \in \mathbb{R}^{N \times N}$, we denote with $\mathrm{diag}(\mathbf{X}, i)$ the i-th diagonal above the main diagonal and with $\mathrm{diag}(\mathbf{X}, -i)$ the i-th diagonal below the main diagonal. These vectors are then concatenated into a vector $\mathbf{d}_i$ defined with:

$$\mathbf{d}_i = \begin{bmatrix} \mathrm{diag}(\mathbf{X}, i)^T & \mathrm{diag}(\mathbf{X}, -i)^T \end{bmatrix}^T$$

We reshape the elements of a square matrix diagonal-wise starting by the main diagonal with the operator $\mathcal{D}$:

$$\mathcal{D}(\mathbf{X}) = \begin{bmatrix} \mathrm{diag}(\mathbf{X}, 0)^T & \mathbf{d}_1^T & \dots & \mathbf{d}_{N-1}^T \end{bmatrix}^T$$

which belongs to $\mathbb{R}^{N^2}$. The notation $\mathrm{BDiag}(\mathbf{X}_i, i = 1..N)$ forms a block-diagonal matrix with $\mathbf{X}_1$ to $\mathbf{X}_N$ located on the block-diagonal.

2.2. Preliminaries

The main computational rules related to the Kronecker product are described in the appendix A of this dissertation. In this section, we review some of the most important properties related to the decomposition of matrices with a sum of Kronecker products. Such a decomposition relies on the existence of block-matrices of equal size and that allows for a re-organization of the entries into a low-rank reshuffled matrix.

Definition 2.1. *van Loan and Pitsianis (1993) Let $m_1, n_1, m_2, n_2 \in \mathbb{R}$. Let $\mathbf{X} \in \mathbb{R}^{m_1 m_2 \times n_1 n_2}$ and $\mathbf{X}_{i,j} \in \mathbb{R}^{m_2 \times n_2}$ such that:*

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_{1,1} & \cdots & \mathbf{X}_{1,n_1} \\ \vdots & \ddots & \vdots \\ \mathbf{X}_{m_1,1} & \cdots & \mathbf{X}_{m_1,n_1} \end{bmatrix} \tag{2.5}$$

then the re-shuffle operator $\mathcal{R}(\mathbf{X}) \in \mathbb{R}^{m_1 n_1 \times m_2 n_2}$ is defined as:

$$\mathcal{R}(\mathbf{X}) = \begin{bmatrix} \mathrm{vec}(\mathbf{X}_{1,1}) & \dots & \mathrm{vec}(\mathbf{X}_{m_1,1}) & \mathrm{vec}(\mathbf{X}_{1,2}) & \dots & \mathrm{vec}(\mathbf{X}_{m_1,n_1}) \end{bmatrix}^T \tag{2.6}$$

There exists a permutation matrix $\mathbf{P}$ in the set $\mathbb{R}^{m_1n_1m_2n_2 \times m_1n_1m_2n_2}$ such that:

$$\text{vec}(\mathcal{R}(\mathbf{X})) = \mathbf{P}\text{vec}(\mathbf{X}) \tag{2.7}$$

Lemma 2.1. *van Loan (2000) Let* $\mathbf{X} = \mathbf{F} \otimes \mathbf{G}$*, with* $\mathbf{F}, \mathbf{G} \in \mathbb{R}^{m_1 \times n_1} \times \mathbb{R}^{m_2 \times n_2}$*. Then:*

$$\mathcal{R}(\mathbf{X}) = \text{vec}(\mathbf{F})\text{vec}(\mathbf{G})^T \tag{2.8}$$

The operation in Lemma 2.1 can also be reversed by the definition of the inverse vec operator ivec(.).

Lemma 2.2. *van Loan and Pitsianis (1993) Let* $\mathbf{X}$ *be defined as in Definition 2.1 and let an SVD of* $\mathcal{R}(\mathbf{X})$ *be given as:*

$$\mathcal{R}(\mathbf{X}) = \sum_{\ell=1}^{r} \sigma_\ell \mathbf{u}_\ell \mathbf{v}_\ell^T \tag{2.9}$$

and let $\text{ivec}\big(\mathbf{u}_\ell\big) = \mathbf{U}_\ell$*,* $\text{ivec}\big(\mathbf{v}_\ell\big) = \mathbf{V}_\ell$*, then:*

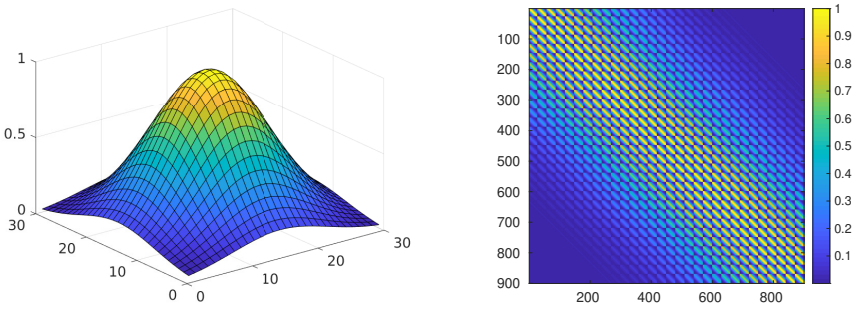
$$\mathbf{X} = \sum_{\ell=1}^{r} \sigma_\ell \mathbf{U}_\ell \otimes \mathbf{V}_\ell \tag{2.10}$$

The integer r is called the *Kronecker rank* of $\mathbf{X}$ with respect to the chosen block partitioning of $\mathbf{X}$ as given in Definition 2.1. When r is much smaller than N, $\mathbf{X}$ is said to have low-Kronecker rank. From Lemma 2.2, looking for a low-Kronecker rank approximation of a matrix is equivalent to finding a low-rank approximation of the reshuffled matrix. The operator $\mathcal{R}$ as defined in Definition 2.1 that forms a reshuffled matrix of minimal rank r is not unique: reshuffling the block-matrices row-wise rather than column-wise would yield the same Kronecker rank for $\mathbf{X}$. It then corresponds to the transpose of $\mathcal{R}(\mathbf{X})$. The sizes of the blocks should be chosen such that the Kronecker rank is minimal. These are usually infered in engineering applications to the dimensions of the sensor array.

Let a real function from $\mathbb{R}^2$ to $\mathbb{R}$, and separable in both coordinates. If this function describes the static behaviour of a particular mode of a system, a basis may be retrieved concatening columnwise the vectorized maps for all modes into a matrix. Figure 2.1 illustrates when the function is a Gaussian function. The matrix in the right may represent the influence that the actuators has on the wavefront in adaptive optics. Figure 2.2 depicts the reshuffled matrix and a single non-zero singular values.

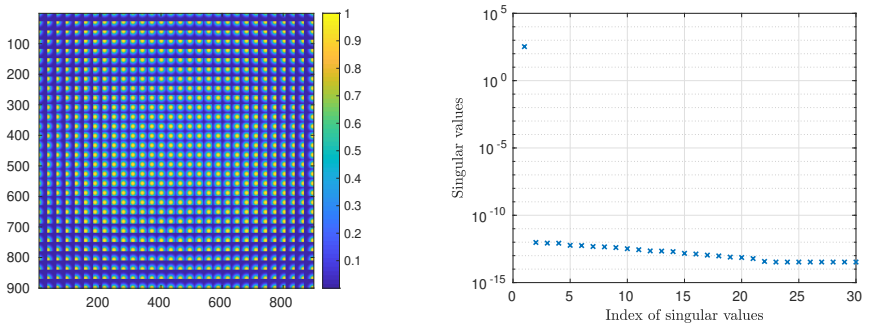
Definition 2.2. *(α-decomposable matrices, Massioni (2014))*
Let us consider a network of subsystems such that the latter belong to α different classes, themselves composed of ℓ_i subsystems. Let $\mathcal{P} \in \mathbb{R}^{L \times L}$ be an adjacency matrix. Define $\beta_j = \sum_{i=1}^{j} \ell_i$ (with $\beta_0 = 0$) and $\mathbf{I}_{[a_1:a_2]}$ as an $L \times L$ diagonal matrix which contains 1 in the diagonal entries of indices from a_1 to a_2 (included) and 0 elsewhere, then an α-decomposable matrix (for a given α) is a matrix of the following kind:

$$\mathbf{M} = \sum_{i=1}^{\alpha} \big(\mathbf{I}_{[\beta_{i-1}+1:\beta_i]} \otimes \mathbf{L}^{(i)} + \mathbf{I}_{[\beta_{i-1}+1:\beta_i]}\mathcal{P} \otimes \mathbf{N}^{(i)}\big) \tag{2.11}$$



(a) (b)

Figure 2.1: On the left is depicted a Gaussian function, whose discretized values on a regular grid are lifted into a column of the matrix on the right-hand-side. This matrix concatenates columnwise such vectorized maps, which have been obtained for all other positions for the peak value of the Gaussian.



(a) (b)

Figure 2.2: The matrix on the left-hand side is the reshuffled version of the one in Figure 2.1. Its first singular values are shown on the right-hand-side.

The matrices $\mathbf{L}^{(i)}$ *are the diagonal blocks of* $\mathbf{M}$ *that model the local dynamics, while the influence from the neighborhood is represented by the matrices* $\mathbf{N}^{(i)}$*, according to the structure of* $\mathcal{P}$*.*

When a state-transition matrix of a state-space model belongs to the class of α-decomposable matrices, the associated network has a known interconnection pattern whose adjacency matrix is $\mathcal{P}$ where α represents the number of non-identical subsystems in the network. For $\alpha = 1$ (and $\beta_1 = L$), these matrices are simply called *decomposable* matrices.

As a generalization of this class of structured matrices, we define next the class of sums-of-Kronecker product matrices.

Definition 2.3. *The class of sums-of-Kronecker product matrices contains matrices of the following kind:*

$$\mathbf{M} = \sum_{i=1}^{r} \mathbf{M}_a^{(i)} \otimes \mathbf{M}_b^{(i)} \tag{2.12}$$

with $\mathbf{M}_a^{(i)} \in \mathbb{R}^{m_1 \times n_1}$ *and* $\mathbf{M}_b^{(i)} \in \mathbb{R}^{m_2 \times n_2}$*. This class is denoted with* $\mathcal{K}_{2,r}$*. The matrices* $\mathbf{M}_a^{(i)}, \mathbf{M}_b^{(i)}$ *are called factor matrices.*

With this class of sums-of-Kronecker matrices, it is not necessary to have knowledge of the adjacency matrix $\mathcal{P}$ as with decomposable matrices. Therefore, the topology of the network need not to be known in advance. Moreover, the network may be composed of heterogeneous subsystems without any further specifications on the structure of the factor matrices. When describing large-scale networks, this structure is advantageous for its high compression capabilities. We now set (m_1, n_1, m_2, n_2) all equal to N. While N^4 entries are necessary to describe $\mathbf{M}$ in the unstructured case, only $2rN^2$ elements are required in the low Kronecker rank framework.

The next lemma provides insight on the benefits of using the class of Kronecker matrices to speed up simple linear algebra operations.

Lemma 2.3. *van Loan and Pitsianis (1993) Let* $\mathbf{x} \in \mathbb{R}^{N^2}$*. Then, the orders of magnitude of the computational complexity for matrix-vector multiplication, matrix-matrix multiplication and matrix inversion is as follows:*

	$\mathbf{A}, \mathbf{B} \in \mathbb{R}^{N^2 \times N^2}$	$\mathbf{A}, \mathbf{B} \in \mathcal{K}_{2,r}$
$\mathbf{Ax}$	$\mathcal{O}(N^4)$	$\mathcal{O}(rN^3)$
$\mathbf{AB}$	$\mathcal{O}(N^6)$	$\mathcal{O}(r^2N^3)$
$\mathbf{A}^{-1}$ *(case* $r = 1$*)*	$\mathcal{O}(N^6)$	$\mathcal{O}(N^3)$

The complexity obtained with the Kronecker parametrization considers the operations required for forming the factor matrices only.

Proof. The matrix vector multiplication $\mathbf{Ax} = \left(\sum_{i=1}^{r} \mathbf{A}_{\ell,i} \otimes \mathbf{A}_{r,i}\right)\mathbf{x}$ is rewritten into $\sum_{i=1}^{r} \mathbf{A}_{r,i} \text{ivec}(\mathbf{X}) \mathbf{A}_{\ell,i}^T$. The complexity in the matrix format is $2rN^3$ compared to N^4 without exploiting the sums-of-Kronecker structure. When computing the matrix-matrix multiplication, only the products between factor matrices are computed yielding a cost of r^2N^3. The inverse for $\mathbf{A}$ is determined via $\mathbf{A}^{-1} = \mathbf{A}_{\ell,1}^{-1} \otimes \mathbf{A}_{r,1}^{-1}$.

Computing $\mathbf{A}_{\ell,1}^{-1}$ and $\mathbf{A}_{r,1}^{-1}$ costs $\mathcal{O}(N^3)$. ■

Remark 2.1. *Approximating the inverse of large-scale low-Kronecker rank matrices $\mathbf{A} \in \mathcal{K}_{2,r}$ when the Kronecker rank is larger than one is an on-going research topic which Beylkin and Mohlenkamp (2005), Giraldi et al. (2014) and more recently Varnai (2017) have investigated.*

From Lemma 2.3, efficient linear algebra operations are possible when r is much smaller than N which is the class of Kronecker models we are interested in.

2.3. Problem formulation

Low-Kronecker rank matrices are now used to model the input-output relationship of 2D networked systems.

2.3.1. QUARKS models

Let us consider a regular grid with $N \times N$ nodes, with N larger than one, and each node is associated with a scalar sensor signal. Although the framework that we present here extends straightforwardly to arrays with nodes having multiple outputs, we only dwell on this case in Section 2.6. The sensor readings at the time instant k are stored in the matrix $\mathbf{S}(k) \in \mathbb{R}^{N \times N}$ as:

$$\mathbf{S}(k) = \begin{bmatrix} s_{1,1}(k) & s_{1,2}(k) & \cdots & s_{1,N}(k) \\ s_{2,1}(k) & s_{2,2}(k) & & s_{2,N}(k) \\ \vdots & \vdots & \ddots & \vdots \\ s_{N,1}(k) & s_{N,2}(k) & \cdots & s_{N,N}(k) \end{bmatrix} \tag{2.13}$$

In this chapter, we will consider that the temporal dynamics of this array of sensors are governed by the following VAR model:

$$\mathrm{vec}\big(\mathbf{S}(k)\big) = \sum_{i=1}^{p} \mathbf{A}_i \mathrm{vec}\big(\mathbf{S}(k-i)\big) + \mathrm{vec}\big(\mathbf{E}(k)\big) \tag{2.14}$$

where $\mathrm{vec}\big(\mathbf{E}(k)\big)$ is a zero-mean white noise with identity covariance matrix. Covariance estimation for low-Kronecker rank matrices has been addressed in Tsiligkaridis and Hero (2013) and is not the subject of further investigations in this chapter. The spatial dynamics are embedded within the structure of the matrices. For example, the spatial invariance is represented with a block-Toeplitz pattern for A, the spatial invariance and infinitely large dimensions with a circulant matrix (of finite size), the separability of a certain function with a Kronecker product. The latter decomposition is introduced to recast a two-dimensional problem into two coupled one-dimensional problems. The coefficient matrices $\mathbf{A}_i$ in the VAR model (2.14) are assumed to belong to the set $\mathcal{K}_{2,r}$. To address an identification problem, we parametrize these coefficient matrices as:

$$\mathbf{A}_i = \sum_{j=1}^{r_i} \mathbf{A}_i^{(j)}, \qquad \mathbf{A}_i^{(j)} = \mathbf{M}(\mathbf{b}_i^{(j)})^T \otimes \mathbf{M}(\mathbf{a}_i^{(j)}) \tag{2.15}$$

with the vectors $\mathbf{a}_i^{(j)}$ and $\mathbf{b}_i^{(j)}$ parametrizing the matrices $\mathbf{M}(\mathbf{a}_i^{(j)})$ and $\mathbf{M}(\mathbf{b}_i^{(j)})$ in an affine manner[1]. If no additional structure is enforced on $\mathbf{M}(\mathbf{a}_i^{(j)})$, then $\mathbf{a}_{i,\ell}^{(j)}$ denotes the ℓ-th column. With the notation $\mathrm{vec}\big(\mathbf{S}(k)\big) = \mathbf{s}(k)$, the VAR model (2.14) can be rewritten as,

$$\mathbf{s}(k) \quad = \quad \sum_{i=1}^{p} \big(\sum_{j=1}^{r_i} \mathbf{M}(\mathbf{b}_i^{(j)})^T \otimes \mathbf{M}(\mathbf{a}_i^{(j)}) \big) \mathbf{s}(k-i) + \mathbf{e}(k) \tag{2.16}$$

We can also write the VAR model (2.16) as,

$$\mathbf{S}(k) = \sum_{i=1}^{p} \big(\sum_{j=1}^{r_i} \mathbf{M}(\mathbf{a}_i^{(j)}) \mathbf{S}(k-i) \mathbf{M}(\mathbf{b}_i^{(j)}) \big) + \mathbf{E}(k) \tag{2.17}$$

The VAR(X) models (2.16) or (2.17) are called *Kronecker VARX network models* and abbreviated with *QUARKS models*.

2.3.2. The identification problem of QUARKS models

Given the model structure of the QUARKS models, the problem of identifying these models from measurement sequences $\{\mathbf{S}(k)\}_{k=1..N_t}$ is fourfold:

1. The temporal order index p.
2. The spatial order index r_i for each coefficient matrix.
3. The parametrization of the matrices $\mathbf{M}(\mathbf{a}_i^{(j)})$ and $\mathbf{M}(\mathbf{b}_i^{(j)})$. An example of a parametrization of the matrices $\mathbf{M}(\mathbf{a}_i^{(j)})$ and $\mathbf{M}(\mathbf{b}_i^{(j)})$ is (block) Toeplitz, or banded.
4. The estimation of the parameter vectors $\mathbf{a}_i^{(j)}$, $\mathbf{b}_i^{(j)}$ up to an ambiguity transformation. This requires the specification of a cost function. An example of such a cost function using the model (2.17) is the following least squares cost function, for data batches with N_t points:

$$\min_{\mathbf{a}_i^{(j)}, \mathbf{b}_i^{(j)}} \sum_{k=p+1}^{N_t} \| \mathbf{S}(k) - \sum_{i=1}^{p} \big(\sum_{j=1}^{r_i} \mathbf{M}(\mathbf{a}_i^{(j)}) \mathbf{S}(k-i) \mathbf{M}(\mathbf{b}_i^{(j)}) \big) \|_F^2 \tag{2.18}$$

By the selection of the parameter p and the particular choices of the parametrization in step 3 above, various special cases of restricting the coefficient matrices $\mathbf{A}_i$ in (2.14) to particular sets such as $\mathcal{K}_{2,r_i}$ can be considered. Further constraints to the least-squares cost function (2.18) might be introduced to look for sparsity in the parametrization vectors $\mathbf{a}_i^{(j)}$ and $\mathbf{b}_i^{(j)}$.

[1] For now, we do not precise the size of $\mathbf{a}_i^{(j)}$ and $\mathbf{b}_i^{(j)}$ as it depends on the chosen parametrization of the factor matrices. For example, if these are unstructured, they are composed of N^2 elements each.

The non-uniqueness of the optimal solution for the cost function (2.18) is highlighted next. One way to solve this estimation problem is via:

$$\begin{aligned} \min_{\mathbf{A}_i, \mathbf{a}_i^{(j)}, \mathbf{b}_i^{(j)}} \quad & \sum_{k=p+1}^{N_t} \|\mathbf{s}(k) - \sum_{i=1}^{p} \mathbf{A}_i \mathbf{s}(k-i)\|_2^2 \\ \text{s.t} \quad & \mathbf{A}_i = \sum_{j=1}^{r_i} \mathbf{M}(\mathbf{b}_i^{(j)})^T \otimes \mathbf{M}(\mathbf{a}_i^{(j)}) \end{aligned} \tag{2.19}$$

From (2.7), the reshuffling operator $\mathcal{R}$ is bijective in $\mathbb{R}^{N^2 \times N^2}$, therefore the above minimization problem is equivalent to:

$$\begin{aligned} \min_{\mathbf{A}_i, \mathbf{U}_i, \mathbf{V}_i} \quad & \sum_{k=p+1}^{N_t} \|\mathbf{s}(k) - \sum_{i=1}^{p} \mathbf{A}_i \mathbf{s}(k-i)\|_2^2 \\ \text{s.t} \quad & \mathcal{R}(\mathbf{A}_i) = \mathbf{U}_i \mathbf{V}_i^T \end{aligned} \tag{2.20}$$

where:

$$\begin{aligned} \mathbf{U}_i &= \begin{bmatrix} \text{vec}\big(\mathbf{M}(\mathbf{a}_i^{(1)})\big) & \dots & \text{vec}\big(\mathbf{M}(\mathbf{a}_i^{(r_i)})\big) \end{bmatrix} \\ \mathbf{V}_i &= \begin{bmatrix} \text{vec}\big(\mathbf{M}(\mathbf{b}_i^{(1)})\big) & \dots & \text{vec}\big(\mathbf{M}(\mathbf{b}_i^{(r_i)})\big) \end{bmatrix} \end{aligned}$$

For a non-singular matrix $\mathbf{T}_i \in \mathbb{R}^{r_i \times r_i}$, the constraint (2.20) can be equivalently written as:

$$\mathcal{R}(\mathbf{A}_i) = \widetilde{\mathbf{U}}_i \widetilde{\mathbf{V}}_i^T \tag{2.21}$$

where: $\widetilde{\mathbf{U}}_i = \mathbf{U}_i \mathbf{T}_i$ and $\widetilde{\mathbf{V}}_i^T = \mathbf{T}_i^{-1} \mathbf{V}_i^T$. The matrix $\mathbf{T}_i$ is called the ambiguity transformation. The non-uniqueness of the factor matrices is not an issue for practical use of QUARKS models as it does not affect the prediction-error.

Remark 2.2. *Let $m \in \{1, .., N^2\}$. Blind source separation* (2.4) *as described in Boussé et al. (2017b) reshapes either (or both) the mixing vectors $\mathbf{M}(m, :)$ and sources $\mathbf{S}(m, :)$ in* (2.4) *to form low-rank matrices. Then, there exists* different *left and right matrices for each mixing vector $\mathbf{M}(m, :)$ such that $\mathcal{R}(\mathbf{M}(m, :)) = \mathbf{u}_m \mathbf{v}_m^T$, or equivalently,*

$$\mathbf{M}(m, :) = \sum_{j=1}^{r} \mathbf{u}_m(:, j)^T \otimes \mathbf{v}_m(:, j)^T \tag{2.22}$$

where $\mathbf{u}_m \in \mathbb{R}^{I \times r}, \mathbf{v}_m \in \mathbb{R}^{J \times r}$ for two scalars I, J. The parameters I, J are user-defined contrary to the QUARKS modeling, where $I, J = N$. Hence, all mixing vectors are decoupled independently contrary to the description for the QUARKS model (2.15) *which assumes that the reshuffling into a matrix of* both *the rows and columns of the mixing matrix $\mathbf{M}$ is low-rank.*
We illustrate in the case where $p = 1$ and $\mathbf{M} = \mathbf{A}_1$. If rank$(\mathcal{R}(\mathbf{M})) = r$, then

rank$(\mathcal{R}(\mathbf{M}(m,:))) = r$ *and rank*$(\mathcal{R}(\mathbf{M}(:,m))) = r$. *Fixing* I, J *to* N *and considering* N^2 *sources, there are* $2rN^3$ *unknown coefficients to estimate whereas the modeling* (2.15) *represents the coefficient matrices with* $2rN^2$ *entries. The QUARKS modeling decreases the data storage requirements by an order of magnitude.*

An important challenge in solving the parameter estimation problem (2.18) is the *computational efficiency* for the case when the size N of the array is assumed to be large.

2.4. Regularization inducing stability and sparsity

We aim at regularizing the least-squares (2.18) to favour stable VAR models and that are such that the spatial correlations decay with the distance. We propose two additional costs which add up to (2.18) without altering the convergence properties.

The Kronecker rank is assumed equal for all i, i.e $r_i = r$, without constraining the insights in this section. We introduce the notations:

$$\mathbf{M}_{\mathbf{a}_i} = \left[\mathbf{M}(\mathbf{a}_i^{(1)})^T \quad \cdots \quad \mathbf{M}(\mathbf{a}_i^{(r_i)})^T\right]^T, \quad \mathbf{M}_{\mathbf{a}} = \left[\mathbf{M}_{\mathbf{a}_1}^T \quad \cdots \quad \mathbf{M}_{\mathbf{a}_p}^T\right]^T$$

The matrix $\mathbf{M}_{\mathbf{b}}$ is similarly defined.

2.4.1. Stability of VAR models

The stability for VAR models is guaranteed in Chiuso and Pillonetto (2012) by modeling the impulse response from one node to all the other ones in the network as a zero-mean Gaussian process and with an adequately chosen covariance matrix, which ensures that the parameters of the impulse response are decaying with increasing temporal index. We refer to Chen et al. (2012) and Pillonetto et al. (2014) for kernel methods applied to system identification. In this paragraph, we integrate these results as an additional regularization to the cost function (2.18). A Diagonal-Correlated kernel is used and the associated positive definite matrix $\mathbf{P}_t$ is defined with:

$$p_{t_{i,j}} = \xi^{\frac{i+j}{2}} \eta^{|i-j|} \tag{2.23}$$

for $i, j = 1..p$, and where the parameters η, ξ are such that $-1 \leq \eta \leq 1$ and $0 \leq \xi < 1$. The optimal ones are determined either by grid search or within the framework of Bayesian optimization and tune both the decay rate and the smoothness of the impulse response. T Let $\mathbf{W}_t$ be a square root of $\mathbf{P}_t^{-1}$. As there is no prior information nor physical meaning to distinguish between the different factor matrices, these are regularized independently with the additional cost:

$$r_t(\mathbf{M}_{\mathbf{a}}, \mathbf{M}_{\mathbf{b}}) = \sum_{j=1}^{r} \|\mathbf{Q}_t \begin{bmatrix} \mathbf{U}_1(:,\jmath)\mathbf{V}_1(:,\jmath)^T \\ \vdots \\ \mathbf{U}_p(:,\jmath)\mathbf{V}_p(:,\jmath)^T \end{bmatrix} \|_F^2 \tag{2.24}$$

where $\mathbf{Q}_t = \mathbf{W}_t \otimes \mathbf{I}_{N^2}$. Such a regularization r_t is bilinear in the unknowns.

2.4.2. Spatial sparsity

Real graphs or the regular networks from discretized Partial Differential Equations are such that each node is connected to a very limited number of other nodes with respect to the network's size. In the latter case, the neighborhood is localized which gives rise to a multi-banded structure of the full coefficient matrices, equivalent to a banded structure of each factor matrix. In case of high spatial coupling, we rather tune the decay of the parameters away from the main diagonal rather than minimizing the number of non-zero entries. It will become clear in the next section that we would like to avoid all non-differentiable functions in the cost function, hence the focus is laid on kernel methods rather than on minimizing the ℓ_1-norm of the factor matrices. An exponentially decreasing sequence has been studied in Chiuso and Pillonetto (2012) for sparse network identification. Following the same line of thoughts, we introduce a diagonal matrix $\mathbf{K}_s \in \mathbb{R}^{N^2 \times N^2}$ such that:

$$\mathbf{K}_s = \begin{bmatrix} \mathbf{I}_N k_1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{2(N-1)} k_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0} & \mathbf{I}_2 k_N \end{bmatrix} \tag{2.25}$$

where the scalars k_i are such that $0 < k_i < k_{i+1}$. For example, a valid choice of such scalars is $k_i = e^{\zeta i}$ with $\zeta > 0$. An additional cost which favours factor matrices with values decaying away from the main diagonal reads,

$$r_s(\mathbf{M_a}, \mathbf{M_b}) = \sum_{i=1}^{p} \sum_{j=1}^{r} \|\mathbf{K}_s \mathcal{D}\big(\mathbf{M}(\mathbf{a}_i^{(j)})\big) \mathcal{D}\big(\mathbf{M}(\mathbf{b}_i^{(j)})\big)^T \mathbf{K}_s^T\|_F^2 \tag{2.26}$$

2.4.3. Structured factor matrices

The parametrization of the factor matrices based on prior knowledge of the network may help either to further reduce the computational complexity of the model identification step, or to cast the model into a structure useful for control. The first category include banded, symmetric, Toeplitz and circulant patterns. Exploring such structures on the factor matrices is very attractive numerically as the number of parameters to be estimated reduces further. The block-Toeplitz Toeplitz-blocks structure arises e.g when modeling 2D homogeneous spatially-invariant phenomena on a rectangular grid. Many functions in optics are isotropic, for example the Airy function, or the wavefront covariance matrix $\mathbf{C}_{\phi,0}$, and can be modeled with a sum of few Kronecker terms. The Kronecker and block-Toeplitz Toeplitz-blocks structures are related, but not equivalent.

Lemma 2.4. *Let* $\mathbf{X} \in \mathbb{R}^{N^2 \times N^2}$.
If $\mathbf{X}$ *is symmetric block-Toeplitz, then* $\mathbf{X}$ *has a Kronecker rank at most equal to* N.
If $\mathbf{X}$ *has a Kronecker rank of one, it does not* in general *imply neither that* $\mathbf{X}$ *is block-Toeplitz nor has Toeplitz-blocks.*

Proof. The first proposition is proved by using the reshuffling operator $\mathcal{R}$. It is then observed that the Toeplitz-blocks are not used in reducing further the Kronecker

rank. For the second point, the factor matrices may be for example randomly generated. ■

The second category contains for example the sparse (with unknown pattern of non-zero entries) or SSS structure. The SSS structure is more general than the Toeplitz, especially when it comes to model spatially-varying systems. As mentioned in Chapter 1, the efficient use of SSS matrices has been thoroughly studied in Rice and Verhaegen (2009) although the extension to Multi-Level structures is an ongoing research question. Modeling each factor matrix of the model as SSS enables significant improvements in the computational cost for future simple linear algebra operations. For example, the cost for standard matrix computations scales linearly with respect to the matrix size. For example, inverting a matrix $\mathbf{M}$ belonging to $\mathbb{R}^{N^2\times N^2}$ written as $\mathbf{M}=\mathbf{M}_1\otimes\mathbf{M}_2$ in which both $\mathbf{M}_1$, $\mathbf{M}_2$ have a SSS structure requires $\mathcal{O}(N)$ operations instead of $\mathcal{O}(N^6)$. For identifying factors with a SSS structure, the factors are first identified without any particular parametrization, and second, the SSS generators are extracted from the low-rank off-diagonal submatrices. This two-step procedure is proposed because such a parametrization for the matrices $\mathbf{M}(\mathbf{a}_i^{(j)}),\mathbf{M}(\mathbf{b}_i^{(j)})$ is not affine in the parameters $\mathbf{a}_i^{(j)},\mathbf{b}_i^{(j)}$.

2.4.4. The regularized cost function for QUARKS identification

The cost function for the identification of sparse stable QUARKS models reads:

$$\begin{aligned}\min_{\mathbf{a}_i^{(j)},\mathbf{b}_i^{(j)}} \quad & \sum_{k=p+1}^{N_t}\|\mathbf{S}(k)-\sum_{i=1}^{p}\big(\sum_{j=1}^{r_i}\mathbf{M}(\mathbf{a}_i^{(j)})\mathbf{S}(k-i)\mathbf{M}(\mathbf{b}_i^{(j)})\big)\|_F^2 \\ & +\mu\cdot r_t(\mathbf{M_a},\mathbf{M_b})+\lambda\cdot r_s(\mathbf{M_a},\mathbf{M_b}) \end{aligned} \tag{2.27}$$

where μ,λ are regularization parameters. The cost function (2.27) belongs to the class of multi-convex problems in which fixing one set of variables yields a convex problem. Adding regularization to the cost function aims at decreasing the prediction error of the estimated VAR model when dealing with noisy and short data batches rather than speeding up the convergence as done in Li et al. (2013).

Remark 2.3. *The regularization in* (2.27) *is* bilinear *contrary to the one analyzed in Udell et al. (2016), Baldi and Hornik (1989) within the framework of Principal Component Analysis (PCA). Based on Udell et al. (2016), a regularization for* (2.20) *would minimize a (weighted) sum of the Frobenius norm of the factor matrices.*

2.5. A bi-convex cost function

The factor matrices are assumed unstructured in the upcoming sections.

2.5.1. An Alternating Least Squares approach

The regularized least-squares representation (2.27) is bilinear in its unknowns but features factor matrices of size $N\times N$ only. It has moreover the advantage that constraints on the parametrization of the matrices $\mathbf{M}(\mathbf{a}_i^{(j)})$ and $\mathbf{M}(\mathbf{b}_i^{(j)})$ can be more

easily taken into consideration than via a low-rank minimization on the large-scale reshuffled matrix, $\mathcal{R}(\mathbf{A}_i)$. A couple of optimization routines are candidates. A non-linear optimization scheme such as the separable least-squares in Bruls et al. (1999) proceeds with two steps, one of which however consists of non-linear optimization. Iterative hierarchical algorithms have been derived as a generalization of the linear Gauss-Seidel iterations for solving coupled Sylvester matrix equations in Ding and Chen (2005). Similarly as in Hoff (2015), we propose to address (2.19) by solving a sequence of linear least-squares and using ALS, which is a special case of the block *non-linear* Gauss-Seidel method as highlighted in Li et al. (2013).
We now rewrite the cost function (2.27) into two updates, (2.30) and (2.31), which are solved iteratively until convergence to a stationary point. The data-fitting term for updating $\mathbf{M_b}$ is,

$$\min_{\mathbf{M_b}} \quad \|\widetilde{\mathbf{S}} - \overline{\mathbf{X}}_a \mathbf{M_b}\|_F^2 \tag{2.28}$$

where:

$$\begin{aligned}
\widetilde{\mathbf{S}} &= \begin{bmatrix} \widetilde{\mathbf{S}}_{1,1} & \dots & \widetilde{\mathbf{S}}_{1,N} \\ \vdots & & \vdots \\ \widetilde{\mathbf{S}}_{N,1} & \dots & \widetilde{\mathbf{S}}_{N,N} \end{bmatrix}, \quad \widetilde{\mathbf{S}}_{i,j} = \begin{bmatrix} s_{i,j}(p+1) \\ \vdots \\ s_{i,j}(N_t) \end{bmatrix} \\
\overline{\mathbf{X}}_a &= \begin{bmatrix} \overline{\mathbf{X}}_{a,1} & \dots & \overline{\mathbf{X}}_{a,p} \end{bmatrix}, \quad \overline{\mathbf{X}}_{a,i} = \begin{bmatrix} \overline{\mathbf{X}}_{a,i,1} & \dots & \overline{\mathbf{X}}_{a,i,r} \end{bmatrix} \\
\overline{\mathbf{X}}_{a,i,j} &= (\mathbf{I}_N \otimes \widetilde{\mathbf{U}}_i) \begin{bmatrix} \mathbf{a}_{i,1}^{(j)} \otimes \mathbf{I}_N \\ \vdots \\ \mathbf{a}_{i,N}^{(j)} \otimes \mathbf{I}_N \end{bmatrix} \\
\widetilde{\mathbf{U}}_i &= \begin{bmatrix} \mathbf{S}(p+1-i)(1,:) & \dots & \mathbf{S}(p+1-i)(\mathtt{N},:) \\ \vdots & & \vdots \\ \mathbf{S}(N_t-i)(1,:) & \dots & \mathbf{S}(N_t-i)(\mathtt{N},:) \end{bmatrix}
\end{aligned}$$

The term $\mu \cdot r_t(\mathbf{M_a}, \mathbf{M_b})$ is rewritten as $\|\mathbf{F}_b(\mathbf{M_a})\mathbf{M_b}\|_F^2$, where $\mathbf{F}_b(\mathbf{M_a})$ contains $p \times p$ block-matrices. The block at position (i, j) is equal to:

$$\sqrt{\mu} w_{t,(i,j)} \text{BDiag}(\mathbf{I}_N \otimes \text{vec}(\mathbf{M}(\mathbf{a}_i^{(m)})), m = 1..r)$$

Moreover, a matrix $\mathbf{G}_b(\mathbf{M_a})$ is derived such that the regularization for spatial sparsity reads:

$$\lambda \cdot r_s(\mathbf{M_a}, \mathbf{M_b}) = \|\mathbf{G}_b(\mathbf{M_a})\text{vec}(\mathbf{M_b})\|_2^2 \tag{2.29}$$

where:

$$\begin{aligned}
\mathbf{G}_b(\mathbf{M_a}) &= \sqrt{\lambda}\mathbf{P}_{r,s}\text{BDiag}(\mathbf{G}_{b,j}(\mathbf{M_a}), j = 1..r)\mathbf{P}_{c,s} \\
\mathbf{G}_{b,j}(\mathbf{M_a}) &= \text{BDiag}((\mathbf{K}_s \otimes \mathbf{K}_s)\mathcal{D}(\mathbf{M}(\mathbf{a}_i^{(j)}), i = 1..p)
\end{aligned}$$

The matrices $\mathbf{P}_{r,s}$ and $\mathbf{P}_{c,s}$ permute respectively the rows and columns such that $\mathbf{G}_b(\mathbf{M_a})$ is block-diagonal. We denote the i-th block in the main block-diagonal with

$\mathbf{G}_b(\mathbf{M_a})[i]$. The cost function (2.27) is then separable for each column of $\mathbf{M_b}$:

$$\min_{\mathbf{M_b}} \sum_{i=1}^{N} \| \underbrace{\begin{bmatrix} \widetilde{\mathbf{S}} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}}_{\mathbf{Y}} - \underbrace{\begin{bmatrix} \overline{\mathbf{X}}_a \\ \mathbf{F}_b(\mathbf{M_a}) \\ \mathbf{G}_b(\mathbf{M_a})[i] \end{bmatrix}}_{\mathbf{F}_{\mathbf{b}i}} \mathbf{M_b}(:, \mathtt{i}) \|_F^2 \tag{2.30}$$

Similarly, the least-squares for updating $\mathbf{M_a}$ is:

$$\min_{\mathbf{M_a}} \sum_{i=1}^{N} \|\mathbf{Y} - \underbrace{\begin{bmatrix} \overline{\mathbf{X}}_b \\ \mathbf{F}_a(\mathbf{M_b}) \\ \mathbf{G}_a(\mathbf{M_b})[i] \end{bmatrix}}_{\mathbf{F}_{\mathbf{a}i}} \mathbf{M_a}(:, \mathtt{i}) \|_F^2 \tag{2.31}$$

where:

$$\begin{aligned} \overline{\mathbf{X}}_b &= \begin{bmatrix} \overline{\mathbf{X}}_{b,1,1} & \dots & \overline{\mathbf{X}}_{b,1,r} & \dots & \overline{\mathbf{X}}_{b,p,r} \end{bmatrix} \\ \overline{\mathbf{X}}_{b,i,j} &= (\mathbf{I}_N \otimes \widetilde{\mathbf{U}}_i) \begin{bmatrix} \mathbf{I}_N \otimes \mathbf{b}_{i,1}^{(j)} \\ \vdots \\ \mathbf{I}_N \otimes \mathbf{b}_{i,N}^{(j)} \end{bmatrix} \end{aligned}$$

The least-squares (2.30) and (2.31) are iteratively solved starting with some random initial guess for $\mathbf{M_a}$ until some stopping criterion is reached. The iterations are stopped when the decrease between two consecutive values of the cost function is lower than a given threshold. Algorithm 2.1 summarizes the steps.

2.5.2. Convergence

The musings in Mohlenkamp (2013) detail properties about ALS and multilinear fittings in general. The cost function is monotonically decreasing during the iterations and the rate of convergence is at most linear. The convergence of the global matrices $\{\mathbf{A}_i\}_{i=1..p}$ is a necessary condition but not sufficient for stopping the iterations. Again because of the non-uniqueness, the factor matrices might still change and compensate each other without modifying the value of the cost function. Whether the entries of the factor matrices converge is a more adequate question. We answer this question after defining a *true* value for the factors. Although all of the ones that minimize the cost function are equivalent, we assume that the factors with a particular norm of the columns represent the true value we would like to see converge. To do this, we consider a normalized version of Algorithm 2.1 and prove that the iterations converge to a fixed point of a particular functional. We assume the temporal order p and spatial order r to be both equal to one, that both regularization parameters equal to zero, and that the norm of $\mathbf{M_b}(:, \mathtt{i})$ is known (which is rarely the case in practice). The two following modifications to the ALS are introduced. First, the columns ${\mathbf{M_b}}^{(\kappa)}(:, \mathtt{i})$ for i in the set $\{1, .., N\}$, are normalized after line 8,

$$\mathbf{M_b}_1^{(\kappa)}(:, \mathtt{i}) \leftarrow \mathbf{M_b}_1^{(\kappa)}(:, \mathtt{i}) \frac{\|\mathbf{M_b}_1(:, \mathtt{i})\|_2}{\|\mathbf{M_b}_1^{(\kappa)}(:, \mathtt{i})\|_2} \tag{2.32}$$

Algorithm 2.1: ALS for QUARKS identification

Input : $\{\mathbf{S}(k)\}_{k=1..N_t}, r, p, \{\|\mathbf{M_b}_j(:,\mathtt{i})\|_2\}_{i=1..N,j=1..p}$

Output: $\{\widehat{\mathbf{M_a}_j}, \widehat{\mathbf{M_b}_j}\}_{j=1..p}$

/* Default values */

$\kappa = 1, \kappa_{max} = 50, \epsilon = \infty, \epsilon_{min} = 10^{-3}$

/* Initial guesses */

$\mathbf{M_a}^{(0)} = \mathtt{randn(Nrp,N)}$

/* Start ALS */

while $\kappa < \kappa_{max}$ *and* $\epsilon > \epsilon_{min}$ **do**

  /* Optimize over $\mathbf{M_b}$ */

  Compute $\mathbf{F}_0^T\mathbf{F}_0$ where: $\mathbf{F}_0 \leftarrow \begin{bmatrix} \overline{\mathbf{X}}_a^{(\kappa-1)} \\ \mathbf{F}_b(\mathbf{M_a}^{(\kappa-1)}) \end{bmatrix}$

  for $i = 1..N$ **do**

    Form $\mathbf{F_b}_i$ from $\mathbf{G}_b(\mathbf{M_a})\,[i]$

    $\mathbf{M_b}^{(\kappa)}(:,\mathtt{i}) \leftarrow (\mathbf{F_b}_i^T\mathbf{F_b}_i)^{-1}\mathbf{F_b}_i^T\mathbf{Y}(:,\mathtt{i})$

  end

  /* Optimize over $\mathbf{M_a}$ */

  Compute $\mathbf{F}_0^T\mathbf{F}_0$ where $\mathbf{F}_0 \leftarrow \begin{bmatrix} \overline{\mathbf{X}}_b^{(\kappa)} \\ \mathbf{F}_a(\mathbf{M_b}^{(\kappa)}) \end{bmatrix}$

  for $i = 1..N$ **do**

    Form $\mathbf{F_a}_i$

    $\mathbf{M_a}^{(\kappa)}(:,\mathtt{i}) \leftarrow (\mathbf{F_a}_i^T\mathbf{F_a}_i)^{-1}\mathbf{F_a}_i^T\mathbf{Y}(:,\mathtt{i})$

  end

  /* Check stopping criterion */

  $c^{(\kappa)} \leftarrow \|\widetilde{\mathbf{S}} - \overline{\mathbf{X}}_b^{(\kappa)}\mathbf{M_a}^{(\kappa)}\|_F^2$

  $\epsilon \leftarrow |c^{(\kappa)} - c^{(\kappa-1)}|$

  $\kappa \leftarrow \kappa + 1$

end

Second, and after convergence to a stationary point, the sign ambiguities are dealt with by multiplying both factors with the sign of the known element,

$$\widehat{\mathbf{M}_{\mathbf{b}1}} \leftarrow \mathbf{M}_{\mathbf{b}1}^{(\kappa-1)}\text{sign}(b_{1,1}^{(\kappa-1)}), \quad \widehat{\mathbf{M}_{\mathbf{a}1}} \leftarrow \mathbf{M}_{\mathbf{a}1}^{(\kappa-1)}\text{sign}(b_{1,1}^{(\kappa-1)}) \tag{2.33}$$

such that the first (non-zero) entry of $\widehat{\mathbf{M}_{\mathbf{b}1}}$ is strictly positive. We denote the i-th column $\mathbf{a}_{1,i}^{(1)}$ in short with $\mathbf{a}_i$, and the vector concatenating all of them with $\mathbf{a}$. A functional representation of the three steps in Algorithm 2.1 (including the aforementioned normalization) reads:

$$\widehat{\mathbf{b}}^{(\kappa)} = \mathcal{F}_1(\widehat{\mathbf{a}}^{(\kappa-1)}), \quad \widehat{\mathbf{b}_{\mathbf{n}}}^{(\kappa)} = \mathcal{F}_2(\widehat{\mathbf{b}}^{(\kappa)}), \quad \widehat{\mathbf{a}}^{(\kappa)} = \mathcal{F}_3(\widehat{\mathbf{b}_{\mathbf{n}}}^{(\kappa)}) \tag{2.34}$$

These equations can be expressed using a single operator $\mathcal{F}$ mapping the estimate $\widehat{\mathbf{a}}^{(\kappa-1)}$ to $\widehat{\mathbf{a}}^{(\kappa)}$:

$$\widehat{\mathbf{a}}^{(\kappa)} = \mathcal{F}_3(\mathcal{F}_2(\mathcal{F}_1(\widehat{\mathbf{a}}^{(\kappa-1)}))) = \mathcal{F}(\widehat{\mathbf{a}}^{(\kappa-1)}) \tag{2.35}$$

Lemma 2.5. [The Contraction Mapping Theorem, Granas and Dugundji (2001)] *Let (X, D) be a non-empty complete metric space where D is a metric on X. Let $\mathcal{F} : X \to X$ be a contraction mapping on X, i.e., there is a non-negative real number $Q < 1$ such that $D(\mathcal{F}(\mathbf{x}), \mathcal{F}(\mathbf{y})) \leq QD(\mathbf{x}, \mathbf{y})$, for all $\mathbf{x}, \mathbf{y} \in X$. Then the map $\mathcal{F}$ admits one and only one fixed point $\mathbf{x}^\star \in X$ which means $\mathbf{x}^\star - \mathcal{F}(\mathbf{x}^\star) = 0$. Furthermore, this fixed point can be found from the convergence of an iterative sequence defined by $\mathbf{x}^{(\kappa+1)} = \mathcal{F}\big(\mathbf{x}^{(\kappa)}\big)$ for $\kappa = 1, 2, ...$ with an arbitrary starting point $\mathbf{x}^{(0)}$ in X.*

If $\mathbf{a}$ is a fixed point of $\mathcal{F}$, then the gradient with respect to $\mathbf{a}$ of the cost function (2.27) (simplified from the assumptions made in this section) is zero. If $\mathbf{a}$ is not changed during one full cycle of the algorithm, then the gradient with respect to $\mathbf{b}$ of the cost function (2.27) is also zero. Consequently, the gradient of the cost function with respect to both $\mathbf{a}$ and $\mathbf{b}$ is zero. A fixed point of $\mathcal{F}$ is a stationary point of the cost function in the minimization (2.27). The reverse implication is not necessarily true: there are many other stationary points that are discarded from the analysis when normalizing. If the fixed point is unique as we show in this very particular case of ALS, it corresponds to the targeted factor matrices for which the norm of the columns is assumed to be known. We refer to these as the true values. We now define a set associated to the true value $\mathbf{a}$:

$$\mathcal{X}_{\boldsymbol{a}} \quad = \quad \{\widehat{\boldsymbol{a}} \in \mathbb{R}^{N^2} | \forall i \in \{1, ..., N\}, \|\widehat{\mathbf{a}}_i\|_2 \leq \|\mathbf{a}_i\|_2\}$$

Theorem 2.1. *Let $p = 1, r = 1$ and $(\lambda, \mu) = (0, 0)$.*
If the following statements are true:

- **A1***: the noise components are independent identically distributed with zero-mean and finite variance.*
- **A2***: the matrix $\widetilde{\mathbf{U}}_1$ is full column rank.*

- **A3**: *either* $\|\mathbf{b}_i\|_2$ *or* $\|\mathbf{a}_i\|_2$ *is known for all* i *and the first non-zero entry of* $\mathbf{b}$ *or* $\mathbf{a}$ *is strictly positive.*

- **A4**: *the initial guess* $\widehat{\mathbf{a}}^{(0)}$ *is non-zero.*

Then, the map $\mathcal{F} : \mathcal{X}_{\boldsymbol{a}} \to \mathcal{X}_{\boldsymbol{a}}$ *is a contraction on* $\mathcal{X}_{\boldsymbol{a}}$ *when* $N_t \to \infty$ *and has a unique fixed point which corresponds to the true parameters* $\boldsymbol{a}$*.*

Proof. The convergence proof relies on the work in Li et al. (2015) where the result is established when the unknowns are vectors. The non-trivial extensions are reported in the appendix of this chapter. ■

The assumption **A2** corresponds to the persistency of excitation from the data and is a key ingredient in the convergence. Theorem 2.1 proves that whatever the non-zero initial conditions the iterations (2.34) converge to a fixed point asymptotically when N_t approaches infinity. When the temporal order is strictly larger than one, the solution to an update in line 7 or 12 in Algorithm 2.1 is unique if and only if the matrix $\bar{\mathbf{X}}_b$ (or $\bar{\mathbf{X}}_a$) is full rank. This condition provides indications on how to choose the initial guess. In practice, we choose randomly generated initial guesses independent for each factor matrix such that $\bar{\mathbf{X}}_a^{(0)}$ is full rank.

2.5.3. Computational complexity

Lemma 2.6. *The computational cost for estimating the QUARKS scales with* $\mathcal{O}(N^4 N_t)$ *compared to* $\mathcal{O}(N^6)$ *in the unstructured case. If the regularization parameter* λ *is zero, the cost reduces to* $\mathcal{O}(N^3 N_t)$*.*

Proof. Using (2.14) with temporal data within the range $\{1, ..., N_t\}$ with $N_t \geq N^2 p$ to recover a unique solution, we write:

$$\underbrace{\begin{bmatrix}\mathbf{s}(p+1) & \dots & \mathbf{s}(N_t)\end{bmatrix}}_{\mathbf{S_f}} = \begin{bmatrix}\mathbf{A}_1 & \dots & \mathbf{A}_p\end{bmatrix} \underbrace{\begin{bmatrix}\mathbf{s}(p) & \dots & \mathbf{s}(N_t-1) \\ \vdots & & \vdots \\ \mathbf{s}(1) & \dots & \mathbf{s}(N_t-p)\end{bmatrix}}_{\mathbf{S_p}} + \mathbf{E_p} \tag{2.36}$$

The least-squares estimation for the coefficient matrices is hence equal to:

$$\begin{bmatrix}\widehat{\mathbf{A}}_1 & \dots & \widehat{\mathbf{A}}_p\end{bmatrix} = \mathbf{S_f}\mathbf{S_p}^T\left(\mathbf{S_p}\mathbf{S_p}^T\right)^{-1} \tag{2.37}$$

The dependency on the number of temporal samples is kept: a correct identification in noisy conditions often requires $N_t \geq N^2 p$. The complexity for estimating unstructured VAR is $\mathcal{O}(N^4 N_t)$.

We assume that the Kronecker rank and the number of iterations to reach convergence are independent of N. In practice, larger arrays require a larger number of temporal samples and therefore, N_t is included in the computational count. The lines 4, 7, 9, and 12 are the most computationally costly of Algorithm 2.1.

If $\lambda = 0$: forming $\bar{\mathbf{X}}_a^{(\kappa-1)}$ requires $(N_t - p)rp$ matrix-matrix multiplications of size $N \times N$. The number of temporal samples is such that $N(N_t - p) \geq Nrp$

to guarantee a unique solution of each subproblem without regularization. The complexity is $\mathcal{O}(N^3N_t)$ flops. Computing its inverse requires $\mathcal{O}(N^3)$ whereas right-multiplying the latter with $\mathbf{F}_{\mathbf{b}_i}{}^T$ reaches $\mathcal{O}(N^3N_t)$. The total cost for Algorithm 2.1 with $\lambda = 0$ reaches $\mathcal{O}(N^3N_t)$ where $N_t \gg rp$.

If $\lambda \neq 0$: the cost for computing $\mathbf{F}_{\mathbf{b}i}$ boils down to computing $\mathbf{F}_0^T\mathbf{F}_0$ because $\mathbf{G}_b(\mathbf{M}_{\mathbf{a}}{}^{(\kappa-1)})[i]$ is sparse. Computing the inverse of $\mathbf{F}_{\mathbf{b}i}{}^T\mathbf{F}_{\mathbf{b}i}$ requires $\mathcal{O}(N^3)$ flops whereas multiplying the inverted matrix with $\mathbf{F}_{\mathbf{b}i}{}^T$ costs $\mathcal{O}(N^3N_t)$. These two operations need to be repeated N times, although it should be performed in parallel. The cost for computing the lines 9 and 12 is similar to the above discussion.

Operation	**Flops**
Unstructured estimation	
$\mathbf{S_p}\mathbf{S_p}{}^T$	$\mathcal{O}(N^4N_t)$
$(\mathbf{S_p}\mathbf{S_p}{}^T)^{-1}$	$\mathcal{O}(N^6)$
$\mathbf{S_f}\mathbf{S_p}{}^T$	$\mathcal{O}(N^4N_t)$
QUARKS estimation	
Lines 4 and 9	$\mathcal{O}(N^3N_t)$
Lines 7 and 12 (for each i)	$\mathcal{O}(N^3N_t)$
Lines 7 and 12 (total for all i)	$\mathcal{O}(N^4N_t)$
Total (with $\lambda \neq 0$)	$\mathcal{O}(N^4N_t)$
Total (with $\lambda = 0$)	$\mathcal{O}(N^3N_t)$

Table 2.1: Computational complexity.

■

2.6. Numerical examples: batches of data

The identification method is first illustrated with a randomly generated QUARKS model and then with an application to AO.

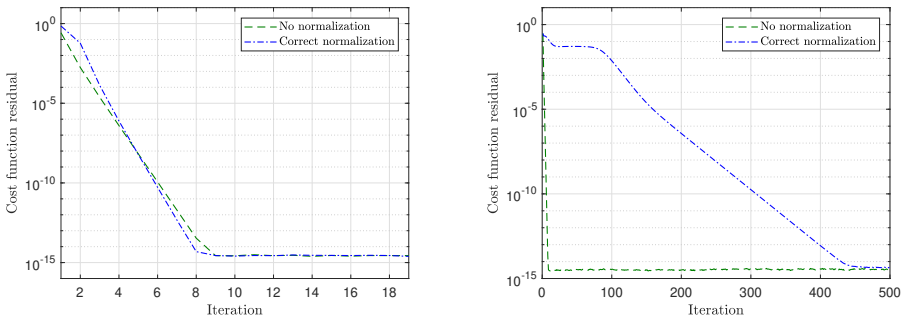
2.6.1. Illustrating convergence

We first illustrate the convergence of Algorithm 2.1 with different normalizations for a randomly generated QUARKS model whose temporal order and Kronecker rank is known,

$$\mathbf{S}(k) = \sum_{i=1}^{p}\sum_{j=1}^{r}\mathbf{M}(\mathbf{a}_i^{(j)})\mathbf{U}(k-i)\mathbf{M}(\mathbf{b}_i^{(j)}) \tag{2.38}$$

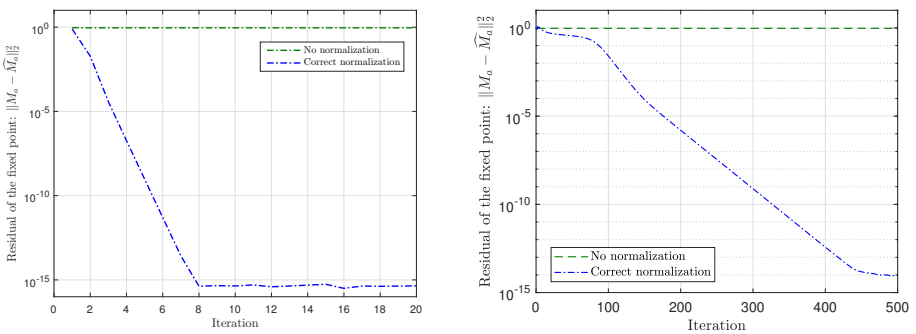
where $\mathbf{S}(k) \in \mathbb{R}^{10\times 10}$. The factor matrices $\mathbf{M}(\mathbf{a}_i^{(j)})$ and $\mathbf{M}(\mathbf{b}_i^{(j)})$ have random entries following a uniform distribution. The entry in the first row and column is strictly positive. The input is a white Gaussian noise with unit variance. The number of temporal samples N_t is set to 10^3. Both λ, μ are set to 0. Two scenarios were tested to analyze the influence of the temporal and spatial order (p, r) on the convergence.

In Figure 2.3-(a) and Figure 2.4-(a), the pair (p, r) is set to $(2, 1)$. Figure 2.3-(a) plots the residual of the QUARKS cost function as a function of the iteration



(a) (b)

Figure 2.3: Evolution of the cost function as a function of the number of iterations with two normalizations (no normalization, normalization as explained in 2.5.2). (a): the pair (p, r) is set to $(2, 1)$. (b): the pair (p, r) is set to $(1, 2)$.



(a) (b)

Figure 2.4: Evolution of the least squares between the true value $\mathbf{M}(\mathbf{a}_i^{(j)})$ and its estimate as a function of the number of iterations with two normalizations (no normalization, normalization as explained in 2.5.2). (a): the pair (p, r) is set to $(2, 1)$. (b): the pair (p, r) is set to $(1, 2)$.

number for both normalized and non-normalized algorithms. Convergence to a global minimum is observed for both cases. The convergence towards a unique fixed point is shown with Figure 2.4-(a) which displays the least-squares residual between the true value $\mathbf{M}(\mathbf{a}_i^{(j)})$ and its estimate. When normalizing the columns of the factor matrix, the factor matrices converge to their true values whereas it is not the case for the non-normalized version. Although both algorithms reach a global minimum, the solution to the QUARKS identification problem is not unique as highlighted with Figure 2.4-(b), and both solutions are equivalent as they provide a similar output-error (up to machine precision). The case $(p, r) = (1, 2)$ is analyzed in Figure 2.3-(a) and Figure 2.4-(b). Normalizing may affect the convergence speed to a global minimum. The convergence of the estimates to a fixed point when r is larger than one cannot be guaranteed especially because of the ambiguity transform.

2.6.2. Case study: Adaptive optics

The turbulence is generated according to the Multiscale Phase Screen Synthesis approach detailed in Beghi et al. (2011). More specifically, only the low resolution process is used here based on the Fast Fourier Transform Moving Average (FFT-MA) generator. In short, the phase screen $\mathbf{x}$ with dimensions $m \times m$ can be represented as a MA model:

$$x_{u,v} = \sum_{k_u,k_v} \theta_{k_u,k_v} \epsilon_{u-k_u,v-k_v} \tag{2.39}$$

with $\boldsymbol{\epsilon}$ a zero-mean white noise process with unit variance and $\boldsymbol{\theta}$ the MA coefficients. To determine the MA coefficients $\boldsymbol{\theta}$, the spatial covariance matrix $\mathbf{C}_\phi$ of the atmospheric turbulence based on the Von Karman theory is considered, such that:

$$c_{\phi_{u,v}} = \sum_{k_u,k_v} \theta_{k_u,k_v} \theta_{u+k_u,v+k_v} \tag{2.40}$$

The coefficients $\boldsymbol{\theta}$ can now be calculated from the spatial covariance matrix $\mathbf{C}_\phi$ using the FFT-MA generator. Since $c_{\phi_{u,v}}$ tends to zero for large u, v, the index terms k_u, k_v can be seen as finite and are assumed to be $k_u = -\delta, \ldots, \delta$ and $k_v = -\delta, \ldots, \delta$. The wind speed is simulated by generating an over-sized turbulence phase screen and moving a smaller aperture over this phase screen, see Fig. 2.5.

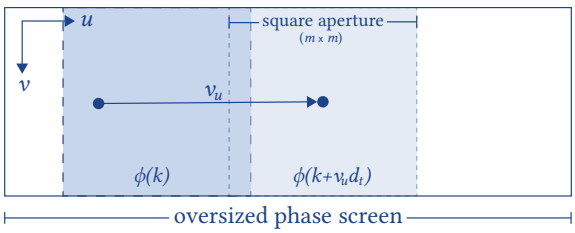


Figure 2.5: Generating an over-sized phase screen over which a smaller square aperture will move. By varying the speed v_u at which this aperture will move, the simulated wind speed is changed.

Two layers of turbulence with different statistics and windspeed are located on conjugated planes and added up to form the wavefront measured by the sensor. The atmospheric turbulence is a stochastic process, therefore 50 realizations are carried out. The default parameters for AO simulations are listed in Table 2.2.

Model	
$N \times N$ WFS sensor points	10×10
SNR sensor noise	15 dB
D aperture diameter	1 m
Turbulence	
Number of layers	2
$m \times m$ turbulence phase screen	31×31
r_0 Fried parameter	$\{0.2, 0.4\}$ m
L_0 outer scale	10 m
δ MA neighborhood	50
Horizontal wind speed	$\{1, 2\}$ pixels/sample
Vertical wind speed	0 pixels/sample

Table 2.2: Parameters for the numerical simulation - QUARKS

Three methods for identification are compared:

1. an unstructured least squares,

$$\min_{\mathbf{A}_i} \sum_{k=p+1}^{N_t} \|\mathbf{s}(k) - \sum_{i=1}^{p} \mathbf{A}_i \mathbf{s}(k-i)\|_2^2 \tag{2.41}$$

2. a regularized sparse least-squares, Kim et al. (2008),

$$\min_{\mathbf{A}_i} \sum_{k=p+1}^{N_t} \|\mathbf{s}(k) - \sum_{i=1}^{p} \mathbf{A}_i \mathbf{s}(k-i)\|_2^2 + \tau \sum_{i=1}^{p} \|\mathrm{vec}(\mathbf{A}_i)\|_1 \tag{2.42}$$

   where τ makes a trade-off between the sparsity of the coefficient matrices and the fit to the data.

3. QUARKS identification (2.27) with Algorithm 2.1. Algorithm 2.1 is initialized only once, randomly. The stopping criterion parameter ϵ is set respectively to 10^{-5}. The maximum number of iterations κ_{max} is 100. The hyperparameters were randomly searched within the bounds mentioned in Section 2.5 and within the range $[0, 5]$ for (λ, μ): the set of hyperparameters over 20 realizations that yields the lowest prediction-error is selected. The curse of dimensionality that appears when choosing hyperparameters with grid search is bypassed with random search, Bergstra and Bengio (2012). Bayesian optimization or online non-linear optimization for hyperparameter estimation are outside the scope of this chapter.

The performance is checked on a validation dataset containing 5×10^3 temporal points. The results are discussed based on the Variance Accounted For (VAF) between the signals $\mathbf{s}(k+1)$ and $\widehat{\mathbf{s}}(k+1)=\sum_{i=1}^{p}\widehat{\mathbf{A}}_i\mathbf{s}(k-i)$:

$$\mathrm{VAF}(\mathbf{s}(k),\widehat{\mathbf{s}}(k))=\max\big(0,\big(1-\frac{\frac{1}{N_t}\sum_{k=p+1}^{N_t}\|\mathbf{s}(k)-\widehat{\mathbf{s}}(k)\|_2^2}{\frac{1}{N_t}\sum_{k=p+1}^{N_t}\|\mathbf{s}(k)\|_2^2}\big)\times100\big) \tag{2.43}$$

The VAF between two identical signals $\mathbf{s}(k)$ and $\widehat{\mathbf{s}}(k)$ reaches 100%. The experiments are carried out on MatlabR2016b using a desktop computer with a CPU Intel Xeon E5-2609.

Illustration of QUARKS identification

The identification set contains 5×10^3 temporal measurements. The temporal order of the VAR model is set to 2. We first choose a Kronecker rank within $\{1,...,5\}$. The parameters λ and μ in (2.27) are set to 0. The minimization (2.42) is solved for τ in the range $\mathrm{logspace}(0,4,8)$.

We define a measure that we call *model complexity* as the number of non-zero entries needed to construct the p coefficient matrices. For example, the complexity of a QUARKS model is at most $2prN^2$ (only the non-zero elements of the factor matrices) whereas it reaches a total of pN^4 for the full least squares estimation. It is illustrated in Figure 2.6 that displays the VAF with respect to the number of non-zero elements (with truncated entries at 1% of the maximum value) needed to construct the full coefficient matrix $\mathbf{A}_1$. The prediction error is computed on a validation dataset *after* truncation. We emphasize that no truncation on the elements of the factor matrices is done for the Kronecker model.

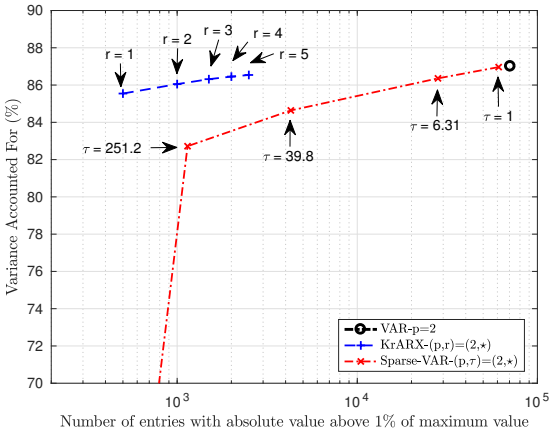


Figure 2.6: Variance Accounted For (%) versus complexity of model. A blue cross corresponds to an estimate with given Kronecker rank. Each red cross corresponds to a regularization parameter τ on the sparsity prior in (2.42). Two points are not visible on the plot: $(\tau,\%\text{non-zero values},\mathrm{VAF})\in\{(1.5849\times10^3,389,45.3),(10^4,124,0)\}$.

For example, a total of 500 non-zero values are necessary to build the Kronecker factors associated to $\mathbf{A}_1$ and reaches 85.54% accuracy. The VAF obtained with the sparse identification decreases with increasing regularization parameter τ as expected while the number of non-zero entries decreases for a high prior on sparsity. This trade-off between the complexity of the model and the accuracy of the prediction error is present in the QUARKS modeling as well. While the estimated matrix with ℓ_1 minimization tries to reduce the number of non-zero entries, the matrix obtained with QUARKS modeling does not exhibit sparse patterns but a prominent multi-level structure. The lower the spatial order r, the lower the model complexity and the higher the prediction error is.

Influence of the hyperparameters

The regularization with r_s and r_t in (2.19) is the most beneficial with short data batches or in noisy environments. The difference with the case $(\lambda, \mu) = (0, 0)$ is all the more significant when the ratio $\frac{N_t}{Nrp}$ is low. The temporal order is set to 4 and the Kronecker rank to 2. There are 500 points in the identification batch. Figure 2.7 displays the VAF on validation data with and without regularization.

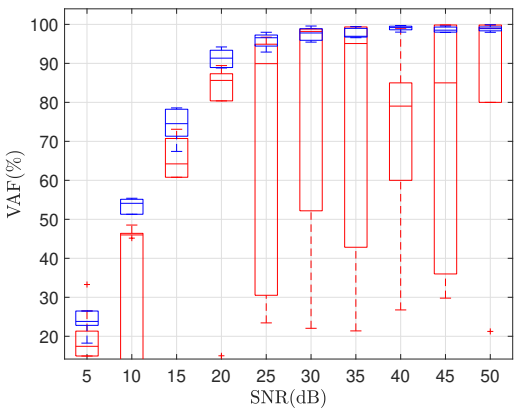


Figure 2.7: Variance Accounted For (%) versus the signal-to-noise ratio. Red: without regularization nor normalization. Blue: with both regularization and normalization.

Regularizing the cost function in noisy situations and with relatively few data samples leads to substantial improvements over the non-regularized QUARKS identification. It especially reduces the variance of the prediction error while the performance of the non-regularized version with few temporal samples is very unreliable. Random search has interesting performances as it exploits the fact that some hyperparameters may not contribute a lot for obtaining good solutions in the example at hand.

Scalability

One advantage of the new modeling paradigm is to reduce the computational complexity for estimating large-scale VARX models. No regularization is considered in this section in order to analyze whether the QUARKS identification in Algorithm 2.1

scales with $\mathcal{O}(N^3 N_t)$. The number of time samples for QUARKS identification is such that $N_t = 10prN$ whereas it is $N_t = 50N^2$ in the unstructured case. These values were fixed such that the prediction-error is similar for both methods. The reduction in the regression coefficient in the linear plots $log10(\text{Time}) = a \times log10(N) + b$ as shown in Figure 2.8 is significant using the QUARKS.

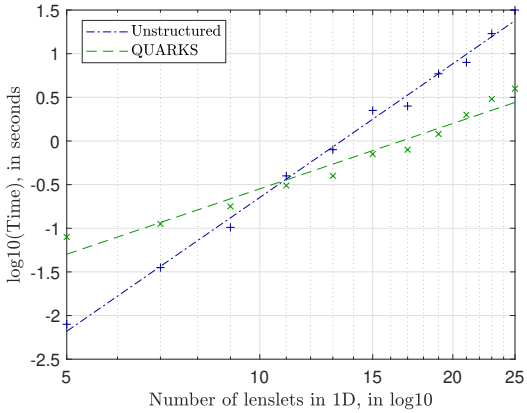


Figure 2.8: Evolution of the computational time with respect to the size of the 2D array. The linear model fitted with the QUARKS method is: $log10(\text{Time}) = 2.55 \times log10(N) - 3.10$, $\sigma = 0.34$ while it is: $log10(\text{Time}) = 5.03 \times log10(N) - 5.68$, $\sigma = 0.27$ with the unstructured least-squares.

2.7. Recursive updates

The computational complexity can be alleviated using recursive algorithms. The temporal order is equal to one in this section only (without loss of generality as later illustrated in numerical experiments).

2.7.1. RLS for updating unstructured VAR models

We now have an estimate of the parameter matrix $\mathbf{A}_1$ at time instant k, denoted with $\widehat{\mathbf{A}_1}(k)$. The variables $\hat{\mathbf{a}}_j(k)$ for all $j = 1 \dots N^2$ denote the rows of the matrix $\widehat{\mathbf{A}_1}$. Whenever a new measurement $\mathbf{s}(k+1)$ becomes available, these estimates are updated. Such an update is the fusion of the prior information and the information about $\mathbf{A}_1$ derived from the new measurements. This fusion can be interpreted as optimizing the following cost function for all the rows $\mathbf{a}_j$ of $\mathbf{A}_1$:

$$\min_{\mathbf{a}_j} \quad \lambda[\mathbf{a}_j - \hat{\mathbf{a}}_j(k)]\mathbf{P}_j(k)^{-1}[\mathbf{a}_j - \hat{\mathbf{a}}_j(k)]^T + (s_j(k+1) - \mathbf{s}(k)^T\mathbf{a}_j^T)^2 \tag{2.44}$$

where λ is a forgetting factor in the interval $]0, 1]$ and $\mathbf{P}_j(k)$ represents the covariance matrix defined as:

$$\mathbf{P}_j(k) = \mathbb{E}[(\mathbf{a}_j - \hat{\mathbf{a}}_j(k))^T(\mathbf{a}_j - \hat{\mathbf{a}}_j(k))]$$

The equation (2.44) is equivalently written as:

$$\begin{aligned} &\min_{\mathbf{a}_j} \quad \boldsymbol{\mu}^T\boldsymbol{\mu} \quad \text{subject to:} \\ &\begin{bmatrix} s_j(k+1) \\ \hat{\mathbf{a}}_j^T(k) \end{bmatrix} = \begin{bmatrix} \mathbf{s}(k)^T \\ \mathbf{I}_N \end{bmatrix} \mathbf{a}_j^T + \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \lambda^{-1/2}\mathbf{P}_j(k)^{1/2} \end{bmatrix} \boldsymbol{\mu}(k) \end{aligned} \tag{2.45}$$

where $\boldsymbol{\mu}(k)$ is a Gaussian noise with mean zero and identity covariance matrix. The solution to this least squares problem is given by the following recursive equations:

$$\begin{aligned} \hat{\mathbf{a}}_j(k+1)^T &= \hat{\mathbf{a}}_j(k)^T + \mathbf{g}_j(k+1)[s_j(k+1)^T - \mathbf{s}(k)^T\hat{\mathbf{a}}_j(k)^T] \\ \mathbf{g}_j(k+1) &= \lambda^{-1}\mathbf{P}_j(k)\mathbf{s}(k)[1 + \lambda^{-1}\mathbf{s}(k)^T\mathbf{P}_j(k)\mathbf{s}(k)]^{-1} \\ \mathbf{P}_j(k+1) &= \lambda^{-1}[\mathbf{P}_j(k) - \mathbf{g}_j(k+1)\mathbf{s}(k)^T\mathbf{P}_j(k)] \end{aligned} \tag{2.46}$$

If $\mathbf{P}_j(0)$ is chosen identical for all j, then $\mathbf{g}_j(k)$ and $\mathbf{P}_j(k)$ are independent of j and can be written as $\mathbf{g}(k)$ and $\mathbf{P}(k)$. We summarize the algorithm updating the estimates of $\mathbf{a}_j(k)$ and $\mathbf{P}(k)$ in a computationally efficient manner in Algorithm 2.2. Transposing the scalar value $s_j(k+1)$ on line 4 has no effect here, in the following section this value will be a vector which makes the transposition necessary. The initial estimate for $\mathbf{A}_1$ can be determined by doing an initial offline identification step or can be set to a random matrix. The initial value for the matrix $\mathbf{P}$ is usually set to $\delta\mathbf{I}_{N^2}$ where δ is a design parameter. Choosing δ depends on how much confidence is placed in the initial guess of $\mathbf{A}_1$, e.g a low value for δ means that a high amount of confidence in the initial guess.

Algorithm 2.2: RLS

Input : $\mathbf{s}(k+1), \mathbf{s}(k), \widehat{\mathbf{A}_1}(k-1), \mathbf{P}(k-1), \lambda, N$
Output : $\widehat{\mathbf{A}_1}(k), \mathbf{P}(k)$

$\mathbf{g}(k) = \lambda^{-1}\mathbf{P}(k-1)\mathbf{s}(k)[1 + \lambda^{-1}\mathbf{s}(k)^T\mathbf{P}(k-1)\mathbf{s}(k)]^{-1}$
$\mathbf{P}(k) = \lambda^{-1}[\mathbf{P}(k-1) - \mathbf{g}(k)\mathbf{s}(k)^T\mathbf{P}(k-1)]$
for $j = 1 \dots N$ **do**
  $\hat{\mathbf{a}}_j(k)^T = \hat{\mathbf{a}}_j(k-1)^T + \mathbf{g}(k)[s_j(k+1)^T - \mathbf{s}(k)^T\hat{\mathbf{a}}_j(k-1)^T]$
end

2.7.2. RLS for QUARKS models

We now address the question whether this scheme can be adapted to recursively update a QUARKS model. Let a QUARKS model with both p and r equal to one: $\mathbf{S}(k) = \mathbf{CS}(k-1)\mathbf{B} + \mathbf{E}(k)$. The particularity here is that there are two matrices that need to be updated, thus creating a bilinear least-squares problem with no closed-form solution. A similar problem was tackled for estimating recursively bilinear systems in the case where $\mathbf{B}$ and $\mathbf{C}$ are vectors in Wang et al. (2016). The initial estimate of the matrix $\mathbf{C}$, denoted as $\widehat{\mathbf{C}}(k)$, is used to update the estimate of $\mathbf{B}$, denoted as $\widehat{\mathbf{B}}(k+1)$. The factor $\mathbf{C}$ is then updated by fixing the previously obtained estimate for $\mathbf{B}$, resulting in one ALS update. For each time step k, the factor matrices $\mathbf{B}$ and $\mathbf{C}$ are updated once. A schematic is presented in Fig. 2.9.

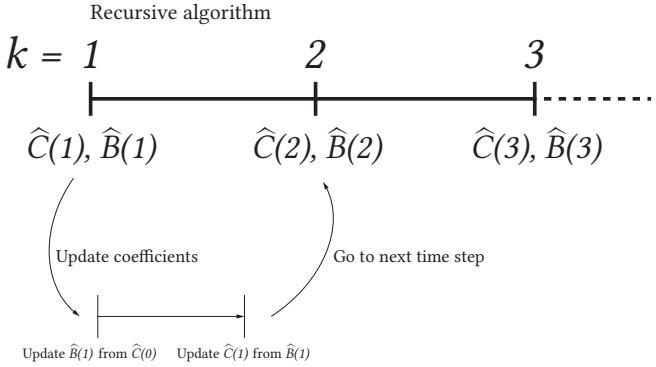


Figure 2.9: Showing a time line of the QUARKS-RLS algorithm. Every time step k, we calculate estimates $\widehat{\mathbf{C}}(k)$ and $\widehat{\mathbf{B}}(k)$ using two alternating steps.

Similar to the previous section, we partition $\widehat{\mathbf{B}}(k), \widehat{\mathbf{C}}(k)$ and $\mathbf{S}(k)$ as follows:

$$\widehat{\mathbf{C}}(k) = \begin{bmatrix} \hat{\mathbf{c}}_1(k) \\ \vdots \\ \hat{\mathbf{c}}_N(k) \end{bmatrix} \qquad \widehat{\mathbf{B}}(k) = \begin{bmatrix} \hat{\mathbf{b}}_1(k) & \cdots & \hat{\mathbf{b}}_N(k) \end{bmatrix}$$

$$\mathbf{S}(k) = \begin{bmatrix} \mathbf{s}_1(k) & \cdots & \mathbf{s}_N(k) \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{s}}_1(k) \\ \vdots \\ \bar{\mathbf{s}}_N(k) \end{bmatrix}$$

The variables $\mathbf{s}_j(k)$ and $\bar{\mathbf{s}}_j(k)$ are now vectors instead of scalars as in the previous subsection. We moreover introduce the new variable $\mathbf{U}_c(k)$ as $\mathbf{U}_c(k) = \widehat{\mathbf{C}}(k)\mathbf{S}(k)$, and consider the following problem for updating the estimate of the columns $\mathbf{b}_j(k)$ of the matrix $\mathbf{B}(k)$ using $\widehat{\mathbf{C}}(k)$ inspired by the solution of the previous section:

$$\min_{\mathbf{b}_j} \quad \lambda^{-1}[\mathbf{b}_j - \hat{\mathbf{b}}_j(k)]^T \mathbf{P}_b^{-1}(k)[\mathbf{b}_j - \hat{\mathbf{b}}_j(k)] + ||\mathbf{s}_j(k+1) - \mathbf{U}_c(k)\mathbf{b}_j||_2^2 \tag{2.47}$$

The solution to (2.47) is provided by Algorithm 2.2 and can be written as:

$$[\widehat{\mathbf{B}}(k), \mathbf{P}_b(k)] = RLS(\mathbf{S}(k+1), \mathbf{U}_c(k), \widehat{\mathbf{B}}(k-1), \mathbf{P}_b(k-1), \lambda, N)$$

The second step of the Alternating Least Squares consists of updating $\mathbf{C}(k)$ based on the estimate $\widehat{\mathbf{B}}(k)$:

$$\min_{\mathbf{c}_j} \quad \lambda^{-1}[\mathbf{c}_j - \hat{\mathbf{c}}_j(k)]\mathbf{P}_c^{-1}(k)[\mathbf{c}_j - \hat{\mathbf{c}}_j(k)]^T + ||\bar{\mathbf{s}}_j(k+1)^T - \mathbf{U}_b(k)\mathbf{c}_j^T]||_2^2 \tag{2.48}$$

where $\mathbf{U}_b(k) = \widehat{\mathbf{B}}(k)^T \mathbf{S}(k)^T$. The solution to (2.48) is obtained by running Algorithm 2.2 with the following parameters:

$$[\widehat{\mathbf{C}}(k)^T, \mathbf{P}_c(k)] = RLS(\mathbf{S}(k+1)^T, \mathbf{U}_b(k), \widehat{\mathbf{C}}(k-1)^T, \mathbf{P}_c(k-1), \lambda, N)$$

Applying the ALS algorithm for updating the matrices $\widehat{\mathbf{B}}(k)$ and $\widehat{\mathbf{C}}(k)$ in the minimization problems in (2.47) and (2.48) results in the RLS algorithm for QUARKS models as defined in Algorithm 2.3.

Algorithm 2.3: QUARKS-RLS Algorithm

$\mathbf{P}_b(0) = \delta \mathbf{I}_N, \mathbf{P}_c(0) = \delta \mathbf{I}_N$
for $1 \leq k < N_t$ **do**
  $\mathbf{U}_c(k) = \widehat{\mathbf{C}}(k)\mathbf{S}(k)$
  $[\widehat{\mathbf{B}}(k), \mathbf{P}_b(k)] = RLS(\mathbf{S}(k+1), \mathbf{U}_c(k), \widehat{\mathbf{B}}(k-1), \mathbf{P}_b(k-1), \lambda, N)$
  $\mathbf{U}_b(k) = \widehat{\mathbf{B}}(k)^T \mathbf{S}(k)^T$
  $[\widehat{\mathbf{C}}(k)^T, \mathbf{P}_c(k)] = RLS(\mathbf{S}(k+1)^T, \mathbf{U}_b(k), \widehat{\mathbf{C}}(k-1)^T, \mathbf{P}_c(k-1), \lambda, N)$
end

By performing these steps at each new time step k, we obtain a recursive least squares algorithm for low-Kronecker rank structured models. The initial guess for the coefficient matrices $\widehat{\mathbf{B}}(0), \widehat{\mathbf{C}}(0)$ is obtained using the QUARKS.

2.7.3. Computational complexity

Recursive algorithms do not store the whole data batch but only the last measurement and the matrices $\mathbf{P}_b, \mathbf{P}_c$. This makes them attractive even for offline use on large-scale stationary data. The online computational complexity to update the model is constrained by the frequency of operation of the system and reducing it is the main target.

Lemma 2.7. *The computational complexity for updating recursively a QUARKS model is $\mathcal{O}(N^3)$ compared to $\mathcal{O}(N^4)$ in the unstructured case.*

Proof. When considering the RLS equations in Algorithm 2.2 without the Kronecker structure, the most computationally complex operation is a matrix-vector-multiplication $\mathbf{P}(k)\mathbf{s}(k)$. The complexity is $\mathcal{O}(N^4)$. For the RLS equations using the Kronecker structured matrices in Algorithm 2.3, the most complex operation is a matrix-matrix-multiplication $\mathbf{P}_b(k)\mathbf{S}(k)$ scaling with $\mathcal{O}(N^3)$. ■

2.8. Numerical examples: recursive updates

The algorithm is validated using an application to AO using synthetic and validation data.

2.8.1. Synthetic data

The wind speed is simulated by generating an over-sized turbulence phase screen and moving a smaller aperture over this phase screen, see Fig. 2.5. In order to create non-stationary turbulence, the wind speed varies by moving the aperture over the phase screen at a piece-wise constant speed v_u. More specifically, we divide the simulation in a number of time sections of equal length, each of which has constant wind speed in each section. In this simulation, we use the piece-wise constant wind speed distribution in the horizontal direction: [4 1 3 9 5] pixels/sample with a simulation duration of $20 \cdot 10^3$ samples. Each piece-wise constant section consists of $4 \cdot 10^3$ samples such that an over-sized phase screen of size $(4 \cdot 10^3 \cdot (4+1+3+9+5)+m) \times m$ is generated.

Two datasets are generated under the same atmospheric conditions, see Table 2.3. For the non-recursive methods, one dataset is used for offline identification and the other one is used for validation. The first dataset is used to generate a starting value for the recursive identification methods. We perform 100 Monte-Carlo simulations. The number of iterations needed for ALS to achieve a difference in residual less than 10^{-3} is determined with and without normalization over 100 simulations. The average number of iterations needed with normalization is 8.53 and without normalization 7.16.

In Figure 2.10, the accuracy of the estimates of the coefficient matrices over an entire simulation duration is shown. The VAF is computed for each simulation. The mean and standard deviation over all Monte-Carlo simulations are then calculated and represented with the shaded area. In red, the accuracy of the recursive QUARKS algorithm can be seen, compared to the accuracy of the non-recursive QUARKS algorithm (blue) when the identification is performed on the whole dataset assuming

Model	
$N \times N$ WFS sensor points	9×9
SNR sensor noise	20 dB
D aperture diameter	1 m
Turbulence	
$m \times m$ turbulence phase screen	28×28
r_0 Fried parameter	0.2 m
L_0 outer scale	10 m
δ MA neighborhood	50
Horizontal wind speed	[4 1 3 9 5] pixels/sample
Vertical wind speed	0 pixels/sample
Identification data set	
N_t phase samples	10×10^3
N_a ALS iterations	20
Simulation data set	
N_t phase samples	20×10^3
λ forgetting parameter	0.9988
$\mathbf{P}(1)$ initial value	$\mathbf{I}_N$

Table 2.3: Parameters for the numerical simulation - recursive QUARKS

the latter is stationary. The recursive estimation with the scalable method proposed in subsection 2.7.2 reaches higher performances when the temporal dynamics have reached stationarity. In green, the VAF is plotted for a QUARKS fixed sliding window (FSW) model. This is obtained by estimating a QUARKS model at each new time sample, with the regression data within a fixed sliding window containing the last 200 time samples. This last method is an upper bound on the accuracy that can be achieved with the QUARKS method for non-stationary modelling at the expense of a much higher computational cost as a new ALS is solved at each time sample. The number of ALS iterations to meet the stopping criterion however, is very small when the atmospheric conditions are slowly time-varying. In purple, the recursive QUARKS algorithm is shown where normalization is applied to one of the factor matrices during recursive estimation. It shows that using normalization with recursive QUARKS decreases the rate of convergence and slightly decreases the overall accuracy of the algorithm.

We now investigate the *online* computational complexity in Figure 2.11 with timing experiments on different sizes of sensor. Online, $2pr$ matrix-matrix multiplications (MMM) are required for the QUARKS; Algorithm 2.3 and $2pr$ MMM are required for QUARKS-RLS; solving the QUARKS and $2pr$ MMM are required for QUARKS-FSW. A linear model $log10(Time) = a \times log10(N) + b$ was fitted to the timing data and we are particularly interested in the parameter a as it indicates how well the method scales with increasing size of the sensor. The lower a, the better the scalability. The online computational complexity for QUARKS, QUARKS-RLS and QUARKS-FSW scale theoretically with $\mathcal{O}(N^3)$, and regression coefficients a of respectively $1.56, 1.54, 2.33$ are obtained. Although the size of the temporal window

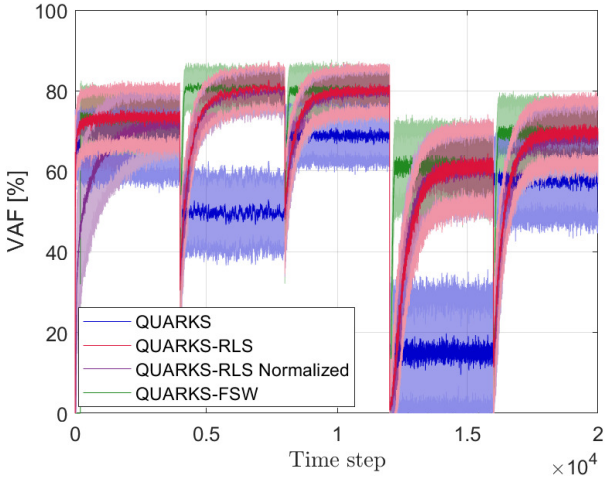


Figure 2.10: Comparing the spatial VAF for QUARKS-RLS (red), fixed sliding window (FSW) for QUARKS (green) and non-recursive QUARKS model over the entire simulation duration assuming non-stationary turbulence (blue). The standard deviation for each of the three method is shaded.

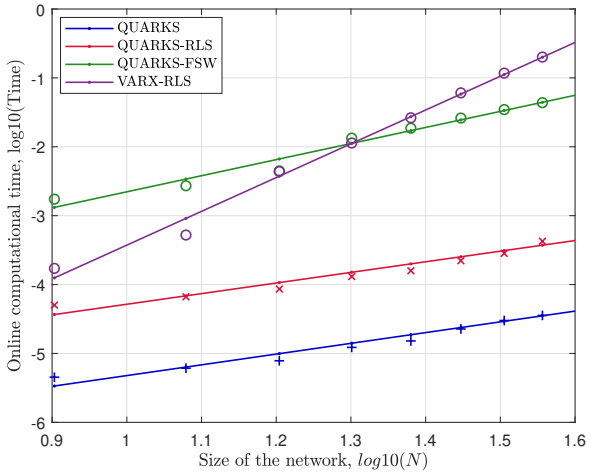


Figure 2.11: Timings experiments for QUARKS (blue), recursive QUARKS (red), QUARKS with a fixed sliding window (green) and unstructured VARX-RLS (purple) for different sensor sizes. The coefficients of the models are: for QUARKS, $(a, b) = (1.56, -6.88)$; for QUARKS-RLS, $(a, b) = (1.54, -5.83)$; for QUARKS-FSW with length of sliding window 200, $(a, b) = (2.33, -4.99)$ and for VARX-RLS, $(a, b) = (4.90, -8.33)$.

for QUARKS-FSW is constant over N, it still shows lower scalability than QUARKS and QUARKS-RLS. Furthermore, the regression coefficient for the unstructured RLS is 4.90 and hence, a relative difference of 3.36 with QUARKS-RLS.

2.8.2. Laboratory validation

We now consider the AO laboratory setup used to test the proposed identification approach. A schematic is shown in Figure 2.12.

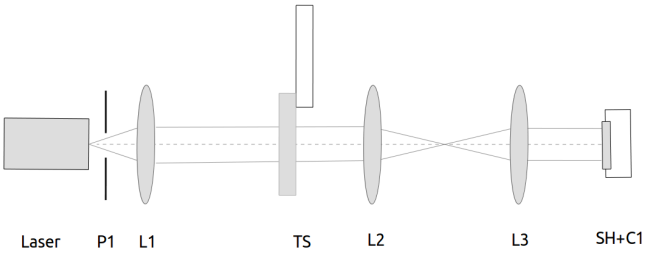


Figure 2.12: Schematic of the laboratory testbed. The light emitted from the laser goes through the pupil P1 and the lens L1. It is collimated when reaching the turbulence plate TS, that is placed at the focal plane of the lens L2. The lens L3 conjugates TS with the sensor SH+C1.

The light is emitted from a laser source ($\lambda = 635$nm) and is then collimated into a beam of size $D = 9$mm using the lens L1. The atmospheric turbulence for a single frozen layer is generated using a pseudo-random phase plate TS machined by Lexitek, Inc. The optical path difference is defined as follows. A phase design that follows the spatial Kolmogorov distribution is generated and is then multiplied by a factor that varies with angle from the center of the array equal to $(1 + 1/5 sin\theta)^{-5/6}$. The effect is to produce a phase design where the local value of the Fried parameter r_0 varies as $(1 + 1/5 \cdot sin\theta)$. Different wind speed conditions are simulated by rotating the disk. The speed changes every 50 time samples, as the speed in Rounds Per Minute (RPM) is set with,

$$\text{RPM}(k) = \frac{1}{2} + \frac{T}{6} \sin\left(\frac{2\pi}{N_t} 50 \lfloor k/50 \rfloor\right) \tag{2.49}$$

for $T \in \{1, 1.5, 2, 2.5, 3\}$.

The beam goes through the turbulence disk that is placed at the focal plane of the lens L2, $f_1 = 10$cm. The lens L3 has a focal length of 10cm and forms a telescope with L2. An OKOtech Shack-Hartmann wavefront sensor SH+C1, 1-inch optical format, with a lenslet array pitch of 300μm and focal length 18.6mm, is placed perpendicular to the optical beam path at the focal point of L3. The turbulence phase profile and the grid of lenslets are in conjugated planes. An array of 28×28 lenslets is selected among which 566 are illuminated and considered as active. The Kronecker structure does not adapt well to circular apertures and we consider the rectangular aperture of the active lenslets. The slopes signals corresponding to the non-active lenslets are set to 0. Such an approximation implies larger prediction errors at the boundary of the pupil, although this effect is all the more mitigated if the factor matrices are sparse. At each time sample, the non-zero values predicted outside the circular aperture are set to 0. The sampling frequency is $f_S = 12.5$Hz.

The Greenwood per sample frequency ratio (1.34) is upper-bounded with 0.14, which is well below the Nyquist criteria, and hence no temporal aliasing occurs.

We collect $N_t = 3 \times 10^3$ samples for each value of T. A number of 0.5×10^3 samples is used for a batch-wise identification of a QUARKS model, which serves as an initial guess for obtaining temporally varying estimates on the next 2.5×10^3 samples.

The Kronecker rank and the temporal order both take values within the set $\{1, 3\}$. The accuracy is measured by calculating *for each lenslet* the VAF (averaged in both the horizontal and vertical direction of the slopes signal) between the true signal and the reconstructed signal. Such a measurement is different from the previous subsection in which the VAF was computed spatially for each time sample. We compare the accuracy of the QUARKS algorithm with the recursive QUARKS-RLS algorithm for varying conditions of non-stationarity in Table 2.4. The relative improvement between the VAF for the QUARKS-RLS and the VAF for the QUARKS with $p = 3$ is indicated as *Ratio* in Table 2.4.

	QUARKS-RLS		QUARKS		Diag-RLS	Ratio
T, $r = 1$	$p = 1$	$p = 3$	$p = 1$	$p = 3$	$p = 1$	(%)
1	81.15	84.41	79.06	78.10	51.98	8.08
1.5	81.26	84.18	78.36	77.16	52.45	9.09
2	80.04	82.86	73.48	72.09	52.75	14.9
2.5	79.77	81.18	71.10	70.48	52.32	15.2
3	79.76	81.01	65.76	64.53	50.51	25.5
T, $r = 3$						
1	82.23	84.42	80.02	78.65		7.34
1.5	82.08	84.50	80.84	77.87		8.51
2	80.85	83.54	73.10	72.53		15.1
2.5	80.22	83.12	71.47	71.05		17.0
3	80.91	80.83	65.66	65.76		22.9

Table 2.4: Laboratory testbed experiment: VAF (%)

In this AO configuration with one turbulence disk and at relatively low Greenwood per sample frequency ratio, increasing the temporal order or the Kronecker rank of the model leads to little improvements. Moreover, when increasing the amplitude T of the sine function, and hence the non-stationarity, the recursive algorithm is better equipped to handle the large changes induced by the varying rotational speed of the turbulence disk. The accuracy does decrease relatively less compared to the non-recursive case.

2.9. Conclusion

Each coefficient matrix of the VAR model is parametrized with a sum of few Kronecker matrices which offers high data compression for large sensor grids. Estimating in least-squares sense the data matrices gives rise to a bilinear problem which is addressed using Alternating Least Squares. The convergence of the estimates to a fixed point was proven in very particular conditions and assuming persistency of excitation and non-zero initial guesses.

The QUARKS may also be identified in a recursive fashion to deal with non-stationary data or simply to reduce the memory complexity. Importantly, it alleviates the memory burden on QUARKS in so far as only the left and right covariance matrices along with the last measurement sample need to be stored, as opposed to the whole dataset. A numerical validation for adaptive optics purposes was proposed on synthetic and laboratory testbed data. Although the discussion has dealt with a temporal order and a Kronecker rank equal to one, the algorithms generalise as shown in the experimental section.

The algorithm has been presented for 2D dynamical systems and can be generalized to higher dimensions by using a Kronecker product of multiple matrices instead of only two matrices in which case larger compression rates are achieved. Such higher order modeling for 2D arrays is obtained by tensorizing the sensor data $\mathbf{S}(k)$ and allows to establish a new trade-off between accuracy and computational complexity. It will be described further in Chapter 4 and 6.

Appendix. Proof for Theorem 2.1

In this appendix, we derive the proof of convergence for the ALS with a very particular normalization. The proof builds on Li et al. (2015) and only the main changes compared to the vector form are highlighted.

Notations. The noise term $\widetilde{\mathbf{E}}$ is defined similarly as $\widetilde{\mathbf{S}}$ from the noise components $\mathbf{e}(k)$. Moreover, $\widetilde{\mathbf{s}} = \text{vec}(\widetilde{\mathbf{S}}), \widetilde{\mathbf{e}} = \text{vec}(\widetilde{\mathbf{E}}), \widetilde{\mathbf{U}} = \widetilde{\mathbf{U}}_1, \mathbf{M} = \mathbf{I}_N \otimes \widetilde{\mathbf{U}}$. The iteration counter κ is left out. The notation $\lambda_{max}(\mathbf{X})$ is used for the spectral radius of the matrix $\mathbf{X}$.

First, an inner product for matrices in $\mathbb{R}^{(N(N_t-1)+N^3)\times N}$ is defined.

Definition 2.4. *Let $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{N\times N}$ and denote their columns with $\mathbf{x}_i, \mathbf{y}_i$. For two matrices $\overline{\mathbf{X}}, \overline{\mathbf{Y}}$ such that:*

$$\overline{\mathbf{X}} = \mathbf{M} \begin{bmatrix} \mathbf{I}_N \otimes \mathbf{x}_1 \\ \vdots \\ \mathbf{I}_N \otimes \mathbf{x}_N \end{bmatrix}$$

and similarly for $\overline{\mathbf{Y}}$, the inner product on $\mathbb{R}^{(N(N_t-1)+N^3)\times N}$ is defined with:

$$\langle \overline{\mathbf{X}}, \overline{\mathbf{Y}} \rangle = \lambda_{max}(\widetilde{\mathbf{U}}^T \widetilde{\mathbf{U}}) \text{vec}(\mathbf{X})^T \text{vec}(\mathbf{Y})$$

Lemma 2.8. *For the matrix* $\overline{\mathbf{X}} \in \mathbb{R}^{(N(N_t-1)+N^3)\times N}$ *and the inner product in Definition 2.4, the quantity* $\|\overline{\mathbf{X}}\|_2 = \sqrt{\langle \overline{\mathbf{X}}, \overline{\mathbf{X}} \rangle}$ *is a norm.*

Proof. The proof contains four points.

1. $\|\overline{\mathbf{X}}\|_2$ is positive because the spectral radius and the Euclidean norm are both positive.
2. If $\|\overline{\mathbf{X}}\|_2 = 0$, and with $\widetilde{\mathbf{U}}^T \widetilde{\mathbf{U}} \neq 0$, then $\|\mathbf{x}\|_2 = 0$ and $\mathbf{x} = 0$. This implies that $\overline{\mathbf{X}} = 0$.
3. Let $\alpha \in \mathbb{R}$. $\|\alpha\overline{\mathbf{X}}\|_2^2 = \lambda_{max}\big(\alpha^2(\widetilde{\mathbf{U}}^T \widetilde{\mathbf{U}})\big)\|\mathbf{x}\|_2^2 = |\alpha|\|\mathbf{X}\|_2^2$
4. The triangular inequality reads:

$$\|\overline{\mathbf{X}} + \overline{\mathbf{Y}}\|_2 = \sqrt{\lambda_{max}(\widetilde{\mathbf{U}}^T \widetilde{\mathbf{U}})}\|\mathbf{x}+\mathbf{y}\|_2 \leq \|\overline{\mathbf{X}}\|_2 + \|\overline{\mathbf{Y}}\|_2$$

using the triangular inequality on the Euclidean norm. ■

For example, the matrix $\overline{\mathbf{X}}_b$ has the structure of $\overline{\mathbf{X}}$ in Definition 2.4. We define two sets associated to the true values $\boldsymbol{a}$ and $\boldsymbol{b}$:

$$\mathcal{X}_{\boldsymbol{a}} = \{\widehat{\boldsymbol{a}} \in \mathbb{R}^{N^2} | \forall i \in \{1,...,N\}, \|\widehat{\mathbf{a}}_i\|_2 \leq \|\mathbf{a}_i\|_2\} \quad (2.50)$$

$$\mathcal{X}_{\boldsymbol{b}} = \{\widehat{\boldsymbol{b}} \in \mathbb{R}^{N^2} | \forall i \in \{1,...,N\}, \|\widehat{\mathbf{b}}_i\|_2 = \|\mathbf{b}_i\|_2, \widehat{b}_1 > 0\} \quad (2.51)$$

Let $\widehat{\boldsymbol{a}} \in \mathcal{X}_{\mathbf{a}}, \widehat{\boldsymbol{b}} \in \mathcal{X}_{\mathbf{b}}$.

$\mathcal{F}$ maps $\mathcal{X}_a$ to $\mathcal{X}_a$

The solution of one least-squares update is written with:

$$\begin{aligned} \widehat{\boldsymbol{a}} &= \mathcal{F}_3(\widehat{\mathbf{b}}) \\ &= \big(\mathbf{I}_N \otimes (\overline{\mathbf{X}}_{\widehat{\mathbf{b}}}^T \overline{\mathbf{X}}_{\widehat{b}})^{-1}\overline{\mathbf{X}}_{\widehat{b}}^T\big)\widetilde{\mathbf{s}} \\ &= \big(\mathbf{I}_N \otimes (\overline{\mathbf{X}}_{\widehat{b}}^T \overline{\mathbf{X}}_{\widehat{b}})^{-1}\overline{\mathbf{X}}_{\widehat{b}}^T\big)\big((\mathbf{I}_N \otimes \overline{\mathbf{X}}_b)\boldsymbol{a} + \widetilde{\mathbf{e}}\big) \end{aligned} \quad (2.52)$$

Using the partition of $\mathbf{a}$ into the N vectors $\mathbf{a}_i$ of size N, we write (2.52):

$$\widehat{\boldsymbol{a}}_i = (\overline{\mathbf{X}}_{\widehat{b}}^T \overline{\mathbf{X}}_{\widehat{b}})^{-1}\overline{\mathbf{X}}_{\widehat{b}}^T \overline{\mathbf{X}}_b \boldsymbol{a}_i + \mathbf{e}_i \quad (2.53)$$

which corresponds to the vector form studied in Li et al. (2015). We assumed the noise has a finite variance when N_t goes to infinity which implies:

$$\lim_{N_t\to\infty} \|(\overline{\mathbf{X}}_{\widehat{b}}^T \overline{\mathbf{X}}_{\widehat{b}})^{-1}\overline{\mathbf{X}}_{\widehat{b}}^T\|_2 \|\mathbf{e}_i\|_2 = 0 \quad (2.54)$$

Therefore, the Euclidean norm of $\widehat{\boldsymbol{a}}_i$ is upper-bounded as follows:

$$\lim_{N_t\to\infty} \|\widehat{\boldsymbol{a}}_i\|_2 \leq \lim_{N_t\to\infty} \|(\overline{\mathbf{X}}_{\widehat{b}}^T \overline{\mathbf{X}}_{\widehat{b}})^{-1}\overline{\mathbf{X}}_{\widehat{b}}^T \overline{\mathbf{X}}_b\|_2 \|\boldsymbol{a}_i\|_2 \quad (2.55)$$

$$\leq \lim_{N_t \to \infty} \frac{\|\overline{\mathbf{X}}_{\widehat{b}}^T \overline{\mathbf{X}}_b\|_2}{\|\overline{\mathbf{X}}_{\widehat{b}} \overline{\mathbf{X}}_{\widehat{b}}\|_2} \|\boldsymbol{a}_i\|_2$$

$$\leq \lim_{N_t \to \infty} \frac{\|\widehat{\mathbf{b}}^T \mathbf{b}\|_2}{\|\widehat{\mathbf{b}}^T \widehat{\mathbf{b}}\|_2} \|\boldsymbol{a}_i\|_2 \tag{2.56}$$

The last inequality is obtained using the definition of the inner product in Lemma 2.8. We conclude with the following lemma.

Lemma 2.9. *Let* $\mathbf{b}, \widehat{\mathbf{b}} \in \mathbb{R}^N$. *If* $\|\widehat{\boldsymbol{b}}\|_2 = \|\boldsymbol{b}\|_2$, *then* $\|\widehat{\boldsymbol{b}}^T \boldsymbol{b}\|_2 \leq \|\widehat{\boldsymbol{b}}^T \widehat{\boldsymbol{b}}\|_2$. *The inequality is strict if* $\widehat{\boldsymbol{b}} \neq \epsilon \boldsymbol{b}$ *for* $\epsilon \in \{-1, 1\}$.

$\widehat{\boldsymbol{b}} \in \mathcal{X}_{\boldsymbol{b}}$ implies $\|\widehat{\boldsymbol{b}}\|_2 = \|\boldsymbol{b}\|_2$ and therefore, $\|\widehat{\boldsymbol{a}}_i\|_2 \leq \|\boldsymbol{a}_i\|_2$ when N_t goes to infinity. The functional $\mathcal{F}$ maps $\mathcal{X}_{\boldsymbol{a}}$ to $\mathcal{X}_{\boldsymbol{a}}$.

Upper bound on Q

We now introduce the Lispchitz constant[2] $Q = \|\frac{d\mathcal{F}}{d\widehat{\boldsymbol{a}}}\|_2$. From $\widehat{\boldsymbol{a}}^{(\kappa+1)} = \mathcal{F}_3(\mathcal{F}_2(\mathcal{F}_1(\widehat{\boldsymbol{a}}^{(\kappa)})))$, we decompose:

$$Q = \left\| \frac{d\mathcal{F}}{d\widehat{\boldsymbol{b}}} \frac{d\widehat{\boldsymbol{b}}}{d\widehat{\boldsymbol{b}}_{\mathbf{n}}} \frac{d\widehat{\boldsymbol{b}}_{\mathbf{n}}}{d\widehat{\boldsymbol{a}}} \right\|_2 \leq \left\| \frac{d\mathcal{F}_3}{d\widehat{\boldsymbol{b}}} \right\|_2 \left\| \frac{d\mathcal{F}_2}{d\widehat{\boldsymbol{b}}_{\mathbf{n}}} \right\|_2 \left\| \frac{d\mathcal{F}_1}{d\widehat{\boldsymbol{a}}} \right\|_2 \tag{2.57}$$

We further detail each norm in (2.57) and start the analysis with $\|\frac{d\mathcal{F}_3}{d\widehat{\boldsymbol{b}}}\|_2$.

Lemma 2.10. *(Li et al. (2015))Let* $f(.)$ *be defined with* $f(\widehat{\boldsymbol{b}}) := \mathbf{I}_N \otimes (\overline{\mathbf{X}}_{\widehat{b}}^T \overline{\mathbf{X}}_{\widehat{b}})^{-1} \overline{\mathbf{X}}_{\widehat{b}}^T$. *Under Assumption* ***A2****, the magnitude of the directional derivative of* $f(\widehat{\boldsymbol{b}})$ *along a vector* $\mathbf{u}$ *attains its maximum when* $\mathbf{u}$ *is in the same direction as* $\widehat{\boldsymbol{b}}$.

When taking the derivative of f with respect to $\widehat{\boldsymbol{b}}$, the maximum norm is obtained when the gradient is taken along the direction of $\widehat{\boldsymbol{b}}$, i.e a deviation from $\boldsymbol{b}$, denoted with $\Delta\mathbf{b}$, is in the same direction as $\widehat{\boldsymbol{b}}$. Using the derivations from the previous section and introducing a normalized deviation $\overrightarrow{\boldsymbol{b}}$ equal to $\frac{\Delta\mathbf{b}}{\|\Delta\mathbf{b}\|_2}$:

$$\left\| \frac{d\mathcal{F}_3}{d\widehat{\boldsymbol{b}}} \right\|_2 \leq \frac{\|\overrightarrow{\boldsymbol{b}}^T \boldsymbol{b}\|_2}{\|\widehat{\boldsymbol{b}}^T \widehat{\boldsymbol{b}}\|_2} \|\boldsymbol{a}\|_2 \tag{2.58}$$

From the definition of the unit vector $\overrightarrow{\boldsymbol{b}}$, it can be expressed as a function of $\widehat{\boldsymbol{b}}$ with $\|\widehat{\boldsymbol{b}}^T \boldsymbol{b}\|_2 = \|\overrightarrow{\boldsymbol{b}}^T \boldsymbol{b}\|_2 \|\boldsymbol{b}\|_2$. Then, (2.58) is written as:

$$\left\| \frac{d\mathcal{F}_3}{d\widehat{\boldsymbol{b}}} \right\|_2 \leq \frac{\|\widehat{\boldsymbol{b}}^T \boldsymbol{b}\|_2}{\|\widehat{\boldsymbol{b}}^T \widehat{\boldsymbol{b}}\|_2} \frac{\|\boldsymbol{a}\|_2}{\|\boldsymbol{b}\|_2} \tag{2.59}$$

[2]We recall that for a function $f : \mathbb{R} \to \mathbb{R}$ continuous and differentiable, if $|f'(x)| \leq M$, then f is Lipschitz with Lipschitz constant M. The derivative form for Q that we present here differs from the inequality presented in Lemma 2.5.

Now evaluating the derivative of $\mathcal{F}_2$ related to the normalization step, we write:

$$\left\| \frac{d\mathcal{F}_2}{d\widehat{\boldsymbol{b}}_{\mathbf{n}}} \right\|_2 = \left\| \frac{d\widehat{\boldsymbol{b}}}{d\widehat{\boldsymbol{b}}_{\mathbf{n}}} \right\|_2 \leq \frac{\|\boldsymbol{b}\|_2}{\|\widehat{\boldsymbol{b}}_{\mathbf{n}}\|_2} \tag{2.60}$$

We need to relate $\|\boldsymbol{b}\|_2$ and $\|\widehat{\boldsymbol{b}}_{\mathbf{n}}\|_2$.

Lemma 2.11. *For all* $i \in \{1, ..., N\}$, $\|\widehat{\boldsymbol{a}}_i\|_2 = \|\boldsymbol{a}_i\|_2$ *and* $\|\widehat{\mathbf{b}_{\mathbf{n}i}}\|_2 = \|\boldsymbol{b}_i\|_2$.

Proof. Asymptotically,

$$\widehat{\boldsymbol{a}} = \left(\mathbf{I}_N \otimes (\overline{\mathbf{X}}_{\widehat{b}}^T \overline{\mathbf{X}}_{\widehat{b}})^{-1} \overline{\mathbf{X}}_{\widehat{b}}^T \overline{\mathbf{X}}_b\right) \boldsymbol{a}$$

and therefore, for all $i \in \{1, \ldots, N\}$:

$$\widehat{\boldsymbol{a}}_i = (\overline{\mathbf{X}}_{\widehat{b}}^T \overline{\mathbf{X}}_{\widehat{b}})^{-1} \overline{\mathbf{X}}_{\widehat{b}}^T \overline{\mathbf{X}}_b \boldsymbol{a}_i$$

Multiplying by $\overline{\mathbf{X}}_{\widehat{b}}$ on both left sides and using similar arguments as in Li et al. (2015), $\overline{\mathbf{X}}_{\widehat{b}} \begin{bmatrix} \widehat{\boldsymbol{a}}_1 & \ldots & \widehat{\boldsymbol{a}}_N \end{bmatrix} = \overline{\mathbf{X}}_{\mathbf{b}} \begin{bmatrix} \boldsymbol{a}_1 & \ldots & \boldsymbol{a}_N \end{bmatrix}$. The right-hand side term reads:

$$\overline{\mathbf{X}}_{\mathbf{b}} \begin{bmatrix} \boldsymbol{a}_1 & \ldots & \boldsymbol{a}_N \end{bmatrix} = \mathbf{M} \begin{bmatrix} \mathbf{I}_N \otimes \mathbf{b}_1 \\ \vdots \\ \mathbf{I}_N \otimes \mathbf{b}_N \end{bmatrix} \begin{bmatrix} \mathbf{a}_1 & \ldots & \mathbf{a}_N \end{bmatrix}$$

and hence, for all $i, j \in \{1, \ldots, N\}^2$:

$$\mathbf{M}(\mathbf{I}_N \otimes \mathbf{b}_i)\mathbf{a}_j = \mathbf{M}(\mathbf{I}_N \otimes \widehat{\mathbf{b}}_i)\widehat{\mathbf{a}}_j$$

The matrix $\mathbf{M}$ is full column rank, it follows:

$$\begin{aligned} (\mathbf{I}_N \otimes \mathbf{b}_i)\mathbf{a}_j &= (\mathbf{I}_N \otimes \widehat{\mathbf{b}}_i)\widehat{\mathbf{a}}_j \\ \mathbf{b}_i \mathbf{a}_{j_k} &= \widehat{\mathbf{b}}_i \widehat{\mathbf{a}}_{j_k} \end{aligned}$$

Therefore, since $\mathbf{a}_{j_k} \in \mathbb{R}$ and $\mathbf{b} \in X_{\boldsymbol{b}}$, it follows: $\|\mathbf{b}_i\|_2 |\mathbf{a}_{j_k}| = \|\widehat{\mathbf{b}}_i\|_2 |\widehat{\mathbf{a}}_{j_k}|$ and then, $|\mathbf{a}_{j_k}| = |\widehat{\mathbf{a}}_{j_k}|$, for all k in the set $\{1, ..., N\}$. Finally, it comes $\|\mathbf{a}_j\|_2 = \|\widehat{\mathbf{a}}_j\|_2$. A similar reasoning starting from the relation between $\widehat{\mathbf{b}_{\mathbf{n}}}$ and $\boldsymbol{b}$ yields $\|\widehat{\mathbf{b}_{\mathbf{n}i}}\|_2 = \|\boldsymbol{b}_i\|_2$. ■

We can conclude:

$$\left\| \frac{d\mathcal{F}_2}{d\widehat{\boldsymbol{b}}_{\mathbf{n}}} \right\|_2 \leq 1 \tag{2.61}$$

Therefore, we use (2.59) and (2.61) to upper-bound the constant Q with:

$$Q \leq \frac{\|\overrightarrow{\boldsymbol{b}}^T \boldsymbol{b}\|_2}{\|\widehat{\boldsymbol{b}}^T \widehat{\boldsymbol{b}}\|_2} \|\boldsymbol{a}\|_2 \frac{\|\overrightarrow{\boldsymbol{a}}^T \boldsymbol{a}\|_2}{\|\widehat{\boldsymbol{a}}^T \widehat{\boldsymbol{a}}\|_2} \|\boldsymbol{b}\|_2 \leq \frac{\|\widehat{\boldsymbol{b}}^T \boldsymbol{b}\|_2}{\|\widehat{\boldsymbol{b}}^T \widehat{\boldsymbol{b}}\|_2} \frac{\|\widehat{\boldsymbol{a}}^T \boldsymbol{a}\|_2}{\|\widehat{\boldsymbol{a}}^T \widehat{\boldsymbol{a}}\|_2} \tag{2.62}$$

We conclude that $Q < 1$ using Lemma 2.9.

3
Identifying Kronecker-structured state-space models

We consider the identification of deterministic matrix state-space models (MSSM) of the following form:

$$\begin{aligned} \mathbf{X}(k+1) &= \mathbf{A}_1\mathbf{X}(k)\mathbf{A}_2^T + \mathbf{B}_1\mathbf{U}(k)\mathbf{B}_2^T \\ \mathbf{Y}(k) &= \mathbf{C}_1\mathbf{X}(k)\mathbf{C}_2^T + \mathbf{E}(k) \end{aligned}$$

for all time dependent quantities and matrices of appropriate dimensions. Due to the large size of these matrices, vectorization does not allow the use of standard subspace methods such as N4SID or MOESP. The resulting Kronecker structure that appears in the system matrices due to vectorization is exploited for developing a scalable subspace-like identification approach. This approach consists of first estimating the Markov parameters associated to the MSSM via the solution of a regularized bilinear least-squares problem that is solved in a globally convergent manner. Second, a low-rank minimization problem subject to bilinear constraints is tackled which optimized variables are subsequently used to form a third order tensor and eventually, to estimate the state-sequence and the lower-dimensional matrices $\mathbf{A}_1, \mathbf{A}_2, \mathbf{B}_1, \mathbf{B}_2, \mathbf{C}_1, \mathbf{C}_2$*. A numerical example on a large-scale adaptive optics system demonstrates the ability of the algorithm to handle the identification of Kronecker-structured stochastic state-space models in a scalable manner, which results in more compact models.*

This chapter is published in:
B. Sinquin and M. Verhaegen, "K4SID: Large-Scale Subspace Identification with Kronecker modeling," in *IEEE Transactions on Automatic Control*, vol. 64, no. 3, pp. 960-975, 2019.

3.1. Introduction

Let a regular grid be composed of $N \times N$ subsystems, each of which has m inputs and p outputs, and interacting with each other. A new modeling paradigm is introduced to model the state-space matrices which builds on the previous chapter where the class of low-Kronecker rank matrices has been described. The system matrices are assumed to have a Kronecker rank equal to one:

$$\left\{ \begin{array}{lcl} \mathbf{x}(k+1) & = & \big(\mathbf{A}_2 \otimes \mathbf{A}_1\big)\mathbf{x}(k) + \big(\mathbf{B}_2 \otimes \mathbf{B}_1\big)\mathbf{u}(k) \\ \mathbf{y}(k) & = & \big(\mathbf{C}_2 \otimes \mathbf{C}_1\big)\mathbf{x}(k) + \mathbf{e}(k) \end{array} \right. \tag{3.1}$$

When rewriting (3.1) into the matrix state-space model stated in the abstract, the separability assumption is shown more clearly: the products such as $\mathbf{A}_1\mathbf{X}(k)\mathbf{A}_2^T$ require separability of the column operations of the matrix $\mathbf{X}(k)$ from those of the row.

The main contributions of this chapter are the formulation of a new class of 2D spatial-temporal models within the state-space framework and a tailored subspace-like algorithm. For N_t the number of temporal samples used, we present an algorithm to estimate the system matrices with $\mathcal{O}(N^3 N_t)$ computational complexity rather than $\mathcal{O}(N^6)$. The QUARKS presented in the previous chapter now serve as a first step (out of three) in the identification of state-space models when the matrices are of Kronecker rank one. This algorithm is abbreviated with K4SID standing for Kronecker-Structured large-Scale SubSpace IDentification. Moreover, we highlight the performances in terms of data compression and prediction-error with an application to turbulence prediction for large-scale adaptive optics systems. K4SID is compared with SSARX, Hinnen (2007).

A class of multi-linear dynamical systems (MLDS) is introduced in Rogers et al. (2013) for modeling tensor-time series and an expectation-maximization algorithm is presented for estimating parameters. The well-known drawbacks of such methods are the a priori selection of the order and the high computational cost which reaches at least $\mathcal{O}(N^6)$ per iteration. They are often used in combination with subspace methods which provide them with initial estimates. The estimates of K4SID are refined with MLDS for small sizes of the sensor.

The chapter has the following outline. Section 3.2 formulates the identification problem and introduces theoretical results related to the Kronecker state-space model. Section 3.3 summarizes the identification of QUARKS models for estimating a high-order FIR filter in Kronecker form. The estimates feature two sequences of impulse response -with terms of size $pN \times mN$- that are related via a bilinear equation. Section 3.4 analyzes the question why realizing the state-space matrices from these estimates requires to first solve a bilinear low-rank optimization. A method is proposed in Section 3.5 to estimate the factor matrices $\mathbf{A}_1, \mathbf{A}_2, \mathbf{B}_1, \mathbf{B}_2, \mathbf{C}_1, \mathbf{C}_2$ using two consecutive SVD. A realistic numerical example for predicting large-scale wavefront aberrations in an adaptive optics setting is presented in Section 3.6.

Notations. Let $\mathbf{x} \in \mathbb{R}^s$. The Hankel matrix of dimension $\lfloor (s+1)/2 \rfloor \times \lfloor (s+1)/2 \rfloor$ built from the vector $\mathbf{x}$ is written with $\mathcal{H}(x_i)$. The notation extends to block-Hankel matrices.

Nomenclature. A matrix $\mathbf{X}$ written as $\mathbf{X} = \mathbf{X}_1 \otimes \mathbf{X}_2$ is said to have Kronecker rank one. The matrices $\mathbf{X}_1$ and $\mathbf{X}_2$ are called the factor matrices. The matrices $\mathbf{A}_i, \mathbf{B}_i, \mathbf{C}_i$ are called the factored state-space matrices. The terms $\mathbf{C}\mathbf{A}^i\mathbf{B} = \mathbf{C}_2\mathbf{A}_2^i\mathbf{B}_2 \otimes \mathbf{C}_1\mathbf{A}_1^i\mathbf{B}_1$ are the Markov parameters while $\mathbf{C}_j\mathbf{A}_j^i\mathbf{B}_j$ are called the factored Markov parameters.

3.2. Problem Formulation

We consider a 2D array with $N \times N$ nodes, and each of which is associated with m inputs and p outputs. The input data is collected at time instant k into the matrix $\mathbf{U}(k) \in \mathbb{R}^{mN \times N}$:

$$\mathbf{U}(k) = \begin{bmatrix} \mathbf{u}_{1,1}(k) & \dots & \mathbf{u}_{1,N}(k) \\ \vdots & & \vdots \\ \mathbf{u}_{N,1}(k) & \dots & \mathbf{u}_{N,N}(k) \end{bmatrix}$$

where, for $i, j = 1..N, \mathbf{u}_{i,j}(k) \in \mathbb{R}^m$. The output matrix $\mathbf{Y}(k)$ is defined similarly from local signals $\mathbf{y}_{i,j}(k) \in \mathbb{R}^p$. Denote the lifted quantities with $\mathbf{u}(k) = \text{vec}(\mathbf{U}(k))$. The temporal dynamics of the system are modeled with the state-space model (3.1) in which the state-space matrices have a Kronecker rank equal to one:

$$\mathbf{A} = \mathbf{A}_2 \otimes \mathbf{A}_1, \quad \mathbf{B} = \mathbf{B}_2 \otimes \mathbf{B}_1, \quad \mathbf{C} = \mathbf{C}_2 \otimes \mathbf{C}_1 \tag{3.2}$$

with,

$$\begin{array}{lll} \mathbf{A}_2 \in \mathbb{R}^{n_2 \times n_2}, & \mathbf{B}_2 \in \mathbb{R}^{n_2 \times N}, & \mathbf{C}_2 \in \mathbb{R}^{N \times n_2} \\ \mathbf{A}_1 \in \mathbb{R}^{n_1 \times n_1}, & \mathbf{B}_1 \in \mathbb{R}^{n_1 \times mN}, & \mathbf{C}_1 \in \mathbb{R}^{pN \times n_1} \end{array} \tag{3.3}$$

It is equivalently written in a matrix form which we introduce as

$$\left\{ \begin{array}{lcl} \mathbf{X}(k+1) & = & \mathbf{A}_1\mathbf{X}(k)\mathbf{A}_2^T + \mathbf{B}_1\mathbf{U}(k)\mathbf{B}_2^T \\ \mathbf{Y}(k) & = & \mathbf{C}_1\mathbf{X}(k)\mathbf{C}_2^T + \mathbf{E}(k) \end{array} \right. \tag{3.4}$$

Definition 3.1. *The set of generators $\mathcal{S}$ for the Kronecker MSSM* (3.4) *is defined from the factored state-space matrices as follows:*

$$\mathcal{S} = \{\mathbf{A}_1, \mathbf{A}_2, \mathbf{B}_1, \mathbf{B}_2, \mathbf{C}_1, \mathbf{C}_2\}$$

where the dimensions of the corresponding matrices are given in (3.3).

In Lemma 3.1, Lemma 3.2 and Corollary 3.1, we relate the stability, the observability and the minimal realization associated to the large-scale matrices $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ to the sets defined from $(\mathbf{A}_1, \mathbf{B}_1, \mathbf{C}_1)$ and $(\mathbf{A}_2, \mathbf{B}_2, \mathbf{C}_2)$. First, we establish a relationship between the spectral radius of $\mathbf{A}_2 \otimes \mathbf{A}_1$ and the one of the factor matrices $\mathbf{A}_1, \mathbf{A}_2$.

Lemma 3.1. *If the systems associated with $(\mathbf{A}_2, \mathbf{B}_2, \mathbf{C}_2)$ and $(\mathbf{A}_1, \mathbf{B}_1, \mathbf{C}_1)$ are both stable, then the system associated with $(\mathbf{A}_2 \otimes \mathbf{A}_1, \mathbf{B}_2 \otimes \mathbf{B}_1, \mathbf{C}_2 \otimes \mathbf{C}_1)$ is stable. The reverse is not true in general.*

Proof. Let $(i, j) \in \{1, ..., n_2\} \times \{1, ..., n_1\}$. Assume that $\mathbf{A}_2$ and $\mathbf{A}_1$ have eigenvalues, respectively $\mu_{2,i}$ and $\mu_{1,j}$, lying strictly within the unit circle. The eigenvalues of $\mathbf{A}_2 \otimes \mathbf{A}_1$ are $\mu_{2,i}\mu_{1,j}$. If $|\mu_{2,i}| < 1$ and $|\mu_{1,j}| < 1$, then $|\mu_{2,i}\mu_{1,j}| < 1$. However, if

$|\mu_{2,i}\mu_{1,j}| < 1$, it does not guarantee that both $|\mu_{2,i}| < 1$ and $|\mu_{1,j}| < 1$. ■
Let s_1 denote an integer such that $s_1 N \min(p, m) > \max(n_2, n_1)$ and s equal to $2s_1 - 1$. Such choices are explained in Section 3.4. Let the observability matrix built from two matrices $\mathbf{C}_i, \mathbf{A}_i$ be denoted with $\boldsymbol{\mathcal{O}}_{i,s}$ such that:

$$\boldsymbol{\mathcal{O}}_{i,s} = \begin{bmatrix} \mathbf{C}_i \\ \mathbf{C}_i\mathbf{A}_i \\ \vdots \\ \mathbf{C}_i\mathbf{A}_i^{s-1} \end{bmatrix}, \quad \boldsymbol{\mathcal{O}}_s = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \vdots \\ \mathbf{CA}^{s-1} \end{bmatrix} = \begin{bmatrix} \mathbf{C}_2 \otimes \mathbf{C}_1 \\ \mathbf{C}_2\mathbf{A}_2 \otimes \mathbf{C}_1\mathbf{A}_1 \\ \vdots \\ \mathbf{C}_2\mathbf{A}_2^{s-1} \otimes \mathbf{C}_1\mathbf{A}_1^{s-1} \end{bmatrix} \tag{3.5}$$

The extended controllability matrix built from $\mathbf{B}_i, \mathbf{A}_i$ is similarly denoted with $\boldsymbol{\mathcal{C}}_{i,s}$:

$$\begin{aligned} \boldsymbol{\mathcal{C}}_{i,s} &= \begin{bmatrix} \mathbf{B}_i & \mathbf{A}_i\mathbf{B}_i & \dots & \mathbf{A}_i^{s-1}\mathbf{B}_i \end{bmatrix} \\ \boldsymbol{\mathcal{C}}_s &= \begin{bmatrix} \mathbf{B} & \mathbf{AB} & \dots & \mathbf{A}^{s-1}\mathbf{B} \end{bmatrix} \end{aligned}$$

Lemma 3.2. *If $(\mathbf{A}_2 \otimes \mathbf{A}_1, \mathbf{C}_2 \otimes \mathbf{C}_1)$ is observable, then each of the pairs $(\mathbf{A}_2, \mathbf{C}_2)$ and $(\mathbf{A}_1, \mathbf{C}_1)$ is observable.*
The reverse is not true in general.

Proof. Let $n = n_1 n_2$. We start by partitioning the columns of $\boldsymbol{\mathcal{O}}_n$ block-wise:

$$\boldsymbol{\mathcal{O}}_n = \begin{bmatrix} \mathbf{L}_1 & \dots & \mathbf{L}_{n_2} \end{bmatrix} \tag{3.6}$$

where $\mathbf{L}_j \in \mathbb{R}^{npN^2 \times n_1}$ for $j = 1..n_2$. Each block-matrix $\mathbf{L}_j$ is such that:

$$\mathbf{L}_j = \begin{bmatrix} m_{0_{1,j}}\mathbf{W}_0 \\ \vdots \\ m_{0_{N,j}}\mathbf{W}_0 \\ \vdots \\ m_{n-1_{N,j}}\mathbf{W}_{n-1} \end{bmatrix} \tag{3.7}$$

where $\mathbf{M}_{n-1} = \mathbf{C}_2\mathbf{A}_2^{n-1}, \mathbf{W}_{n-1} = \mathbf{C}_1\mathbf{A}_1^{n-1}$. If $\boldsymbol{\mathcal{O}}_n$ is full column rank, so is $\mathbf{L}_j$. It yields the following equalities for the column ranks:

$$\operatorname{rank}(\mathbf{L}_j) = \operatorname{rank}(\boldsymbol{\mathcal{O}}_{1,n}) \tag{3.8}$$

Therefore, $\operatorname{rank}(\boldsymbol{\mathcal{O}}_{1,n}) = n_1$. A similar reasoning on the submatrix $\boldsymbol{\mathcal{O}}_n(1 : pN : pnN^2, 1 : n_1 : n)$ holds to prove that $\operatorname{rank}(\boldsymbol{\mathcal{O}}_{2,n}) = n_2$.
We provide with a counter-example for the reverse side:

$$\mathbf{C}_1 = \begin{bmatrix} 1 & 1 \end{bmatrix}, \quad \mathbf{A}_1 = \begin{bmatrix} 0.4 & 0 \\ 0 & 0.6 \end{bmatrix}, \quad \mathbf{C}_2 = \begin{bmatrix} 1 & 1 \end{bmatrix}, \quad \mathbf{A}_2 = \begin{bmatrix} 0.6 & 0 \\ 0 & 0.4 \end{bmatrix}$$

Both $\boldsymbol{\mathcal{O}}_{1,n}$ and $\boldsymbol{\mathcal{O}}_{2,n}$ are full column rank which is not the case for $\boldsymbol{\mathcal{O}}_n$. ■
Controllability is the dual notion from observability, and therefore, a similar statement can be made for the pairs $(\mathbf{A}_2, \mathbf{B}_2)$ and $(\mathbf{A}_1, \mathbf{B}_1)$ to be controllable.

Definition 3.2. *A minimal realization of* (3.4) *corresponds to a set* $\mathcal{S}$ *such that the extended observability and controllability matrices built respectively from the pairs* $(\mathbf{A}_2 \otimes \mathbf{A}_1, \mathbf{C}_2 \otimes \mathbf{C}_1)$ *and* $(\mathbf{A}_2 \otimes \mathbf{A}_1, \mathbf{B}_2 \otimes \mathbf{B}_1)$ *are of minimal rank* $n_2 n_1$.

Corollary 3.1. *If the set of generators* $\mathcal{S} = \{\mathbf{A}_1, \mathbf{A}_2, \mathbf{B}_1, \mathbf{B}_2, \mathbf{C}_1, \mathbf{C}_2\}$ *corresponds to a minimal realization of the MSSM* (3.4)*, then both sets* $\{\mathbf{A}_1, \mathbf{B}_1, \mathbf{C}_1\}$ *and* $\{\mathbf{A}_2, \mathbf{B}_2, \mathbf{C}_2\}$ *correspond to a minimal realization.*
The reverse is not true in general.

Proof. The proof follows from Lemma 3.2. ■

As shown in van Loan (2000) for standard matrix properties and in the above results, a particularity of the Kronecker product is that the properties relating the global matrices to the factors are often one-sided. Especially, we pointed out in Chapter 1 that modal approaches for system identification are able to guarantee global criteria contrary to the local modeling as e.g in the model of the interconnected string. When deriving algorithms using the factor matrices, we will not be able to guarantee observability nor controllability of the global system but only of the pairs formed from factor matrices.

We now investigate the state-space (3.4) from the input-output relationship, which matrix form reads:

$$\mathbf{Y}(k) = \mathbf{C}_1 \mathbf{A}_1^{k-1} \mathbf{X}(1) {\mathbf{A}_2^{k-1}}^T \mathbf{C}_2^T + \sum_{i=1}^{k-1} \mathbf{C}_1 \mathbf{A}_1^{k-i-1} \mathbf{B}_1 \mathbf{U}(i) \mathbf{B}_2^T {\mathbf{A}_2^{k-i-1}}^T \mathbf{C}_2^T + \mathbf{E}(k) \tag{3.9}$$

Definition 3.3. *Let* N_t *be the number of temporal samples. For* $i \in \{1, 2\}$*, denote:*

$$\mathcal{S}_i = \{\mathbf{A}_1^{(i)}, \mathbf{A}_2^{(i)}, \mathbf{B}_1^{(i)}, \mathbf{B}_2^{(i)}, \mathbf{C}_1^{(i)}, \mathbf{C}_2^{(i)}\}$$

The two sets of generators $\mathcal{S}_1$ *and* $\mathcal{S}_2$ *are said to be equivalent if the input-output behaviour of the associated state-space model* (3.4) *is identical for all* $k = 1..N_t$.

It is well-known that the state-space matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$ in (3.1) modeling the input-output relationship are not unique because of the existence of a non-singular similarity transformation $\mathbf{T} \in \mathbb{R}^{n_2 n_1 \times n_2 n_1}$. Reshuffling (3.1) yields (3.4) if and only if the state-space matrices are all of Kronecker rank one. It is not the case when allowing similarity transformations not written as $\mathbf{T} = \mathbf{T}_2 \otimes \mathbf{T}_1$. We characterize the similarity transformation in the case of Kronecker state-space models and relate equivalent sets of generators in the next lemma.

Lemma 3.3. *The sets of generators* $\mathcal{S}_1$ *and* $\mathcal{S}_2$ *for the Kronecker MSSM equivalently model* (3.4) *if and only if there exist* $\mathbf{T}_1 \in \mathbb{R}^{n_1 \times n_1}, \mathbf{T}_2 \in \mathbb{R}^{n_2 \times n_2}$ *non-singular,* $\mathbf{P}_1 \in \mathbb{R}^{mN \times mN}, \mathbf{P}_2 \in \mathbb{R}^{N \times N}$ *and non-zero scalars* η, c_t *that satisfy:*

$$\forall k \in \{1, ..., N_t\}, \quad \mathbf{P}_1 \mathbf{U}(k) \mathbf{P}_2^T = \mathbf{U}(k) \tag{3.10}$$

$$\begin{cases} \mathbf{A}_2^{(1)} &= \eta \mathbf{T}_2^{-1} \mathbf{A}_2^{(2)} \mathbf{T}_2 \\ \mathbf{B}_2^{(1)} &= \mathbf{T}_2^{-1} \mathbf{B}_2^{(2)} \mathbf{P}_2 \\ \mathbf{C}_2^{(1)} &= c_t \mathbf{C}_2^{(2)} \mathbf{T}_2 \end{cases} \tag{3.11}$$

$$\begin{cases} \mathbf{A}_1^{(1)} &= \frac{1}{\eta}\mathbf{T}_1^{-1}\mathbf{A}_1^{(2)}\mathbf{T}_1 \\ \mathbf{B}_1^{(1)} &= \mathbf{T}_1^{-1}\mathbf{B}_1^{(2)}\mathbf{P}_1 \\ \mathbf{C}_1^{(1)} &= \frac{1}{c_t}\mathbf{C}_1^{(2)}\mathbf{T}_1 \end{cases} \tag{3.12}$$

Proof. (Sufficiency) With such parametrization, the state-space model built from $\mathcal{S}_1$ is equivalent to the model built from $\mathcal{S}_2$:

$$\begin{cases} \tilde{\mathbf{X}}(k+1) &= \mathbf{T}_1^{-1}\big(\frac{1}{\eta}\mathbf{A}_1^{(2)}\mathbf{T}_1\tilde{\mathbf{X}}(k)\mathbf{T}_2^T\eta\mathbf{A}_2^{(2)^T} + \mathbf{B}_1^{(2)}\mathbf{P}_1\mathbf{U}(k)\mathbf{P}_2^T\mathbf{B}_2^{(2)^T}\big)\mathbf{T}_2^{-T} \\ \mathbf{Y}(k) &= \frac{1}{c_t}\mathbf{C}_1^{(2)}\mathbf{T}_1\tilde{\mathbf{X}}(k)\mathbf{T}_2^T c_t\mathbf{C}_2^{(2)^T}) \end{cases} \tag{3.13}$$

where $\tilde{\mathbf{X}}(k) = \mathbf{T}_1^{-1}\mathbf{X}(k)\mathbf{T}_2^{-T}$. This model yields the same input-output behaviour provided that:

$$\mathbf{P}_1\mathbf{U}(k)\mathbf{P}_2^T = \mathbf{U}(k) \tag{3.14}$$

Vectorizing (3.13) yields:

$$\begin{cases} \mathrm{vec}(\mathbf{X}(k+1)) &= \big(\mathbf{A}_2^{(2)} \otimes \mathbf{A}_1^{(2)}\big)\mathrm{vec}(\mathbf{X}(k)) + \big(\mathbf{B}_2^{(2)} \otimes \mathbf{B}_1^{(2)}\big)\mathbf{u}(k) \\ \mathbf{y}(k) &= \big(\mathbf{C}_2^{(2)} \otimes \mathbf{C}_1^{(2)}\big)\mathrm{vec}(\mathbf{X}(k)) \end{cases}$$

(Necessity) We now prove that the similarity transformation $\mathbf{T}$ is necessarily of Kronecker rank one such that both global matrices $\mathbf{A}^{(1)}$ and $\mathbf{A}_T := \mathbf{T}^{-1}\mathbf{A}^{(1)}\mathbf{T}$ are of Kronecker rank one. Let $\mathbf{T} = \sum_{i=1}^{n}\mathbf{T}_{i,\ell} \otimes \mathbf{T}_{i,r}$ such that

$$\mathrm{rank}(\begin{bmatrix}\mathrm{vec}(\mathbf{T}_{1,\ell}) & \dots & \mathrm{vec}(\mathbf{T}_{n,\ell})\end{bmatrix}\begin{bmatrix}\mathrm{vec}(\mathbf{T}_{1,r})^T \\ \dots \\ \mathrm{vec}(\mathbf{T}_{n,r})^T\end{bmatrix}) = n \tag{3.15}$$

for $n > 1$. For simplicity, we assume $n = 2$ although the reasoning still holds for larger values. The matrix $\mathbf{A}_T$ is parametrized with a Kronecker rank-one structure and it will be shown that it implies $\mathbf{T}_{i,\ell} = \mathbf{T}_{j,\ell}$ for all i, j, hence a contradiction with (3.15) and therefore $\mathbf{T}$ is of Kronecker rank one. We start by writing $\mathbf{T}\mathbf{A}_T = \mathbf{A}^{(1)}\mathbf{T}$ with a sum of Kroneckers and reshuffle it into $\mathbf{U}_1\mathbf{V}_1^T = \mathbf{U}_2\mathbf{V}_2^T$, where, for $i \in \{1, 2\}$:

$$\begin{cases} \mathbf{U}_1(:,i) &= \mathrm{vec}(\mathbf{A}_2^{(1)}\mathbf{T}_{i,\ell}), \quad \mathbf{V}_1(:,i) = \mathrm{vec}(\mathbf{A}_1^{(1)}\mathbf{T}_{i,r}) \\ \mathbf{U}_2(:,i) &= \mathrm{vec}(\mathbf{T}_{i,\ell}\mathbf{A}_{T,2}), \quad \mathbf{V}_2(:,i) = \mathrm{vec}(\mathbf{T}_{i,r}\mathbf{A}_{T,1}) \end{cases} \tag{3.16}$$

There exist a non-singular matrix $\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22}\end{bmatrix}$ such that $\mathbf{U}_1 = \mathbf{U}_2\mathbf{P}$. Rewriting the above equation yields:

$$\begin{cases} -p_{11}\mathbf{T}_{1,\ell}\mathbf{A}_{T,2} + \mathbf{A}_2^{(1)}\mathbf{T}_{1,\ell} &= p_{21}\mathbf{T}_{2,\ell}\mathbf{A}_{T,2} \\ -p_{22}\mathbf{T}_{2,\ell}\mathbf{A}_{T,2} + \mathbf{A}_2^{(1)}\mathbf{T}_{2,\ell} &= p_{12}\mathbf{T}_{1,\ell}\mathbf{A}_{T,2} \end{cases} \tag{3.17}$$

This can be transformed into:

$$\begin{cases} \mathbf{A}_2^{(1)}(p_{12}\mathbf{T}_{1,\ell} - p_{11}\mathbf{T}_{2,\ell}) &= (p_{12}p_{21} - p_{11}p_{22})\mathbf{T}_{2,\ell}\mathbf{A}_{T,2} \\ \mathbf{A}_2^{(1)}(p_{21}\mathbf{T}_{2,\ell} - p_{22}\mathbf{T}_{1,\ell}) &= (p_{12}p_{21} - p_{11}p_{22})\mathbf{T}_{1,\ell}\mathbf{A}_{T,2} \end{cases} \tag{3.18}$$

The matrices $\mathbf{A}_2^{(1)}$ and $\mathbf{A}_{T,2}$ are similarly equivalent, which means that:

$$\begin{cases} p_{12}\mathbf{T}_{1,\ell} &= (p_{12}p_{21} - p_{11}p_{22} + p_{11})\mathbf{T}_{2,\ell} \\ p_{21}\mathbf{T}_{2,\ell} &= (p_{12}p_{21} - p_{11}p_{22} + p_{22})\mathbf{T}_{1,\ell} \end{cases} \tag{3.19}$$

As $\text{vec}(\mathbf{T}_{2,\ell})$ and $\text{vec}(\mathbf{T}_{1,\ell})$ are linearly independent, the first equation implies $p_{12} = 0$ and $p_{11}(p_{22} - 1) = 0$ whereas the second equation gives $p_{21} = 0$ and $p_{22}(p_{11} - 1) = 0$. If either p_{11} or p_{22} is zero, then $\mathbf{P}$ is singular which is a contradiction. If both are equal to one, $\mathbf{P} = \mathbf{I}$, thus $\mathbf{T}_{2,\ell}$ is equal to $\mathbf{T}_{1,\ell}$ using (3.18). We conclude that $\mathbf{T}$ is of Kronecker rank one. ■

The matrices $\mathbf{P}_2, \mathbf{P}_1$ in Lemma 3.3 can be further characterized. For all temporal samples until N_t, we form the matrix $\tilde{\mathbf{U}} \in \mathbb{R}^{mN^2 \times N_t}$ from concatenated input data with: $\tilde{\mathbf{U}} = \begin{bmatrix} \text{vec}(\mathbf{U}(1)) & \dots & \text{vec}(\mathbf{U}(N_t)) \end{bmatrix}$. Let us assume $N_t \geq mN^2$ and that the matrix $\tilde{\mathbf{U}}$ is full row rank. Then, the equation (3.14) is equivalently written with:

$$(\mathbf{P}_2 \otimes \mathbf{P}_1 - \mathbf{I}_{mN^2})\text{vec}(\mathbf{U}(k)) = \mathbf{0} \tag{3.20}$$

Concatenating data for all k in $\{1, ..., N_t\}$, it follows:

$$(\mathbf{P}_2 \otimes \mathbf{P}_1 - \mathbf{I}_{mN^2})\tilde{\mathbf{U}} = \mathbf{0} \tag{3.21}$$

Under the assumption that the matrix $\tilde{\mathbf{U}}$ is full row rank, $\mathbf{P}_2 \otimes \mathbf{P}_1 = \mathbf{I}_{mN^2}$. It implies that both $\mathbf{P}_2$ and $\mathbf{P}_1$ are diagonal, and, for all $i = 1..mN, j = 1..N$, $p_{1,(i,i)}p_{2,(j,j)} = 1$. Hence, $\mathbf{P}_2 = b_t\mathbf{I}_N$ and $\mathbf{P}_1 = \frac{1}{b_t}\mathbf{I}_{mN}$ for some non-zero scalar b_t.

In other words, when the similarity transformation $\mathbf{T}$ is unstructured, the Kronecker rank one structure in the vectorized state-space model is lost for the matrix $\mathbf{T}^{-1}\mathbf{A}\mathbf{T}$. Therefore, the vector state-space model with matrices $(\mathbf{T}^{-1}(\mathbf{A}_2 \otimes \mathbf{A}_1)\mathbf{T}, \mathbf{T}^{-1}(\mathbf{B}_2 \otimes \mathbf{B}_1), (\mathbf{C}_2 \otimes \mathbf{C}_1)\mathbf{T})$ cannot be rewritten in general with a matrix state-space model as in (3.4). It is of particular interest as the identification algorithm we propose relies on the matrix state-space model (3.4) and the global matrices are never formed. The global similarity transformation is not involved.

In order to derive a scalable identification algorithm, the following assumptions on the data and system matrices in (3.4) are made:

- **A1:** The pair $(\mathbf{A}_2 \otimes \mathbf{A}_1, \mathbf{C}_2 \otimes \mathbf{C}_1)$ is observable.
- **A2:** The pair $(\mathbf{A}_2 \otimes \mathbf{A}_1, \mathbf{B}_2 \otimes \mathbf{B}_1)$ is controllable.
- **A3:** The eigenvalues of both $\mathbf{A}_2$ and $\mathbf{A}_1$ are strictly within the unit circle.
- **A4:** The matrix $\tilde{\mathbf{U}}$ is full row rank.
- **A5:** The measurement noise is zero-mean white noise with unknown covariance matrix.
- **A6:** The measurement noise is uncorrelated with all past inputs:

$$\text{for all } k \leq j,\ \mathbb{E}\left[\mathbf{u}(k)\mathbf{e}(j)^T\right] = \mathbf{0}$$

The assumptions **A1** to **A4** and **A6** are related to the global system properties and are commonly used in subspace identification, Verhaegen and Verdult (2007). The assumption **A6** is also made in Yu et al. (2018a), Yu and Verhaegen (2018a) in order to focus on the essential building of the subspace identification method(s) like VARX modeling and the state sequence approximation. The generality of the method is illustrated in Section 3.6 in which a model in innovation form is identified.

Problem Formulation: *Assuming **A1** to **A6**, and given the input-output data* $\mathbf{U}(k), \mathbf{Y}(k)$ *from the state-space model in* (3.4) *for* $k = \{1, ..., N_t\}$*, estimate, up to the similarities transformation* $\mathbf{T}_1, \mathbf{T}_2$ *and the ambiguity scaling factors* η, c_t, b_t *defined in Lemma 3.3, the matrices* $\mathbf{A}_1, \mathbf{A}_2, \mathbf{B}_1, \mathbf{B}_2, \mathbf{C}_1, \mathbf{C}_2$ *that correspond to a minimal realization. The challenge lies on deriving an algorithm with* $\mathcal{O}(N^3 N_t)$ *computational complexity.*

Such requirements on the computational cost exclude an identification of the unstructured state-space model with standard subspace methods such as MOESP or SSARX. These methods fail for three main reasons. First, they rely on a QR decomposition of the concatenated block-Hankel matrix built from the input-output sequence, whose size is $(p+m)sN^2 \times (N_t - s + 1)$, for some scalar s. A square lower-triangular Gram-Schmidt matrix is only obtained when $N_t \geq psN^2$ which requires storing huge data samples. Second, with a global system order of $n_2 n_1$, computing the QR decomposition and the SVD of $N^2 \times N^2$ matrices is very costly, $\mathcal{O}(N^6)$ flops. If a prior knowledge of the system order is available, then a rank-$n_2 n_1$ SVD can be computed at a cost of $\mathcal{O}(n_2 n_1 N^4)$. More efficient methods as in Halko et al. (2011) for computing SVD do not break the curse of dimensionality that appears with multi-dimensional systems and still require $\mathcal{O}(log(n_2 n_1)N^4)$ flops. Last, forming the global matrices is a drawback for storage and e.g subsequent control design for real-time applications. For example, computing a matrix-vector multiplication with the dense unstructured matrix requires $\mathcal{O}(N^4)$ instead of $\mathcal{O}(N^3)$ in the matrix form.

The algorithm PBSID provides an alternative route that estimates first a high-order VARX and then computes the SVD of a large-matrix, Chiuso (2007). The computational cost associated with the latter operation along with the estimation of unstructured and dense estimates of the state-space matrices reaches $\mathcal{O}(N^6)$ and is reduced in this work by working rather with the factored Markov parameters $\mathbf{C}_2\mathbf{A}_2^i\mathbf{B}_2$ and $\mathbf{C}_1\mathbf{A}_1^i\mathbf{B}_1$.

In the three following sections, we describe the subspace-like method which is decomposed in three major steps. We first identify the factored Markov parameters using a globally convergent algorithm. Such parameters are however estimated up to an unknown scaling factor. Second, a low-rank optimization problem with bilinear constraints is formulated to pave the way for the third step in Section 3.5 where we identify the factored state-space matrices by estimating the state-sequence.

3.3. High-order FIR estimation

3.3.1. A QUARKS model

Based on Assumption **A3**, the output $\mathbf{y}(k)$ can be approximated with a high-order Finite Impulse Response model for all $k > s$:

$$\begin{aligned}\mathbf{y}(k) &\approx \sum_{i=1}^{s} \mathbf{C}\mathbf{A}^{i-1}\mathbf{B}\mathbf{u}(k-i) + \mathbf{e}(k) \\ &\approx \sum_{i=1}^{s} \big(\mathbf{C}_2\mathbf{A}_2^{i-1}\mathbf{B}_2 \otimes \mathbf{C}_1\mathbf{A}_1^{i-1}\mathbf{B}_1\big)\mathbf{u}(k-i) + \mathbf{e}(k) \end{aligned} \tag{3.22}$$

Denote $\mathbf{M}_{i,\ell} := \mathbf{C}_2\mathbf{A}_2^{i-1}\mathbf{B}_2 \in \mathbb{R}^{N\times N}$ and $\mathbf{M}_{i,r} := \mathbf{C}_1\mathbf{A}_1^{i-1}\mathbf{B}_1 \in \mathbb{R}^{pN\times mN}$. The matrix $\mathbf{M}_\ell = \begin{bmatrix}\mathbf{M}_{1,\ell} & \dots & \mathbf{M}_{s,\ell}\end{bmatrix}$ is denoted as the left-factor impulse response. Similarly, the matrix $\mathbf{M}_r$ is built from the factor matrices $\mathbf{M}_{i,r}$ and called the right-factor impulse response.

By appropriately selecting the parameters as standardly done in the subspace identification literature, Verhaegen and Verdult (2007), the approximation error can be made arbitrarily small, Knudsen (2001).

A computationally efficient and globally convergent algorithm has been derived in Chapter 2 to estimate structured large-scale VARX models when the coefficient-matrices have large dimensions but low-Kronecker rank. In the FIR approximation (3.22), each Markov parameter $\mathbf{M}_i$ has Kronecker rank equal to one, and hence the equation (3.22) can be recast into a minimization on the factor matrices $\mathbf{M}_{i,\ell}, \mathbf{M}_{i,r}$ only. The stability of the impulse responses built from factored matrices is imposed by using kernel regularization methods. The kernel matrix $\mathbf{P}_t \in \mathbb{R}^{s\times s}$ is here introduced along with the decomposition of its inverse with a square-root matrix $\mathbf{K}_t$. Adding the following cost as regularization to a cost function induces stable VARX models:

$$r_t(\mathbf{M}_{i,\ell}, \mathbf{M}_{i,r}) = \|\mathbf{Q}_t \begin{bmatrix}\mathrm{vec}(\mathbf{M}_{1,\ell})\mathrm{vec}(\mathbf{M}_{1,r})^T \\ \vdots \\ \mathrm{vec}(\mathbf{M}_{p,\ell})\mathrm{vec}(\mathbf{M}_{p,r})^T\end{bmatrix} \|_F^2 \tag{3.23}$$

where $\mathbf{Q}_t = \mathbf{W}_t \otimes \mathbf{I}_{N^2}$. The factor matrices $\mathbf{M}_{i,\ell}, \mathbf{M}_{i,r}$ are estimated using Alternating Least Squares on the following least-squares bilinear minimization problem:

$$\min_{\mathbf{M}_{i,r},\mathbf{M}_{i,\ell}} \sum_{k=s+1}^{N_t} \|\mathbf{Y}(k) - \sum_{i=1}^{s} \mathbf{M}_{i,r}\mathbf{U}(k-i)\mathbf{M}_{i,\ell}^T\|_F^2 + \lambda_{ALS} r_t(\mathbf{M}_{i,\ell}, \mathbf{M}_{i,r}) \tag{3.24}$$

where λ_{ALS} is a regularization parameter. An Alternating Least Squares algorithm is proposed and described in Algorithm 3.1. The notation $\mathcal{L}_{\mathbf{M}_r}(\mathbf{M}_\ell)$ is introduced and refers to the the cost function in (3.24) when the optimization variables are only $\mathbf{M}_{i,\ell}$ for all i while $\mathbf{M}_{i,r}$ is fixed. The solution to (3.24) is not unique as summarized in the following Lemma.

Lemma 3.4. *Let N_t tend towards infinity. Let $i \in \{1, ..., s\}$ and $t_i \in \mathbb{R}\backslash\{0\}$. Denote a solution to* (3.24) *with the parameters $\mathbf{M}_{i,\ell}, \mathbf{M}_{i,r}$.*

The set of all solutions $\widehat{\mathbf{M}}_{i,\ell}, \widehat{\mathbf{M}}_{i,r}$ *that yield the same optimal value of the cost function* (3.24) *is such that:*

$$vec(\widehat{\mathbf{M}}_{i,\ell}) = vec(\mathbf{M}_{i,\ell})t_i, \qquad vec(\widehat{\mathbf{M}}_{i,r}) = \frac{1}{t_i} vec(\mathbf{M}_{i,r})$$

Denote the estimates from (3.24) with $\widehat{\mathbf{M}}_{i,\ell}, \widehat{\mathbf{M}}_{i,r}$. They are related to the non-scaled parameters $\mathbf{M}_{i,\ell}, \mathbf{M}_{i,r}$ with:

$$\widehat{\mathbf{M}}_{i,\ell} \approx t_i \mathbf{M}_{i,\ell}, \qquad \widehat{\mathbf{M}}_{i,r} \approx v_i \mathbf{M}_{i,r} \tag{3.25}$$

where $t_i v_i \approx 1$. The non-zero ambiguity constants t_i are however unknowns and *different* for each i.

Algorithm 3.1: Summary of QUARKS estimation

Input : $\{\mathbf{u}(k)\}_{1:N_t}, \{\mathbf{y}(k)\}_{1:N_t}, s, \lambda_{ALS}, \kappa_{max}, \epsilon_{min}$
Output : $\widehat{\mathbf{M}}_r, \widehat{\mathbf{M}}_\ell$

$\kappa \leftarrow 0, \epsilon = \infty$
foreach $i \leq s$ **do**
  $\mathbf{M}_{i,\ell}^{(\kappa)} \leftarrow$ `randn(N,N)`
end
while $\kappa \leq \kappa_{max}$ and $\epsilon > \epsilon_{min}$ **do**
  $\mathbf{M}_r^{(\kappa+1)} \leftarrow \text{argmin } \mathcal{L}_{\mathbf{M}_\ell^{(\kappa)}}(\mathbf{M}_r)$.
  $\mathbf{M}_\ell^{(\kappa+1)} \leftarrow \text{argmin } \mathcal{L}_{\mathbf{M}_r^{(\kappa+1)}}(\mathbf{M}_\ell)$.
  Evaluate the residual $c^{(\kappa)}$
  $\epsilon \leftarrow |c^{(\kappa)} - c^{(\kappa-1)}|$
  $\kappa \leftarrow \kappa + 1$
end
$\widehat{\mathbf{M}}_r \leftarrow \mathbf{M}_r^{(\kappa-1)}$
$\widehat{\mathbf{M}}_\ell \leftarrow \mathbf{M}_\ell^{(\kappa-1)}$

3.3.2. Computational complexity

The computational complexity for the QUARKS was studied in Chapter 2. We assume that the number of iterations κ_{max} are independent of N, which is however not the case for the number of temporal samples N_t. The algorithm scales with $\mathcal{O}(N^3 N_t)$.

3.4. Estimation of the impulse responses up to a scaling factor

In this section, we assume that the matrices $\mathbf{M}_{i,\ell}$ and $\mathbf{M}_{i,r}$ are estimated up to a different non-zero scaling factor as highlighted in Lemma 3.4. This relationship

between the matrices estimated with QUARKS in (3.24) and their true variants hold in an asymptotic consistent manner. We now study how to estimate the factored Markov parameters $w\mathbf{C}_2(\eta\mathbf{A}_2)^{i-1}\mathbf{B}_2$ (for w, η non-zero scalars). Although the forthcoming analysis is performed on the left factor matrices $\mathbf{M}_{i,\ell}$, it is equally valid for the right factor matrices $\mathbf{M}_{i,r}$.

3.4.1. A low-rank block-Hankel matrix.

From the matrices $\{\mathbf{M}_{i,\ell}\}_{i=1..s}$, the realization theory consists in forming a block-Hankel low-rank matrix that is equal to $\boldsymbol{\mathcal{O}}_{2,s_1}\boldsymbol{\mathcal{C}}_{2,s_1}$ for some integer $s_1 \geq n_2$. A SVD is then computed to estimate the range space of the observability matrix $\boldsymbol{\mathcal{O}}_{2,s_1}$. In the analysis from this section, the approximations in (3.25) are considered as equalities. Because the unknowns t_i are dependent on the index i, the block-Hankel matrix built from the estimated matrices $\{\widehat{\mathbf{M}}_{i,\ell}\}_{i=1..s}$ is in general not low-rank. Each of the factored Markov parameters need to be multiplied with a scalar on which we formulate conditions such that the resulting block-Hankel matrix has rank equal to n_2. We describe this statement in Theorem 3.1 for which Lemma 3.5 is needed.

Lemma 3.5. *(Partial Realization Problem) Gragg and Lindquist (1983). Let $\delta \in \mathbb{N}$. Let $s = 2s_1 - 1$ with s_1 an integer strictly larger than δ. For $i = 1..s$, let $x_i \in \mathbb{R}$ such that rank$\big(\mathcal{H}(x_i)\big) = \delta$.*
Then, there exists a realization $(\mathbf{a}, \mathbf{b}, \mathbf{c})$ of minimal degree δ with $\mathbf{a} \in \mathbb{R}^{\delta\times\delta}, \mathbf{b} \in \mathbb{R}^{\delta\times 1}, \mathbf{c} \in \mathbb{R}^{1\times\delta}$ such that for $i = 1..s, x_i = \mathbf{c}\mathbf{a}^{i-1}\mathbf{b}$. This decomposition is unique up to a similarity transformation.

The triplet $(\mathbf{a}, \mathbf{b}, \mathbf{c})$ defines a partial realization on the finite sequence $\{x_i\}_{i=1..s}$.

Theorem 3.1. *Let $s = 2s_1 - 1$ with s_1 an integer such that $s_1 N \min(p, m) \geq \max(n_2, n_1)$. For $i \in \{1, .., s\}$, let (α_i, t_i) be non-zero scalars and let the matrices $\widehat{\mathbf{M}}_{i,\ell}$ satisfy:*

$$\widehat{\mathbf{M}}_{i,\ell} = t_i \mathbf{M}_{i,\ell} \tag{3.26}$$

with rank$(\mathcal{H}(\mathbf{M}_{i,\ell})) = n_2$.

- *If rank$\big(\mathcal{H}(\alpha_i\widehat{\mathbf{M}}_{i,\ell})\big) = n_2$, then rank$\big(\mathcal{H}(\alpha_i t_i)\big) = 1$.*
- *If $\alpha_i t_i = \eta^{i-1}$ for a non-zero scalar η, then:*

$$\textit{rank}\big(\mathcal{H}(\alpha_i\widehat{\mathbf{M}}_{i,\ell})\big) = n_2 \tag{3.27}$$

Proof. We derive the proof using the contraposition. In the sequel we denote $x_i = \alpha_i t_i$ and $\mathbf{X}_i = \mathbf{M}_{i,\ell}$. Let $\delta \in \mathbb{N}$ such that $1 < \delta \leq s_1$ and suppose that:

$$\text{rank}\big(\mathcal{H}(x_i)\big) = \delta \tag{3.28}$$

From Lemma 3.5, there exists a realization $(\mathbf{a}, \mathbf{b}, \mathbf{c})$ of minimal degree δ with $\mathbf{a} \in \mathbb{R}^{\delta\times\delta}, \mathbf{b} \in \mathbb{R}^{\delta\times 1}, \mathbf{c} \in \mathbb{R}^{1\times\delta}$ such that for $i = 1..s, x_i = \mathbf{c}\mathbf{a}^{i-1}\mathbf{b}$. If every eigenvalue of $\mathbf{a}$ is 0, then $\mathbf{a}$ is nilpotent (via Cayley-Hamilton) which is forbidden by the

assumption $x_i \neq 0$ for all i. Therefore, the matrix $\mathbf{a}$ has at least one non-zero eigenvalue.

We divide the proof in two cases. If $\mathbf{a}$ is diagonalizable, there exists an invertible matrix $\mathbf{P}$ such that $\mathbf{a} = \mathbf{PDP}^{-1}$ where $\mathbf{D}$ is a diagonal matrix containing the eigenvalues. All eigenvalues λ_i are distinct. Let $k \in \mathbb{N}$. We have:

$$\begin{aligned} \mathbf{ca}^k\mathbf{bCA}^k\mathbf{B} &= \mathbf{C}(\mathbf{ca}^k\mathbf{b})\mathbf{A}^k\mathbf{B} \\ &= \mathbf{CcPD}^k\mathbf{P}^{-1}\mathbf{bA}^k\mathbf{B} \end{aligned} \tag{3.29}$$

Denote $\tilde{\mathbf{c}} = \mathbf{cP}$, $\tilde{\mathbf{b}} = \mathbf{P}^{-1}\mathbf{b}$ and $r_i = \tilde{\mathbf{c}}_i\tilde{\mathbf{b}}_i \neq 0$. We write:

$$\begin{aligned} \mathbf{ca}^k b\mathbf{CA}^k\mathbf{B} &= \mathbf{C}\tilde{\mathbf{c}}\mathbf{D}^k\tilde{\mathbf{b}}\mathbf{A}^k\mathbf{B} \\ &= \mathbf{C}\sum_{i=1}^{\delta}\tilde{\mathbf{c}}_i\lambda_i^k\tilde{\mathbf{b}}_i\mathbf{A}^k\mathbf{B} \\ &= \sum_{i=1}^{\delta} r_i\mathbf{C}(\lambda_i\mathbf{A})^k\mathbf{B} \end{aligned} \tag{3.30}$$

Then, without loss of generality, consider $\delta = 2$. These Markov parameters are associated with the state-space matrices:

$$\tilde{\mathbf{A}} = \begin{bmatrix} \lambda_1\mathbf{A} & \mathbf{0} \\ \mathbf{0} & \lambda_2\mathbf{A} \end{bmatrix}, \quad \tilde{\mathbf{B}} = \begin{bmatrix} \mathbf{B} \\ \mathbf{B} \end{bmatrix}, \quad \tilde{\mathbf{C}} = \begin{bmatrix} r_1\mathbf{C} & r_2\mathbf{C} \end{bmatrix} \tag{3.31}$$

Let $\mathbf{W}_i$ be an eigenvector of $\mathbf{A}$. Then both $\begin{bmatrix} \mathbf{W}_i \\ \mathbf{0} \end{bmatrix}$ and $\begin{bmatrix} \mathbf{0} \\ \mathbf{W}_i \end{bmatrix}$ are eigenvectors of $\tilde{\mathbf{A}}$. The condition $\tilde{\mathbf{C}}\begin{bmatrix} \mathbf{W}_i \\ \mathbf{0} \end{bmatrix} = \mathbf{0}$ or $\tilde{\mathbf{C}}\begin{bmatrix} \mathbf{0} \\ \mathbf{W}_i \end{bmatrix} = \mathbf{0}$ is equivalently written with:

$$r_i\mathbf{CW}_i = \mathbf{0} \tag{3.32}$$

for $i \in \{1, 2\}$. Using the Popov-Belevitch-Hautus (PBH) test and with the assumption that the pair $(\mathbf{A}, \mathbf{C})$ is observable, it implies that $\mathbf{W}_i = \mathbf{0}$. Therefore the pair $(\tilde{\mathbf{A}}, \tilde{\mathbf{C}})$ is observable following the PBH test. It follows that the rank of $\mathcal{H}(x_i\mathbf{X}_i)$ is strictly larger than n_2 and we have a contradiction.

Suppose now that the matrix $\mathbf{a}$ is not diagonalizable. There exists an invertible matrix $\mathbf{P}$ such that $\mathbf{a} = \mathbf{PJP}^{-1}$, where $\mathbf{J}$ is the Jordan matrix. The latter matrix is block-diagonal: each of the so-called Jordan blocks has a size equal to the algebraic multiplicity of the associated eigenvalue. Let q denote the number of blocks (also equal to the number of different eigenvalues) and h_i the multiplicity of the i-th eigenvalue.

Without loss of generality, we assume that λ_i has multiplicity 2. The Jordan blocks $\mathbf{J}_i$ have then the following form:

$$\mathbf{J}_i = \begin{bmatrix} \lambda_i & 1 \\ 0 & \lambda_i \end{bmatrix} \tag{3.33}$$

It can be proven (using e.g induction) that $\mathbf{J}_i^k$ is expressed as:

$$\mathbf{J}_i^k = \begin{bmatrix} \lambda_i^k & k\lambda_i^{k-1} \\ 0 & \lambda_i^k \end{bmatrix} \tag{3.34}$$

The expression in (3.30) reads:

$$\begin{aligned} \mathbf{c}\mathbf{a}^k\mathbf{b}\mathbf{C}\mathbf{A}^k\mathbf{B} &= \mathbf{C}\sum_{i=1}^{q} \tilde{\mathbf{c}}_i \begin{bmatrix} \lambda_i^k & k\lambda_i^{k-1} \\ 0 & \lambda_i^k \end{bmatrix} \tilde{\mathbf{b}}_i \mathbf{A}^k\mathbf{B} \\ &= \sum_{i=1}^{q} \tilde{\mathbf{C}}_i \tilde{\mathbf{A}}_i^k \tilde{\mathbf{B}}_i \end{aligned} \tag{3.35}$$

where,

$$\tilde{\mathbf{A}}_i = \begin{bmatrix} \lambda_i\mathbf{A} & \mathbf{A} \\ \mathbf{0} & \lambda_i\mathbf{A} \end{bmatrix}, \quad \tilde{\mathbf{B}}_i = \begin{bmatrix} \tilde{b}_{i,1}\mathbf{B} \\ \tilde{b}_{i,2}\mathbf{B} \end{bmatrix}, \quad \tilde{\mathbf{C}}_i = \begin{bmatrix} \tilde{c}_{i,1}\mathbf{C} & \tilde{c}_{i,2}\mathbf{C} \end{bmatrix} \tag{3.36}$$

The matrix $\mathbf{a}$ has at least one non-zero eigenvalue from the assumption that all x_i are different from 0. The following therefore assumes $\lambda_i \neq 0$.

The observability matrix associated to the triplet $(\tilde{\mathbf{A}}_i, \tilde{\mathbf{B}}_i, \tilde{\mathbf{C}}_i)$ is written as:

$$\mathcal{V}_{n,i} = \begin{bmatrix} \tilde{c}_{i,1}\mathbf{C} & \tilde{c}_{i,2}\mathbf{C} \\ \tilde{c}_{i,1}\mathbf{C}(\lambda_i\mathbf{A}) & \tilde{c}_{i,1}\mathbf{C}\mathbf{A} + \tilde{c}_{i,2}\mathbf{C}(\lambda_i\mathbf{A}) \\ \vdots & \vdots \end{bmatrix} \tag{3.37}$$

whose rank is equal to the rank of $\frac{1}{\tilde{c}_{i,1}}\boldsymbol{\mathcal{V}}_n$ for $\lambda_i \neq 0, \tilde{c}_{i,1} \neq 0$:

$$\frac{1}{\tilde{c}_{i,1}}\boldsymbol{\mathcal{V}}_{n,i} = \begin{bmatrix} \mathbf{C} & \mathbf{0} \\ \mathbf{C}(\lambda_i\mathbf{A}) & \mathbf{C}(\lambda_i\mathbf{A}) \\ \mathbf{C}(\lambda_i A)^2 & 2\mathbf{C}(\lambda_i\mathbf{A})^2 \\ \vdots & \vdots \end{bmatrix} \begin{bmatrix} \mathbf{I} & \frac{\tilde{c}_{i,2}}{\tilde{c}_{i,1}}\mathbf{I} \\ \mathbf{0} & \frac{1}{\lambda_i}\mathbf{I} \end{bmatrix} \tag{3.38}$$

From Sylvester's inequality, the rank of $\frac{1}{\tilde{c}_{i,1}}\boldsymbol{\mathcal{V}}_{n,i}$ is equal to the rank of the following matrix:

$$\begin{bmatrix} \mathbf{C} & \mathbf{0} \\ \mathbf{C}(\lambda_i\mathbf{A}) & \mathbf{C}(\lambda_i\mathbf{A}) \\ \mathbf{C}(\lambda_i\mathbf{A})^2 & 2\mathbf{C}(\lambda_i\mathbf{A})^2 \\ \vdots & \vdots \end{bmatrix} \tag{3.39}$$

If the pair $(\mathbf{C}, \lambda_i\mathbf{A})$ is observable, then the matrices:

$$\begin{bmatrix} \mathbf{C} \\ \mathbf{C}(\lambda_i\mathbf{A}) \\ \mathbf{C}(\lambda_i\mathbf{A})^2 \\ \vdots \end{bmatrix}, \quad \begin{bmatrix} \mathbf{I} & & & \\ & 2\mathbf{I} & & \\ & & \ddots & \\ & & & (s-1)\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{C} \\ \mathbf{C}(\lambda_i\mathbf{A}) \\ \mathbf{C}(\lambda_i\mathbf{A})^2 \\ \vdots \end{bmatrix} \tag{3.40}$$

are both full column rank. Owing to the zero-block in the upper right part of the matrix (3.39), the rank of $\boldsymbol{\mathcal{V}}_{n,i}$ is strictly larger than n_2 and we again have a contradiction.
The proof for the second bullet is as follows. If $\alpha_i t_i = \eta^{i-1}$ for all i, then:

$$\begin{aligned} \mathcal{H}(\alpha_i \widehat{\mathbf{M}}_{i,\ell}) &= \mathcal{H}(\mathbf{C}_2(\eta\mathbf{A}_1)^{i-1}\mathbf{B}_2) \\ &= \boldsymbol{\mathcal{O}}_{2,s_1}\boldsymbol{\mathcal{C}}_{2,s_1} \end{aligned} \tag{3.41}$$

where $\boldsymbol{\mathcal{O}}_{2,s_1}$ and $\boldsymbol{\mathcal{C}}_{2,s_1}$ are respectively the observability and controllability matrices associated with the pairs $(\mathbf{C}_2, \eta\mathbf{A}_2)$ and $(\eta\mathbf{A}_2, \mathbf{B}_2)$. From Sylvester's inequality, the rank of $\boldsymbol{\mathcal{O}}_{2,s_1}\boldsymbol{\mathcal{C}}_{2,s_1}$ is equal to n_2 which proves that the rank of $\mathcal{H}(\alpha_i\widehat{\mathbf{M}}_{i,\ell})$ is n_2 under the conditions specified in the theorem. ■

Corollary 3.2. *With the notations introduced in Theorem 3.1. If rank*$\big(\mathcal{H}(\alpha_i\widehat{\mathbf{M}}_{i,\ell})\big) = n_2$*, then there exist non-zero scalars* $(a_\alpha, b_\alpha, c_\alpha)$ *such that* $\alpha_i = \frac{c_\alpha a_\alpha^{i-1} b_\alpha}{t_i}$.

Proof. It follows from Theorem 3.1 by using $x_i = \alpha_i t_i$. ■

Corollary 3.3. *With the notations introduced in Theorem 3.1. If rank*$\big(\mathcal{H}(\beta_i\widehat{\mathbf{M}}_{i,r})\big) = n_1$*, then there exist* $(a_\beta, b_\beta, c_\beta)$ *non-zero scalars such that* $\beta_i = c_\beta a_\beta^{i-1} b_\beta t_i$.

Proof. It follows from Theorem 3.1 adapted to $\mathcal{H}(\beta_i\widehat{\mathbf{M}}_{i,r})$ and by using $x_i = \frac{\beta_i}{t_i}$. ■

To summarize, the matrix $\mathcal{H}(\widehat{\mathbf{M}}_{i,\ell})$ is in general not low-rank as indicated in Theorem 3.1 because the scaling factor t_i is different for each factor matrix. The properties of the block-Hankel $\mathcal{H}(\alpha_i\widehat{\mathbf{M}}_{i,\ell})$ have then been studied. If the rank of the latter matrix is minimal, then $\text{rank}\big(\mathcal{H}(\alpha_i t_i)\big) = 1$. There are however an infinite number of sequences $\boldsymbol{\alpha}$ for which such a condition is valid. In the next theorem, we analyze a rank minimization problem featuring both low-rank block-Hankel matrices $\mathcal{H}(\alpha_i\widehat{\mathbf{M}}_{i,\ell})$ and $\mathcal{H}(\beta_i\widehat{\mathbf{M}}_{i,r})$ and study the uniqueness when the scalings α_i, β_i are related with a bilinear constraint.

Theorem 3.2. *The solution to the multi-criteria feasibility problem:*

$$\begin{aligned} \textit{find} \quad & (\boldsymbol{\alpha}, \boldsymbol{\beta}) \\ \textit{s.t} \quad & \{\textit{rank}\big(\mathcal{H}(\alpha_i\widehat{\mathbf{M}}_{i,\ell})\big) = n_2, \textit{rank}\big(\mathcal{H}(\beta_i\widehat{\mathbf{M}}_{i,r})\big) = n_1\} \\ & \forall i \in \{1, ..., s\}, \quad \alpha_i\beta_i = 1 \end{aligned} \tag{3.42}$$

is not unique and feasible values for (3.42) *are obtained for all* α, β *as described in Corollary 3.2 and 3.3 with the additional conditions that* $a_\alpha a_\beta = 1$ *and* $c_\alpha c_\beta b_\alpha b_\beta = 1$.

Proof. Using Corollary 3.2 and 3.3, the above rank conditions are satisfied for all $i = 1..s$:

$$\alpha_i = \frac{c_\alpha a_\alpha^{i-1} b_\alpha}{t_i}, \qquad \beta_i = c_\beta a_\beta^{i-1} b_\beta t_i \tag{3.43}$$

Replacing these expressions inside the bilinear constraint (3.42) yields:

$$\alpha_i\beta_i = c_\alpha c_\beta (a_\alpha a_\beta)^{i-1} b_\alpha b_\beta = 1 \tag{3.44}$$

which implies $a_\alpha a_\beta = 1$ and $c_\alpha c_\beta b_\alpha b_\beta = 1$. ■

Remark 3.1. *With the bilinear constraint $\alpha_i\beta_i = 1$, both α_i and β_i cannot be 0. Moreover, the scalars t_i, v_i are related with $t_i v_i = 1$ using Lemma 3.4. Therefore, both sequences $\alpha_i t_i$ and $\frac{\beta_i}{t_i}$ are non-zero which fulfills the condition expressed in Theorem 3.2.*

3.4.2. A bilinear constrained low-rank optimization

We now propose a method to estimate a set of vectors $\boldsymbol{\alpha}, \boldsymbol{\beta}$ using the constraints that have been derived in the previous paragraph and the estimates $\widehat{\mathbf{M}}_{i,\ell}, \widehat{\mathbf{M}}_{i,r}$ that have been obtained with the QUARKS identification. From (3.42), and without knowing a priori the system orders (n_2, n_1), we formulate a bilinear rank optimization problem:

$$\begin{aligned} \min_{\boldsymbol{\alpha},\boldsymbol{\beta}} \quad & \operatorname{rank}\big(\mathcal{H}(\alpha_i\widehat{\mathbf{M}}_{i,\ell})\big) + \operatorname{rank}\big(\mathcal{H}(\beta_i\widehat{\mathbf{M}}_{i,r})\big) \\ \text{s.t} \quad & \forall i \in \{1,...,s\}, \quad \alpha_i\beta_i = 1 \end{aligned} \tag{3.45}$$

where λ is a regularization parameter that trades between the low-rank priors and the stability constraint. The minimization problem (3.45) is bilinear and features the rank operator which is non-convex. When convexifying the rank operator with the nuclear norm, it belongs to the class of multi-convex optimization problems as described in Nocedal and Wright (2006). The works in Xu and Yin (2013) and Doelman and Verhaegen (2016) both propose an iterative algorithm. Use is made of a Block-Coordinate Update algorithm (BCU) with slack variables, Xu and Yin (2013). The slack variables are used to relax the bilinear constraints and are denoted with q_i for $i \in \{1,...,s\}$. The optimization (3.45) is transformed into:

$$\begin{aligned} \min_{\boldsymbol{\alpha},\boldsymbol{\beta},\mathbf{q}} \quad & \|\mathcal{H}(\alpha_i\widehat{\mathbf{M}}_{i,\ell})\|_\star + \|\mathcal{H}(\beta_i\widehat{\mathbf{M}}_{i,r})\|_\star + \mu\sum_{i=1}^{s} q_i^2 \\ \text{s.t} \quad & \forall i \in \{1,...,s\}, \quad \alpha_i\beta_i - 1 = q_i \end{aligned} \tag{3.46}$$

where μ is a regularization parameter. The higher μ is, the more weight is laid on setting $\mathbf{q}$ to 0. The optimization problem is solved iteratively with non-zero initial guesses and with a low value for μ, hence without necessarily requiring the bilinear equality to hold. At each iteration, the optimization successively solves over $(\boldsymbol{\alpha}, \mathbf{q})$ and then $(\boldsymbol{\beta}, \mathbf{q})$. The number of variables is moreover only $2s$. Algorithm 3.2 details the steps. The notation $\mathcal{B}_{\widehat{\mathbf{M}}_r,\boldsymbol{\alpha},\mu}(\boldsymbol{\beta}, \mathbf{q})$ is introduced for the cost function in (3.46) when the optimization variables are $\boldsymbol{\beta}, \mathbf{q}$ only while $\boldsymbol{\alpha}$ is fixed, and with the regularization parameter μ. The regularization parameter μ shall be gradually increased throughout the iterations to ensure that the bilinear constraint (3.46) is met, Xu and Yin (2013). We highlight the prominent role of the initial value $\mu^{(0)}$ for μ. If it is set too large when optimizing over $(\boldsymbol{\alpha}, \mathbf{q})$, respectively $(\boldsymbol{\beta}, \mathbf{q})$, the variable α_i is fixed to $1/\beta_i$ by the constraint. If it is set too low, α_i goes to 0 and q_i to -1. In that respect, $\mu^{(0)}$ plays the role of a regularization parameter whose optimal value is determined by grid search.

Standard ADMM techniques apply here and a detailed analysis of the optimization updates are inspired from Verhaegen and Hansson (2016) using the linear

operator framework to enforce the block-Hankel structure. The nuclear norm minimization is performed via singular value soft-thresholding. The details are however not reproduced here to focus on the subspace algorithm and the implementation is found online.

Algorithm 3.2: Summary of BCU

Input : $\widehat{\mathbf{M}}_\ell, \widehat{\mathbf{M}}_r, \mu^{(0)}$
Output : $\widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{\beta}}$

/* Default values */
$d = 5, \tau = 5, \kappa_{max} = 40, \epsilon_{min} = 10^{-3}$
$\kappa \leftarrow 0$
foreach $i \leq s$ **do**
  $\alpha_i^{(\kappa)} \leftarrow 1$
end
while $\kappa \leq \kappa_{max}$ and $\epsilon > \epsilon_{min}$ **do**
  $\boldsymbol{\beta}^{(\kappa+1)} \leftarrow \text{argmin } \mathcal{B}_{\widehat{\mathbf{M}}_r, \boldsymbol{\alpha}^{(\kappa)}, \mu^{(\kappa)}}(\boldsymbol{\beta}, \mathbf{q})$.
  $\boldsymbol{\alpha}^{(\kappa+1)} \leftarrow \text{argmin } \mathcal{B}_{\widehat{\mathbf{M}}_\ell, \boldsymbol{\beta}^{(\kappa+1)}, \mu^{(\kappa)}}(\boldsymbol{\alpha}, \mathbf{q})$.
  if $mod(\kappa, d) = 0$ **then**
    $\mu^{(\kappa+1)} \leftarrow \tau \mu^{(\kappa)}$
  end
  $\epsilon \leftarrow \sum_{i=1}^{s} (\alpha_i \beta_i - 1)^2$
  $\kappa \leftarrow \kappa + 1$
end
$\widehat{\boldsymbol{\alpha}} = \boldsymbol{\alpha}^{(\kappa-1)}, \widehat{\boldsymbol{\beta}} = \boldsymbol{\alpha}^{(\kappa-1)}$

3.4.3. Computational complexity

The complexity of Algorithm 3.2 is essentially to solve each ADMM problem. We assume that the number of iterations is independent from N. Prior to performing the ADMM updates, a Gramian matrix is computed based on the sequences $\{\widehat{\mathbf{M}}_{i,r}\}, \{\widehat{\mathbf{M}}_{i,\ell}\}$ with $\mathcal{O}(N^2)$ flops. Two operations that appear in each of the above ADMM algorithm are detailed below. The number of unknowns is only $2s$ and hence, the cost of the primal variable update in each of the ADMM algorithm is not dominated by the matrix inversion but rather forming the matrices prior to solving the least squares, which scales with $\mathcal{O}(N^2)$ only. However, at each iteration of the ADMM algorithm, a SVD of the block-Hankel matrix $\mathcal{H}(\alpha_i^{(\kappa)} \widehat{\mathbf{M}}_{i,\ell})$ (respectively, $\mathcal{H}(\beta_i^{(\kappa)} \widehat{\mathbf{M}}_{i,r})$) is computed with a singular-value soft-thresholding, which is the bottleneck in Algorithm 3.2 as this operation scales with $\mathcal{O}(N^3)$.

3.5. Estimating the state-space matrices

3.5.1. A data-equation in matrix form

In this section, we estimate the state-sequence and then perform a bilinear least-squares optimization on the MSSM model (3.4). Similarly to standard subspace identification methods, a data equation is first written in the form:

$$\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{V}} + \boldsymbol{\mathcal{T}}_{\boldsymbol{u}} + \boldsymbol{\mathcal{E}} \tag{3.47}$$

where the block-Hankel matrix $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{pNs \times NM}$ is as follows:

$$\boldsymbol{\mathcal{Y}} = \begin{bmatrix} \mathbf{Y}(1) & \mathbf{Y}(2) & \dots & \mathbf{Y}(M) \\ \mathbf{Y}(2) & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ \mathbf{Y}(s) & \mathbf{Y}(s+1) & \dots & \mathbf{Y}(N_t) \end{bmatrix} \tag{3.48}$$

with $M = N_t - s + 1$. The block-Hankel matrix $\boldsymbol{\mathcal{E}}$ is similarly built from the noise matrices $\mathbf{E}(k)$. The matrix $\boldsymbol{\mathcal{V}}$ is defined with:

$$\boldsymbol{\mathcal{V}} = \begin{bmatrix} \mathbf{C}_r\mathbf{X}(1)\mathbf{C}_\ell^T & \dots & \mathbf{C}_r\mathbf{X}(M)\mathbf{C}_\ell^T \\ \mathbf{C}_r\mathbf{A}_r\mathbf{X}(1)(\mathbf{C}_\ell\mathbf{A}_\ell)^T & \dots & \vdots \\ \vdots & \dots & \vdots \\ \mathbf{C}_r\mathbf{A}_r^{s-1}\mathbf{X}(1)(\mathbf{C}_\ell\mathbf{A}_\ell^{s-1})^T & \dots & \mathbf{C}_r\mathbf{A}_r^{s-1}\mathbf{X}(M)(\mathbf{C}_\ell\mathbf{A}_\ell^{s-1})^T \end{bmatrix} \tag{3.49}$$

and $\boldsymbol{\mathcal{T}}_{\boldsymbol{u}}$ with:

$$\begin{bmatrix} \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{M}_{1,r}\mathbf{U}(1)\mathbf{M}_{1,\ell}^T & \dots & \mathbf{M}_{1,r}\mathbf{U}(M)\mathbf{M}_{1,\ell}^T \\ \sum_{i=0}^{1}\mathbf{M}_{i+1,r}\mathbf{U}(2-i)\mathbf{M}_{i+1,\ell}^T & & \vdots \\ \vdots & \ddots & \vdots \\ \sum_{i=0}^{s-2}\mathbf{M}_{i+1,r}\mathbf{U}(s-1-i)\mathbf{M}_{i+1,\ell}^T & \dots & \sum_{i=0}^{s-2}\mathbf{M}_{i+1,r}\mathbf{U}(N_t-i)\mathbf{M}_{i+1,\ell}^T \end{bmatrix} \tag{3.50}$$

The data equation in matrix form (3.47) features matrices of sizes of the order N rather than N^2 but with no key structural properties like low-rank as is exploited in standard subspace identification, Verhaegen and Verdult (2007). For example, the matrix $\boldsymbol{\mathcal{V}}$ is in general not low-rank, hence the estimation of the state sequence is not straightforward.

The terms in $\boldsymbol{\mathcal{V}}$ are now embedded into a structured third order tensor denoted with $\boldsymbol{\mathcal{A}}$. Let $\varphi \in \mathbb{N}$ such that $\varphi pN > n_1$ and $\varphi N > n_2$. For all $k = 1..M$, a slice $\boldsymbol{\mathcal{A}}(:,:,k) \in \mathbb{R}^{p\varphi N \times \varphi N}$ is described with:

$$\begin{bmatrix} \mathbf{C}_1\mathbf{X}(k)\mathbf{C}_2^T & \dots & \mathbf{C}_1\mathbf{X}(k)(\mathbf{C}_2\mathbf{A}_2^{\varphi-1})^T \\ \vdots & & \vdots \\ \mathbf{C}_1\mathbf{A}_1^{\varphi-1}\mathbf{X}(k)\mathbf{C}_2^T & \dots & \mathbf{C}_1\mathbf{A}_1^{\varphi-1}\mathbf{X}(k)(\mathbf{C}_2\mathbf{A}_2^{\varphi-1})^T \end{bmatrix} \tag{3.51}$$

We first justify the use of the tensor $\boldsymbol{\mathcal{A}}$ before focusing on estimating its entries. The rank properties are related to its matricizations.

Definition 3.4. *Let $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{p\varphi N \times \varphi N \times M}$ be a third order tensor. The unfolding $\boldsymbol{\mathcal{A}}_{(1)}$ is defined with:*

$$\boldsymbol{\mathcal{A}}_{(1)} = \begin{bmatrix} \boldsymbol{\mathcal{A}}(:,:,1) & \dots & \boldsymbol{\mathcal{A}}(:,:,M) \end{bmatrix} \in \mathbb{R}^{p\varphi N \times \varphi N M} \tag{3.52}$$

Consequently, using (3.51) along with Definition 3.4:

$$\boldsymbol{\mathcal{A}}_{(1)} = \begin{bmatrix} \mathbf{C}_1 \\ \mathbf{C}_1\mathbf{A}_1 \\ \vdots \\ \mathbf{C}_1\mathbf{A}_1^{\varphi-1} \end{bmatrix} \begin{bmatrix} \mathbf{X}(1)\mathbf{C}_2^T & \dots & \mathbf{X}(M)(\mathbf{C}_2\mathbf{A}_2^{\varphi-1})^T \end{bmatrix} \tag{3.53}$$

The rank of the tensor unfolding $\boldsymbol{\mathcal{A}}_{(1)}$ is equal to n_1:

$$\operatorname{rank}(\boldsymbol{\mathcal{A}}_{(1)}) = n_1 < p\varphi N \tag{3.54}$$

Computing an SVD of $\boldsymbol{\mathcal{A}}_{(1)}$ yields:

$$\begin{aligned} \boldsymbol{\mathcal{A}}_{(1)} &= \mathbf{U}_1\mathbf{V}_1 \\ \mathbf{U}_1 &= \boldsymbol{\mathcal{O}}_{\varphi,1}\mathbf{T}_1 \\ \mathbf{V}_1 &= \mathbf{T}_1^{-1}\begin{bmatrix} \mathbf{X}(1)\mathbf{C}_2^T & \dots & \mathbf{X}(M)(\mathbf{C}_2\mathbf{A}_2^{\varphi-1})^T \end{bmatrix} \end{aligned} \tag{3.55}$$

for a non-singular $\mathbf{T}_1$. From (3.55) and more precisely from $\mathbf{U}_1$, the matrices $\mathbf{A}_1$ and $\mathbf{C}_1$ are estimated. By reshaping the matrix $\mathbf{V}_1$, we can write:

$$\begin{aligned} \mathbf{H} &= \begin{bmatrix} \mathbf{T}_1^{-1}\mathbf{X}(1)\mathbf{C}_2^T & \dots & \mathbf{T}_1^{-1}\mathbf{X}(1)(\mathbf{C}_2\mathbf{A}_2^{\varphi-1})^T \\ \vdots & & \vdots \\ \mathbf{T}_1^{-1}\mathbf{X}(M)\mathbf{C}_2^T & \dots & \mathbf{T}_1^{-1}\mathbf{X}(M)(\mathbf{C}_2\mathbf{A}_2^{\varphi-1})^T \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{T}_1^{-1}\mathbf{X}(1) \\ \vdots \\ \mathbf{T}_1^{-1}\mathbf{X}(M) \end{bmatrix} \begin{bmatrix} \mathbf{C}_2^T & \dots & (\mathbf{C}_2\mathbf{A}_2^{\varphi-1})^T \end{bmatrix} \end{aligned} \tag{3.56}$$

The following rank equality holds:

$$\operatorname{rank}(\mathbf{H}) = n_2 < \varphi N \tag{3.57}$$

An SVD on the low-rank matrix $\mathbf{H}$ gives:

$$\begin{aligned} \mathbf{H} &= \mathbf{U}_2\mathbf{V}_2 \\ \mathbf{U}_2 &= \begin{bmatrix} \mathbf{T}_1^{-1}\mathbf{X}(1)\mathbf{T}_2 \\ \vdots \\ \mathbf{T}_1^{-1}\mathbf{X}(M)\mathbf{T}_2 \end{bmatrix} \\ \mathbf{V}_2 &= \mathbf{T}_2^{-1}\begin{bmatrix} \mathbf{C}_2^T & \dots & (\mathbf{C}_2\mathbf{A}_2^{\varphi-1})^T \end{bmatrix} \end{aligned} \tag{3.58}$$

Hence, $\mathbf{U}_2$ provides with an estimate for the state-sequence up to two similarity transformations as presented initially in the proof of Lemma 3.3. The matrix $\mathbf{V}_2$ is

equal to the extended observability matrix $\boldsymbol{\mathcal{O}}_{\varphi,2}$ up to a similarity transformation $\mathbf{T}_2$, and the matrices $\mathbf{A}_2$ and $\mathbf{C}_2$ can then be estimated. From both rank equalities (3.54) and (3.57), we conclude that two consecutive SVDs on respectively $\boldsymbol{\mathcal{A}}_{(1)}$ and $\mathbf{H}$ enable to estimate the state-sequence.

Remark 3.2. *The first SVD* (3.55) *is performed on the unfolded tensor* $\boldsymbol{\mathcal{A}}_{(1)}$ *while the second one* (3.58) *deals with a reduced size matrix* $\mathbf{H}$ *obtained from the right singular vectors in* (3.55)*. A CPD of* $\boldsymbol{\mathcal{A}}$ *does not allow to estimate the state-sequence: it would provide sets of matrices of size* $p\varphi N \times r$*,* $\varphi N \times r$*, and one of size* $M \times r$ *(instead of* $Mr \times r$*). It thus does not capture the full state.*

Remark 3.3. *The feasibility problem* (3.42)*, and similarly the rank minimization* (3.45)*, can lead to unstable* factored *models. For* α_i *as described in Corollary 3.3, the sequence* $\alpha_i\widehat{\mathbf{M}}_{i,\ell}$ *becomes:*

$$\begin{aligned}\alpha_i\widehat{\mathbf{M}}_{i,\ell} &= c_\alpha \eta^{i-1} b_\alpha \mathbf{C}_2\mathbf{A}_2^{i-1}\mathbf{B}_2 \\ &= c_\alpha \mathbf{C}_2(\eta\mathbf{A}_2)^{i-1} b_\alpha \mathbf{B}_2\end{aligned}$$

The stability of $\eta\mathbf{A}_1$ *is not guaranteed depending on the value for* η*. Although this does not affect the estimation of* $\boldsymbol{\alpha}, \boldsymbol{\beta}$*, we wish to recover two stable impulses built from the factor matrices. It is suggested to compute an eigenvalue decomposition of both matrices* $\widehat{\mathbf{A}}_1$ *and* $\widehat{\mathbf{A}}_2$ *and divide the entries of the matrix with the largest spectral radius with e.g* $1.05 \cdot max(\lambda_{max}(\{\mathbf{A}_i\}_{i=1..2}))$ *while counter-scaling the other matrix.*

3.5.2. Estimating the tensor

In the noise-free case $\boldsymbol{\mathcal{E}} = 0$, the terms $\mathbf{C}_1\mathbf{A}_1^i\mathbf{X}(k)(\mathbf{C}_2\mathbf{A}_2^j)^T$ for $i = j$ and located on the main block-diagonal of $\boldsymbol{\mathcal{A}}(:,:,k)$ are available from:

$$\boldsymbol{\mathcal{V}} \approx \boldsymbol{\mathcal{Y}} - \widehat{\boldsymbol{\mathcal{T}}_{\boldsymbol{u}}} \tag{3.59}$$

where $\widehat{\boldsymbol{\mathcal{T}}_{\boldsymbol{u}}}$ is obtained from $\boldsymbol{\mathcal{T}}_{\boldsymbol{u}}$ by replacing all $\mathbf{M}_{i,\ell}, \mathbf{M}_{i,r}$ with $\widehat{\mathbf{M}}_{i,\ell}, \widehat{\mathbf{M}}_{i,r}$.

Furthermore, the block-entries away from the main block-diagonal of $\boldsymbol{\mathcal{A}}(:,:,k)$ feature cross-terms such as $\mathbf{C}_1\mathbf{A}_1^i\mathbf{X}(k)(\mathbf{C}_2\mathbf{A}_2^j)^T$ for $i \neq j$. To cope with both the measurement noise and the estimation of cross-terms, we introduce so-called *virtual* outputs, denoted with $\mathbf{Y}^\sharp$, as follows:

$$\begin{cases} \mathbf{X}(k+1) &= \mathbf{A}_1\mathbf{X}(k)\mathbf{A}_2^T + \mathbf{B}_1\mathbf{U}(k)\mathbf{B}_2^T \\ \mathbf{Y}^\sharp_{g,h}(k) &= \mathbf{C}^\sharp_{1,g}\mathbf{X}(k)\mathbf{C}^{\sharp\,T}_{2,h} \end{cases} \tag{3.60}$$

where $g, h \in \mathbb{N}, \mathbf{C}^\sharp_{1,g} = \mathbf{C}_1\mathbf{A}_1^g$ and $\mathbf{C}^\sharp_{2,h} = \mathbf{C}_2\mathbf{A}_2^h$. In other words, $\mathbf{Y}^\sharp_{g,h}(k)$ is a virtual output associated with the set of Kronecker-generators $\mathcal{S}_{g,h} = \{\mathbf{A}_1, \mathbf{A}_2, \mathbf{B}_1, \mathbf{B}_2, \mathbf{C}_1\mathbf{A}_1^g, \mathbf{C}_2\mathbf{A}_2^h\}$. The virtual outputs $\{\mathbf{Y}^\sharp_{g,h}(k)\}_{k=1..N_t}$ are not known when both g, h are not zero and are approximated with a high-order FIR filter. Let $z \in \mathbb{N}$. For all $k \geq z$:

$$\mathbf{Y}^\sharp_{g,h}(k) \approx \sum_{i=0}^{z-1} \mathbf{C}_1\mathbf{A}_1^{i+g}\mathbf{B}_1\mathbf{U}(k-i-1)(\mathbf{C}_2\mathbf{A}_2^{i+h}\mathbf{B}_2)^T \tag{3.61}$$

$$\mathbf{Y}^{\sharp}_{g,h}(k) \approx \sum_{i=0}^{z-1} \mathbf{M}_{i+g+1,r}\mathbf{U}(k-i-1)\mathbf{M}^{T}_{i+h+1,\ell} \tag{3.62}$$

Using the estimates $\widehat{\mathbf{M}}_{i,\ell}, \widehat{\mathbf{M}}_{i,r}$ for the factored left and right coefficient-matrices in the QUARKS model in Algorithm 3.1 along with the estimates $\widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{\beta}}$ in Algorithm 3.2, the equation (3.62) reads:

$$\mathbf{Y}^{\sharp}_{g,h}(k) \approx \sum_{i=0}^{z-1} \widehat{\beta}_{i+g+1}\widehat{\mathbf{M}}_{i+g+1,r}\mathbf{U}(k-i-1)\widehat{\alpha}_{i+h+1}\widehat{\mathbf{M}}^{T}_{i+h+1,\ell} \tag{3.63}$$

$\mathbf{Y}^{\sharp}_{0,0}(k)$ is an FIR approximation of the noise free model (3.4), hence the coefficients $\widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{\beta}}$ are not needed for computing the virtual output $\mathbf{Y}^{\sharp}_{0,0}(k)$.

We now investigate the requirements on the indices g, h to fill the tensor $\boldsymbol{\mathcal{A}}$ according to (3.51). Both sequences $\widehat{\boldsymbol{\alpha}}$ and $\widehat{\boldsymbol{\beta}}$ have been estimated with s entries, therefore the equation (3.63) implies the following ranges for choosing the triplet (z, g, h):

$$z + g \leq s, \quad z + h \leq s \tag{3.64}$$

For φ strictly smaller than s and larger than $\frac{n}{p}$ so that the rank inequalities (3.54) and (3.57) hold, the combinations of the pair (g, h) for filling $\mathcal{A}$ are obtained with, $g = 0, h \in \{1, \ldots, \varphi - 1\}$, and $g = 0, h = 0$, and $g \in \{1, \ldots, \varphi - 1\}, h = 0$. The maximum value of g is obtained for $g = \varphi - 1$, which implies $z + \varphi - 1 = s$. A total of $2\varphi - 1$ virtual outputs are available within the temporal range $\{z, .., N_t\}$. For each of the associated subsystems in (3.60), a data equation in matrix form similar to (3.59) is written:

$$\boldsymbol{\mathcal{Y}}^{\sharp}_{g,h} = \boldsymbol{\mathcal{V}}_{g,h} + \boldsymbol{\mathcal{T}}_{g,h} \tag{3.65}$$

where, for $M_z = N_t - \varphi + 1$:

$$\boldsymbol{\mathcal{Y}}^{\sharp}_{g,h} = \begin{bmatrix} \mathbf{Y}^{\sharp}_{g,h}(z) & \mathbf{Y}^{\sharp}_{g,h}(z+1) & \ldots & \mathbf{Y}^{\sharp}_{g,h}(M_z) \\ \mathbf{Y}^{\sharp}_{g,h}(z+1) & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ \mathbf{Y}^{\sharp}_{g,h}(z+\varphi-1) & \ldots & \ldots & \mathbf{Y}^{\sharp}_{g,h}(N_t) \end{bmatrix}$$

$$\boldsymbol{\mathcal{V}}_{g,h} = \begin{bmatrix} \mathbf{C}^{\sharp}_{1,g}\mathbf{X}(z){\mathbf{C}^{\sharp}_{2,h}}^{T} & \ldots & \mathbf{C}^{\sharp}_{1,g}\mathbf{X}(M_z){\mathbf{C}^{\sharp}_{2,h}}^{T} \\ \vdots & \ddots & \vdots \\ \mathbf{C}^{\sharp}_{1,g}\mathbf{A}_1^{\varphi-1}\mathbf{X}(z)(\mathbf{C}^{\sharp}_{2,h}\mathbf{A}_2^{\varphi-1})^{T} & \ldots & \mathbf{C}^{\sharp}_{1,g}\mathbf{A}_1^{\varphi-1}\mathbf{X}(M_z)(\mathbf{C}^{\sharp}_{2,h}\mathbf{A}_2^{\varphi-1})^{T} \end{bmatrix}$$

$\boldsymbol{\mathcal{T}}_{g,h}$ is as follows:

$$\begin{bmatrix} \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{C}^{\sharp}_{1,g}\mathbf{B}_1\mathbf{U}(z)(\mathbf{C}^{\sharp}_{2,h}\mathbf{B}_2)^{T} & \ldots & \mathbf{C}^{\sharp}_{1,g}\mathbf{B}_1\mathbf{U}(M_z)(\mathbf{C}^{\sharp}_{2,h}\mathbf{B}_2)^{T} \\ \sum_{i=0}^{1}\mathbf{C}^{\sharp}_{1,g}\mathbf{A}_1^{i}\mathbf{B}_1\mathbf{U}(z+1-i)(\mathbf{C}^{\sharp}_{2,h}\mathbf{A}_2^{i}\mathbf{B}_2)^{T} & \ldots & \vdots \\ \vdots & & \vdots \\ \sum_{i=0}^{\varphi-2}\mathbf{C}^{\sharp}_{1,g}\mathbf{A}_1^{i}\mathbf{B}_1\mathbf{U}(z+\varphi-2-i)(\mathbf{C}^{\sharp}_{2,h}\mathbf{A}_2^{i}\mathbf{B}_2)^{T} & \ldots & \sum_{i=0}^{\varphi-2}\mathbf{C}^{\sharp}_{1,g}\mathbf{A}_1^{i}\mathbf{B}_1\mathbf{U}(N_t-i)(\mathbf{C}^{\sharp}_{2,h}\mathbf{A}_2^{i}\mathbf{B}_2)^{T} \end{bmatrix} \tag{3.66}$$

The matrices $\boldsymbol{\mathcal{V}}_{g,h}$ are estimated with:

$$\boldsymbol{\mathcal{V}}_{g,h} \approx \boldsymbol{\mathcal{Y}}^{\sharp}_{g,h} - \widehat{\boldsymbol{\mathcal{T}}}_{g,h} \tag{3.67}$$

and are contained in the tensor $\boldsymbol{\mathcal{A}}$. A diagonal slice of the tensor $\boldsymbol{\mathcal{A}}$ is defined as follows:

$$\boldsymbol{\mathcal{A}}((i-1)pN+1 : ipN, (i-1)N+1 : iN, :) \tag{3.68}$$

Then, each diagonal slice of the tensor $\boldsymbol{\mathcal{A}}$ contains a matrix $\boldsymbol{\mathcal{V}}_{g,h}$ as illustrated in Figure 3.1 and Figure 3.2.

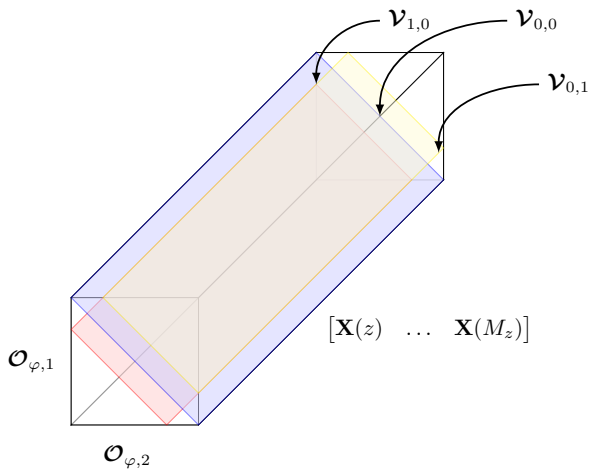


Figure 3.1: Schematic of the tensor $\boldsymbol{\mathcal{A}}$. The position of the observability matrices obtained by writing the data equation for each virtual system are indicated for (g,h) equal to $(0,1),(0,0),(1,0)$.

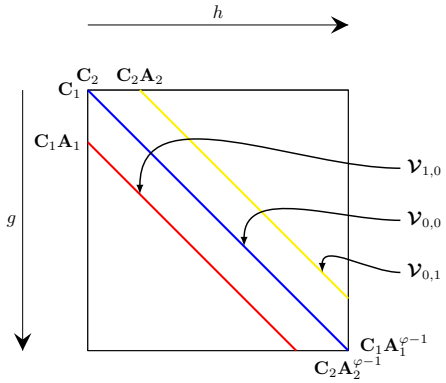


Figure 3.2: Schematic of a slice $\boldsymbol{\mathcal{A}}(:,:,k)$. The position of the observability matrices obtained by writing the data equation for each virtual system are indicated for (g,h) equal to $(0,1),(0,0),(1,0)$.

For example, the block-diagonal terms for each $\boldsymbol{\mathcal{A}}(:,:,k)$ are contained in $\boldsymbol{\mathcal{V}}_{0,0}$, the block-subdiagonal terms are contained in $\boldsymbol{\mathcal{V}}_{1,0}$, the block-superdiagonal terms are

contained in $\boldsymbol{\mathcal{V}}_{0,1}$, etc. To summarize, one diagonal slice is provided by one data equation (with $\boldsymbol{\mathcal{E}} = 0$) corresponding to one MSSM (3.60). The main block-diagonal can be computed without knowing $(\widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{\beta}})$ contrary to all the other slices.

From the estimates of the state sequence $\widehat{\mathbf{X}_\mathbf{T}}(k) = \mathbf{T}_1^{-1}\mathbf{X}(k)\mathbf{T}_2$, the following bilinear least-squares is formulated to recover the matrices $\mathbf{B}_2, \mathbf{B}_1$:

$$\min_{\mathbf{B}_2,\mathbf{B}_1} \sum_{k=z+1}^{M_z-1} \|\big(\widehat{\mathbf{X}_\mathbf{T}}(k+1) - \mathbf{A}_1\widehat{\mathbf{X}_\mathbf{T}}(k)\mathbf{A}_2^T\big) - \mathbf{B}_1\mathbf{U}(k)\mathbf{B}_2^T\|_F^2 \tag{3.69}$$

The minimization problem (3.69) is solved using Alternating Least Squares and starting with a random non-zero initial guess.
The steps for estimating the state-sequence are summarized in Algorithm 3.3.

Algorithm 3.3: Estimation of the state-sequence

Input : $\{\mathbf{u}(k)\}_{1:N_t}, \{\mathbf{y}(k)\}_{1:N_t}, \widehat{\alpha}, \widehat{\beta}, \widehat{\mathbf{M}}_\ell, \widehat{\mathbf{M}}_r, z, \varphi$
Output: $\widehat{\mathbf{A}}_1, \widehat{\mathbf{A}}_2, \widehat{\mathbf{B}}_1, \widehat{\mathbf{B}}_2, \widehat{\mathbf{C}}_1, \widehat{\mathbf{C}}_2$

foreach $\eta = 1 : 2\varphi - 1$ **do**
  if $\eta < \varphi$ **then**
    $g = 0, h = \eta$
  else if $\eta = \varphi$ **then**
    $g = 0, h = 0$
  else if $\eta > \varphi$ **then**
    $g = \eta - \varphi, h = 0.$
  foreach $k = z + 1 : N_t$ **do**
    Compute the virtual outputs with (3.63)
  end
  Compute $\boldsymbol{\mathcal{V}}_{g,h}$ with (3.67)
  `/* Fill-in the tensor` $\boldsymbol{\mathcal{A}}$ `*/`
  foreach $k = z : N_t$ **do**
    Fill block-diag$(\boldsymbol{\mathcal{A}}(:,:,k), \eta - \varphi)$ with the k-th block column of $\boldsymbol{\mathcal{V}}_{g,h}$ according to (3.51).
  end
end
`/* Compute the state-sequence */`
Form the unfolding $\boldsymbol{\mathcal{A}}_{(1)}$ and compute the SVD (3.55)
Estimate $\widehat{\mathbf{A}_1}, \widehat{\mathbf{C}_1}$
Form $\mathbf{H}$ and compute the SVD (3.58)
Estimate $\widehat{\mathbf{A}_2}, \widehat{\mathbf{C}_2}$
`/* Compute the input state-space matrices */`
Solve (3.69) iteratively with ALS and estimate $\widehat{\mathbf{B}_2}, \widehat{\mathbf{B}_1}$

3.5.3. Computational complexity

Computing the virtual outputs requires $(2\varphi-1)N^3N_tz$ flops and therefore, scales with $\mathcal{O}(N^3N_t)$. The first and second SVD cost respectively $\mathcal{O}(N^3M)$ and $\mathcal{O}(\widehat{n_1}N^2M)$. Last, solving the bilinear least-squares in (3.69) requires N^3M_z flops. The overall computational complexity for Algorithm 3.3 is $\mathcal{O}(N^3N_t)$.

3.6. Numerical example

3.6.1. The model

The subspace algorithm is now illustrated with an adaptive optics application. The dimensions of the problem are summarized in Table 3.1.

Model	
$N \times N$ WFS sensor points	$N \in \{6, 8, \ldots, 32\}$
SNR sensor noise	20 dB
D aperture diameter	$N/4$ m
Turbulence	
$m \times m$ turbulence phase screen	$(3N+1) \times (3N+1)$
r_0 Fried parameter	0.2 m
L_0 outer scale	20 m
δ MA neighborhood	50
Horizontal wind speed	3 pixels/sample
Vertical wind speed	0 pixels/sample

Table 3.1: Parameters for the numerical simulation - recursive QUARKS

We aim here at illustrating the subspace identification of Kronecker state-space models in innovation form defined with:

$$\begin{cases} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{K}\mathbf{e}(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{e}(k) \end{cases} \tag{3.70}$$

where $\mathbf{e}(k)$ is a zero-mean white Gaussian noise sequence. The prediction model reads:

$$\widehat{\mathbf{x}}(k+1|k) = \tilde{\mathbf{A}}\widehat{\mathbf{x}}(k|k) + \mathbf{K}\mathbf{y}(k) \tag{3.71}$$

We assume that the matrices $\tilde{\mathbf{A}} = \mathbf{A} - \mathbf{K}\mathbf{C}$, $\mathbf{K}$ and $\mathbf{C}$ have a Kronecker rank one, hence giving rise to the MSSM as follows:

$$\widehat{\mathbf{X}}(k+1|k) = \tilde{\mathbf{A}}_\mathbf{2}\widehat{\mathbf{X}}(k|k)\tilde{\mathbf{A}}_1^T + \mathbf{K}_2\mathbf{Y}(k)\mathbf{K}_1^T \tag{3.72}$$

The performance of Algorithm 3.1 is evaluated for varying size of the lenslet-array and compared to the SSARX method. For each of the 20 realizations, three methods are compared:

- **SSARX**: the centralized identification scheme tailored for the model (3.71). The number of temporal samples is set to the minimum required for the method, $4N^2s + 2s$. The integer s is set to 15.

- **K4SID (Algorithms 3.1+3.2+3.3)** The set of parameters is found below.

- **K4SID+MLDS**: the estimates obtained with K4SID are used for initializing the non-linear Expectation-Maximization algorithm for handling multilinear dynamical systems, Rogers et al. (2013). A number of $10sN$ temporal samples are used for identification and 10 iterations are computed. However, only small sizes of grids with $N \leq 10$ could be handled because of the high computational complexity of MLDS. We have computed a suboptimal model from K4SID by fixing the system orders to lower values than the optimal ones such that MLDS could handle the optimization, i.e n_1, n_2 are set to $2N$.

We summarize the building blocks of the algorithm K4SID along with the chosen parameters used in the simulations below:

- the QUARKS identification (Algorithm 3.1):
  The number of temporal points in the identification set (for Kronecker-based models) is $10sN$. The initial guesses for the matrices $\mathbf{M}_{i,\ell}$ are chosen following a Gaussian distribution with zero-mean and identity covariance matrix. A maximum of 10 iterations is fixed along with a stopping bound ϵ_{min} set to 10^{-5}. The temporal stability of the QUARKS model is ensured by fitting a DC-kernel:
  $$p_{t_{m,n}} = e^{-\eta|m-n|} e^{-\xi/2(m+n)}$$
  for $m, n = 1..s$. The optimization is performed for different hyperparameters λ, η, ξ; the optimal set is found by random search, Bergstra and Bengio (2012), with 10 runs.

- the bilinear low-rank optimization in (3.46) (Algorithm 3.2):
  The algorithm 3.2 for estimating the parameters $\widehat{\boldsymbol{\alpha}}$ and $\widehat{\boldsymbol{\beta}}$ requires an initial value for the regularization parameter, $\mu^{(0)}$. Its value impacts whether the bilinear constraint $\alpha_i \beta_i = 1$ is met. If it is set too low, then both $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are estimated with $\mathbf{0}$, and the bilinear constraint is not met. Therefore, we look for an optimal $\mu^{(0)}$ by grid search. Five values linearly sampled between 1 and 50 are successively as initial value for $\mu^{(0)}$. We call an an optimal value for $\mu^{(0)}$ the one that minimizes the prediction-error for the method K4SID for a particular set of data and grid size. The initial guesses for $\boldsymbol{\alpha}^{(0)}$, $\boldsymbol{\beta}^{(0)}$ are chosen equal to 1.

- State-sequence estimation (Algorithm 3.3):
  The integers φ and z are respectively fixed to $\lfloor \frac{s+1}{2} \rfloor$ and $\varphi - 1$. The system orders $\widehat{n_1}$ and $\widehat{n_2}$ correspond to the index of the singular value that in logarithm is closest to the logarithmic mean of the maximum and minimum singular values for both SVDs (3.55) and (3.58). The maximum number of iterations in the ALS for minimizing (3.69) is set to 10.

The quality criteria is the Variance Accounted For (VAF) between the slopes mea-

surements $\mathbf{y}_{i,j}(k) \in \mathbb{R}^2$ and the predicted $\widehat{\mathbf{y}}_{i,j}(k)$ and is defined with:

$$\max\Big(0, \big(1 - \frac{\frac{1}{N_t}\sum_{k=1}^{N_t} \|\mathbf{y}_{i,j}(k) - \widehat{\mathbf{y}}_{i,j}(k)\|_2^2}{\frac{1}{N_t}\sum_{k=1}^{N_t} \|\mathbf{y}_{i,j}(k)\|_2^2}\big) \times 100\Big)$$

for $N_t = 5\times10^3$ time samples from a validation set independent from the identification set. The VAF is computed for each sensor channel independently, and the mean is taken over the whole measurement grid afterwards. The experiments have been carried out using MatlabR2015b on a desktop computer with a CPU Intel Xeon E5-1620V3/3.5 GHz with 24GB of RAM.

3.6.2. Analyzing the prediction-error

Figure 3.3 displays the VAF on validation data for the four algorithms as a function of the total number of outputs, $2N^2$. We stress that we are not aiming at reaching lower prediction errors than SSARX as structural assumptions are made on the matrices. However, we show that the proposed Kronecker-based modeling handles grids of much larger sizes and with a slight decrease of performance w.r.t the centralized version (when the latter model can be computed).

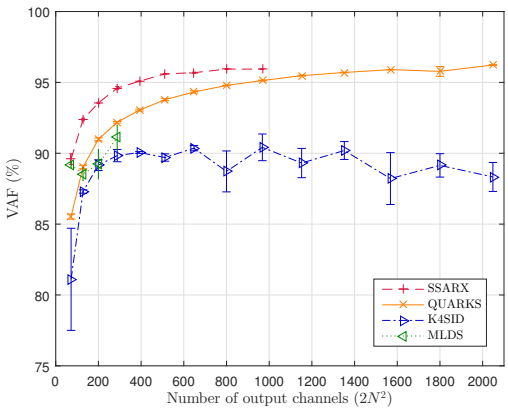


Figure 3.3: VAF (%) on validation data as a function of the number of outputs. No model was computed for SSARX for $N > 20$.

When $N \leq 20$ (that is, $2N^2 \leq 800$), SSARX obtains lower prediction-errors than K4SID. A centralized identification could no longer be carried out for $N > 20$ because of lack of memory which is a well-known problem as explained in Section 3.2. The QUARKS estimation reaches performances similar to the one obtained with SSARX and handles many more outputs. More temporal samples would improve the performance with some impact on the scalability. The performances using K4SID are lower than QUARKS because the low-rank bilinear algorithm (Algorithm 3.2) does not in general converge to the global minimum of (3.45). The difficulties are twofold: both the rank operator and the bilinear constraint have been relaxed to solve a

sequence of convex minimizations rather than a rank minimization problem. The estimates obtained with K4SID serve as good initial values for further optimization using output-error algorithms for Kronecker models as highlighted with the method that combines K4SID and MLDS (in green). However, the latter is computationally cumbersome. Increasing the system order would result in higher performances but in much longer optimization times as MLDS scales with the third power of the global order, that is $\mathcal{O}(N^6)$ for the example considered. Systems could no longer be identified with MLDS for sizes strictly larger than 10×10 and this is corroborated with the timing measurements in the subsection 3.6.4.

3.6.3. Storage complexity

The storage is defined as the number of entries to construct the state-transition matrix $\mathbf{A}$, that is n^2 in the centralized case and $n_2^2 + n_1^2$ in the Kronecker model. We analyse these results further by plotting the storage as a function of the size, which are directly related to the system orders. Figure 3.4 illustrates a dependency of the storage complexity with $N^{4.06}$ for global models while it is only $N^{1.63}$ for the Kronecker-structured model. The order stops increasing for SSARX when $N > 20$ as reaching a user-chosen upper-bound, $n = 2 \cdot 10^3$.

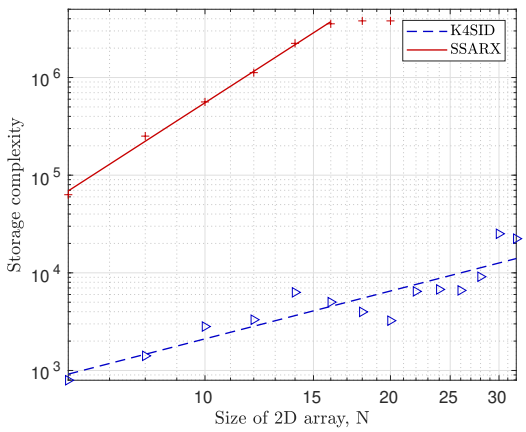


Figure 3.4: Storage complexity as a function of the size of the sensor grid, in log-log scale. The linear model plotted corresponds to $log10(\text{Storage complexity}) = a \times log10(N) + b$.

3.6.4. Timing experiments

We investigate how the computational time for the identification algorithms evolves with N. We lay the emphasis on relative results rather than absolute as the latter are very much hardware-dependent. The SSARX algorithm consists mainly of a QR decomposition, a SVD and a least-squares, while the Kronecker-based methods contains many loops in the QUARKS identification, the bilinear low-rank algorithm and the state-sequence estimation. Consequently, the performances would benefit

from a C implementation. A similar observation is done for the Matlab code used for MLDS. We nonetheless focus on the difference of scaling capabilities of the three algorithms in Figure 3.5. The time for SSARX to identify a model scales with $N^{5.74}$ while it is only $N^{2.49}$ for K4SID. The M-step in the MLDS computes a Kalman smoother, which scales at the very least with $\mathcal{O}(N^6)$ as no Kronecker structure could be exploited in every step and large matrices were used for computations. For example, it requires computing the inverse of the global state covariance matrix at each iteration for each time sample. Consequently, the computational time increases sharply even for moderate sizes of arrays.

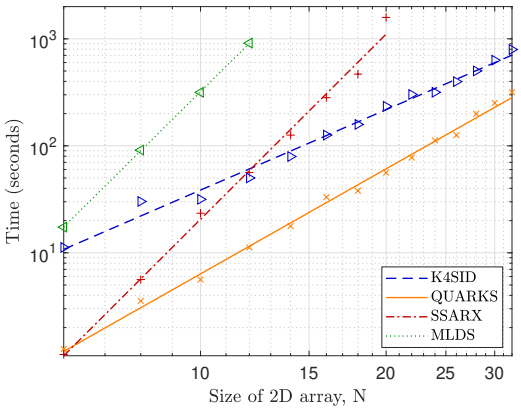


Figure 3.5: Computational time of the model identification as a function of the size of the sensor grid, in loglog scale. The linear model plotted corresponds to $log10(Time) = a \times log10(N) + b$. The regression coefficient is equal to $(2.49, 3.26, 5.74, 5.70)$ for respectively K4SID, QUARKS, SSARX, MLDS.

3.7. Conclusion

Conclusions

In this chapter, we presented a new framework to analyse large scale sensor arrays and identify with $\mathcal{O}(N^3 N_t)$ complexity the state-space matrices when they exhibit a Kronecker rank-1 structure. The algorithm consists of first identifying a QUARKS model in which we estimate the left and right factor matrices up to an unknown parameter that is different for each factored Markov parameter. Next, we formulated some low-rank conditions on a block-Hankel matrix such that the left and right factored impulse responses are retrieved up to a scaling factor. A proposal has been made to use a Block-Coordinate Descent algorithm with slack variables that is solved iteratively and gradually ensures that the bilinear constraint is met. We estimated the state-sequence using two consecutive SVD on a tensor and then, the Kronecker generators from an Alternating Least Squares on the matrix state-space model. The benefits of large-scale modeling with the Kronecker structure have been illustrated with an adaptive optics example and are threefold. First, the Kronecker-

based subspace algorithm handles larger systems than the benchmark SSARX allow. Second, although the method we propose leads to a higher prediction-error than the centralized version, the number of time samples required for using the SSARX method increases with N^2. Last, timing experiments have shown a dependency with $N^{5.65}$ for SSARX instead of $N^{2.40}$ for K4SID.

Recommendations

Improving the estimates using a non-linear output-error algorithm with using the estimates from K4SID as initial estimates.

The estimates obtained with the subspace algorithm can be further refined by using them as initial estimates for Kronecker-based output-error optimization algorithms. The main challenge is to derive them with a complexity not larger than $N^3 N_t$.

The Kronecker rank may not be exactly equal to one in real-life applications.

We have not investigated in this chapter the identification of state-space models in which the matrices (especially $\mathbf{B}, \mathbf{C}$) have a Kronecker rank strictly larger than one, which could help to reduce further on the orders n_2, n_1 while maintaining the same prediction-error performances. When setting $(\mathbf{A}, \mathbf{B}, \mathbf{C}) \in \mathcal{K}_{2,r_A} \times \mathcal{K}_{2,r_B} \times \mathcal{K}_{2,r_C}$, the Kronecker rank of the Markov parameter $\mathbf{C}\mathbf{A}^j\mathbf{B}$ is equal to $r_j := r_C r_A^j r_B$. A similar exponential factor is to be observed for Linear Parameter Varying identification schemes. The exponential increase with r_A is devastating for the performances. It should be assessed whether such model is realistic. This question is again relevant in Chapter 5 when solving discrete Lyapunov equations. In any case, even if r_A is set to one and thus $r_j = r_C r_B =: r$, the bilinear constraint in (3.45) reads $\mathbf{P}\mathbf{Q} = \mathbf{I}_r$ for $\mathbf{P}, \mathbf{Q} \in \mathbb{R}^{r \times r}$.

An alternative to fill the tensor $\boldsymbol{\mathcal{A}}$.

The state-sequence is estimated based on a three-dimensional tensor written as $\boldsymbol{\mathcal{A}}$. The data not lying on the block-superdiagonal was estimated by constructing virtual outputs using FIR models. If the temporal order of these models is too low to guarantee a good accuracy, estimating the set of generators using the state-sequence may be worse than directly with the block-Hankel matrices. Knowing that the three unfoldings of $\boldsymbol{\mathcal{A}}$ are low-rank, that the data lying on the block-superdiagonal is known, and the rest is parametrized as a function of input data and previously estimated quantities, e.g $\widehat{t}_{i,j}, \widehat{\mathbf{M}}_{i,j}$, is it possible to estimate the virtual outputs with $\mathcal{O}(N^3)$ complexity?

4
Scaling up

We extend in this chapter the methods developed in the two previous ones to handle the case where each system matrix is written as a Kronecker product between d matrices, d not necessarily equal to two anymore. This formulation is also well-suited to handle the identification of multi-dimensional systems.

System theory results related to observability, controllability and equivalent classes of systems are first extended from Chapter 3. We then analyse the weaknesses of K4SID and how these motivate two main differences introduced in this chapter: the class of systems considered is restricted to systems where the factored Markov parameters are strictly positive element-wise, and the factored state-space matrices are estimated from a low-rank block-Hankel matrix. An improvement in accuracy and computation time is observed, allowing to scale up the size of systems handled in reasonable time and computing resources.

This chapter is published for the first time in this thesis.
Prof. Hansson contributed to the material presented in this chapter by suggesting to restrict to the class of positive systems in order to linearise the multi-linear equality constraint in the second step of K4SID.

4.1. Introduction

Standard linear algebra operations when identifying the system with K4SID involve matrix multiplications and least-squares such that the overall algorithm scales with $\mathcal{O}(N^3 N_t)$ for N_t the number of temporal samples; hence providing a lower bound on the achievable computational complexity. Although the generators may be parametrized with additional structure such as banded when identifying auto-regressive models, these type of structural patterns on the factor matrices are later not used (and therefore destroyed) when estimating with K4SID the factored state-space matrices because of the unknown similarity transformation relating two equivalent sets of generators. Instead of adding further structure on the factors, this chapter parametrizes the system matrices with more factors in the Kronecker product.

Linear maps between inputs and outputs when no temporal dynamics are involved are often associated to a physical phenomenon (or a device) whose spatial behaviour is measured on a regular grid, e.g the influence matrix $\mathbf{H}$ in (1.22) relates the actuator inputs to the wavefront induced by the mirror. In the latter case, a function representation is available: the influence of each actuator is modelled with a two-dimensional Gaussian function. More generally, the function f maps a set of variables $(x_1, x_2, \ldots, x_d)$ to $\mathbb{R}$ and we may approximate it with a sum of products between d functions ϕ_j^ℓ:

$$f(x_1, \ldots, x_d) \approx \sum_{\ell=1}^{r} \prod_{j=1}^{d} \phi_j^\ell(x_j) \tag{4.1}$$

When, at each time instant, the spatial dynamics are represented by f, the values can equivalently be rewritten into a tensor F such that:

$$F(i_1, \ldots, i_d) \approx \sum_{\ell=1}^{r} \prod_{j=1}^{d} v_j^\ell(i_j), \quad i_j = 1, \ldots, I_j \tag{4.2}$$

where $\mathbf{v}_j^\ell$ is a vector of appropriate size. Mohlenkamp (2013) shows that the multivariate function approximation problem and the tensor approximation problem are the same. The tensor F is able to capture the dynamics of any function f with finite ℓ_2 norm (as defined from the standard inner product in a Hilbert space) when increasing sufficiently r. This tensor F is actually a generalization of the reshuffled matrix we defined in Chapter 2 which we formed using the operator $\mathcal{R}$. For example when f is a Gaussian function defined from $\mathbb{R}^2$ to $\mathbb{R}$, the matrix F has rank one. Consequently, the parametrization of the system matrices with a sum of Kronecker products is very much related to a tensor approximation problem that we make explicit in the second section.

Exploiting the separability of the function f yields a Kronecker product with d factors and the state, input and output are reshuffled into tensors of order d. When the sensor nodes are distributed over d spatial dimensions on a grid of size $N \times \ldots \times N$ (d times), the output vector is reshuffled into a tensor according to the grid dimensions. Alternatively, when the system is two-dimensional, the output signal may also be reshuffled into a tensor of order d such that the product of its dimensions

is equal to N^2, here exploiting the fact that the dimension of the system may be different from the dimension of the model. In other words, we have assumed so far that a parametrization of the system matrices with a Kronecker product between only two matrices are used to model systems whose sensor array is two-dimensional. This assumption is relaxed in this chapter and especially illustrated for adaptive optics in Chapter 6. For example, parametrizing the system matrices with a product of three (or more) Kronecker products may be used when the sensor array is only two-dimensional. The level of accuracy and data-compression wanted is tuned by selecting the Kronecker rank. As a fundament for understanding this particular type of state-space models, we restrict in this chapter to Kronecker rank-one matrices. When the dimension of the model d is larger than the dimension of the sensor array, it increases in general the bias with the true model representation although it can be partly compensated by increasing the Kronecker rank. Identifying state-space matrices written as a sum of Kronecker products is left for future work and was already discussed at the end of Chapter 3.

The questions we would like to answer in this chapter are as follows. Given a two-dimensional sensor grid, is it possible to reshuffle the sensor data into a tensor rather than a matrix with arbitrary choice for the sizes and, therefore, achieve linear computational complexity with respect to the number of nodes, N^2? More generally, what is the computational complexity achievable for identifying the state-space matrices when the vectorized sensor data belongs to $\mathbb{R}^{N^d}$?

We propose in this chapter to extend the methods developed in Chapter 2 and 3 to answer these questions. To cope with the new problem formulation which includes convex optimization with multi-linear constraints and larger datasets, we are willing to improve K4SID.

A first weakness of K4SID is the non-globally convergent block-coordinate update algorithm for handling a low-rank algorithm with bilinear constraints. For a true value of the ambiguity sequence t corresponding to the estimates of the factored Markov parameters, and if the time window used to compute the virtual outputs in $\boldsymbol{\mathcal{A}}$ is sufficiently large, the state sequence is accurately estimated as it only relies on two consecutive SVD. After retrieving the state sequence, the factored matrices related to $\mathbf{B}$ are estimated solving an Alternating Least Squares. However, when handling tensor orders with d larger than 2, forming such a tensor $\boldsymbol{\mathcal{A}}$ and especially computing all the virtual outputs is very involved (also for book-keeping) and requires forming a tensor of order $d+1$ with many missing entries. Two alternatives may be proposed: either formulate a large optimization problem minimizing the rank of each unfolding of this tensor to recover the missing entries, or estimate the state-space matrices without requiring the knowledge of the state-sequence using realization theory on block-Hankel matrices. We study the second option in this chapter.

This chapter is organized as follows. The second section details the equivalence between a sums of Kronecker representation and tensor approximations. The third one introduces tensor state-space models and proposes system theory results to prepare for the numerical procedure presented in the fourth section. This latter section restrains to a class of Kronecker systems where the factored Markov parameters are strictly positive element-wise. The fifth section is a discussion on the proposed

algorithm. Appendix B contains preliminaries on tensors.
Notations specific to the chapter. The inequality sign $>$ is meant element-wise: the relation of order in the set of real strictly positive scalars is such that for two matrices $(\mathbf{X}, \mathbf{Y}) \in \mathbb{R}^{N\times N} \times \mathbb{R}^{N\times N}$, we say that $\mathbf{X} < \mathbf{Y}$ if and only if, for all $(i,j) \in \{1,...,N\}^2, x_{i,j} < y_{i,j}$. Similarly for the exponential and the logarithm functions.

4.2. State-space models for multi-dimensional systems

Let d, I, J three integers. Let $(I_1, \ldots, I_d)$ and $(J_1, \ldots, J_d)$ be two tuples of integers such that $\prod_{j=1}^{d} I_j = I$ and $\prod_{j=1}^{d} J_j = J$. Let $(n_1, \ldots, n_d) \in \mathbb{N}^d$ and $n = \prod_{j=1}^{d} n_j$.

4

Definition 4.1. *Let $r \in \mathbb{N}$. The class of low-Kronecker rank matrices $\mathcal{K}_{d,r}$ contains all matrices $\mathbf{X} \in \mathbb{R}^{J\times I}$ parametrized as follows:*

$$\mathbf{X} = \sum_{\ell=1}^{r} \mathbf{X}_{d,\ell} \otimes \ldots \otimes \mathbf{X}_{1,\ell} \tag{4.3}$$

where, for all $(j,\ell) \in \{1,\ldots,d\} \times \{1,\ldots,r\}$, $\mathbf{X}_{j,\ell} \in \mathbb{R}^{J_j \times I_j}$, and r is the Kronecker rank assumed much smaller than $\min(\{J_j, I_j\}_{j=1..d})$.

This class is related to the CPD of a certain reshuffling. In the case $d = 2$, we have used in Chapter 2 a reshuffling operator $\mathcal{R}$ to write

$$\mathcal{R}(\mathbf{X}) = \sum_{\ell=1}^{r} \mathbf{U}_{2,\ell}\mathbf{U}_{1,\ell}^T = \sum_{\ell=1}^{r} \mathbf{U}_{2,\ell} \circ \mathbf{U}_{1,\ell} \tag{4.4}$$

where $\mathbf{U}_{j,\ell} = \mathrm{vec}(\mathbf{X}_{j,\ell})$. When the matrix $\mathbf{X}$ is a Kronecker product between d terms, the reshuffled operator maps to the d-dimensional space $\mathbb{R}^{J_d I_d \times \ldots \times J_1 I_1}$. It is defined such that:

$$\mathcal{R}(\mathbf{X}) := \mathcal{X} = \sum_{\ell=1}^{r} \mathbf{U}_{d,\ell} \circ \ldots \circ \mathbf{U}_{1,\ell} \tag{4.5}$$

This expression is a CPD of rank r of the tensor $\mathcal{X}$.
For all $j \in \{1,\ldots,d\}$, let $\mathbf{A}_j \in \mathbb{R}^{n_j \times n_j}, \mathbf{B}_j \in \mathbb{R}^{n_j \times I_j}, \mathbf{C}_j \in \mathbb{R}^{J_j \times n_j}$. A LTI system with state space matrices in $\mathcal{K}_{d,1}$ is,

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \\ \mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{e}(k) \end{cases} \tag{4.6}$$

where the measurement noise $\mathbf{e}(k)$ is zero mean white Gaussian with covariance matrix $\sigma_e^2 \mathbf{I}_J$, and such that the state-space matrices are parametrized as:

$$\begin{cases} \mathbf{A} &= \mathbf{A}_d \otimes \ldots \otimes \mathbf{A}_1 \\ \mathbf{B} &= \mathbf{B}_d \otimes \ldots \otimes \mathbf{B}_1 \\ \mathbf{C} &= \mathbf{C}_d \otimes \ldots \otimes \mathbf{C}_1 \end{cases} \tag{4.7}$$

Equivalently, a tensor state-space model (TSSM) reads:

$$\begin{cases} \mathcal{X}(k+1) = \mathcal{X}(k) \times_1 \mathbf{A}_1 \times_2 \ldots \times_d \mathbf{A}_d + \mathcal{U}(k) \times_1 \mathbf{B}_1 \times_2 \ldots \times_d \mathbf{B}_d \\ \mathcal{Y}(k) = \mathcal{X}(k) \times_1 \mathbf{C}_1 \times_2 \ldots \times_d \mathbf{C}_d + \mathcal{E}(k) \end{cases} \tag{4.8}$$

where $\big(\mathcal{X}(k), \mathcal{U}(k), \mathcal{Y}(k), \mathcal{E}(k)\big) \in \mathbb{R}^{n_1 \times \ldots \times n_d} \times \mathbb{R}^{m_1 \times \ldots \times m_d} \times \mathbb{R}^{p_1 \times \ldots \times p_d} \times \mathbb{R}^{p_1 \times \ldots \times p_d}$. We refer to Appendix B for a definition of the symbol $\times_n$. Figure 4.1 depicts the state equation without input for the cases: $d \in \{1, 2, 3\}$.

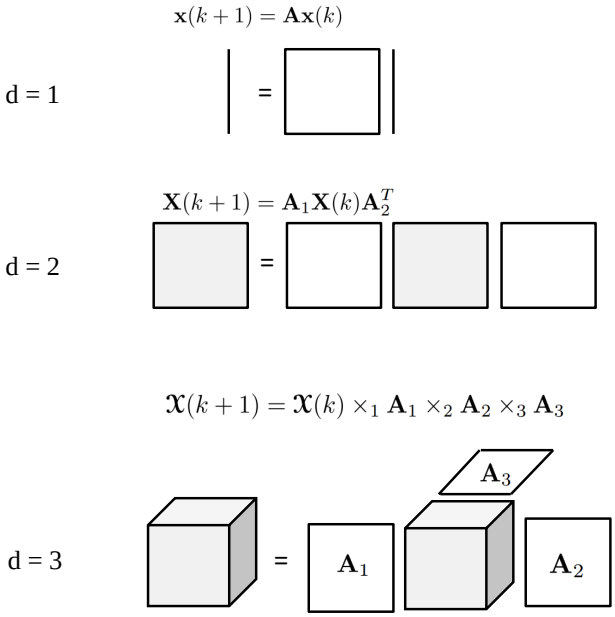


Figure 4.1: Schematic of the state-update equation when d is equal to one (above), two (middle), three (below). The state is represented in grey while the transition matrices are shown in white.

The standard LTI equation in vector form appears for $d = 1$ whereas multi-linear equations are used to represent the dynamics when the state is a d-th order tensor for d larger than one. The n-th mode fiber of the state at instant k is multiplied with rows of $\mathbf{A}_n$. For example when $d = 2$, the rows of $\mathbf{X}(k)$ are multiplied with the rows of $\mathbf{A}_2$ and its columns are multiplied with the rows of $\mathbf{A}_1$.

A large part of the system theory results extend from Chapter 3 and we review them for completeness. The parametrization of the state-space matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ with Kronecker products as in (4.7) is unique up to the trivial indeterminacies (scaling and permutation).

Lemma 4.1. *Let $d \in \mathbb{N}$. If $\lambda_{max}(\mathbf{A}_j) < 1$ for all $j \in \{1, \ldots, d\}$, then $\lambda_{max}(\mathbf{A}_d \otimes \ldots \otimes \mathbf{A}_1) < 1$. The reverse is not true in general.*

Proof. The proof is done by induction on d and builds on existing results derived in Chapter 3. The proposition in the lemma is denoted with $\mathcal{P}(d)$. We initialize with

$d = 1$ for which the implication in the lemma is true. Let us assume $\mathcal{P}(d)$ and prove $\mathcal{P}(d+1)$. Writing $\tilde{\mathbf{A}} := \mathbf{A}_d \otimes \ldots \otimes \mathbf{A}_1$ and using $\mathcal{P}(d)$, we have that $\lambda_{max}(\tilde{\mathbf{A}}) < 1$. Moreover, $\lambda_{max}(\mathbf{A}_{d+1}) < 1$. Using Lemma 3.1, $\lambda_{max}(\mathbf{A}_{d+1} \otimes \tilde{\mathbf{A}}) < 1$ which ends the proof. ■

Lemma 4.2. *Let $d \in \mathbb{N}$ and $j \in \{1, .., d\}$. If the pair $(\mathbf{A}_d \otimes \ldots \otimes \mathbf{A}_1, \mathbf{C}_d \otimes \ldots \otimes \mathbf{C}_1)$ is observable, then each of the pairs $(\mathbf{A}_j, \mathbf{C}_j)$ is observable. The reverse is not true in general.*

Proof. By induction and using Lemma 3.2. ■Let a large LTI system is modeled with an interconnected set of subsystems. If each subsystem is observable, it is not true in general that the global system is observable. For a tensor state space model, if the pairs associated to factor matrices are all observable, the same conclusion holds.

Lemma 4.3. *Let $d \in \mathbb{N}$ and $j \in \{1, .., d\}$. If the pair $(\mathbf{A}_d \otimes \ldots \otimes \mathbf{A}_1, \mathbf{B}_d \otimes \ldots \otimes \mathbf{B}_1)$ is controllable, then each of the pairs $(\mathbf{A}_j, \mathbf{B}_j)$ is controllable. The reverse is not true in general.*

Proof. By induction and using the counterpart of Lemma 3.2 dealing with the controllability matrix. ■

We define here the sets of generators along with their equivalence for yielding an identical input-output behaviour using the tensor model (4.6)-(4.7).

Definition 4.2. *The set of generators $\mathcal{S}$ for the TSSM contains the factored state-space matrices as follows:*

$$\mathcal{S} = \{\{\mathbf{A}_j\}_{j=1..d}, \{\mathbf{B}_j\}_{j=1..d}, \{\mathbf{C}_j\}_{j=1..d}\}$$

A superscript within parenthesis is used to index different sets of generators and is left out when it is clear from context.

Definition 4.3. *Two sets of generators $\mathcal{S}^{(1)}$ and $\mathcal{S}^{(2)}$ are said to be equivalent if the input-output behaviour of the TSSM is identical.*

Lemma 4.4. *Let the input sequence be persistently exciting. Two sets of generators $\mathcal{S}^{(1)}$ and $\mathcal{S}^{(2)}$ equivalently model the TSSM if and only if there exists a similarity transformation $\mathbf{T} \in \mathbb{R}^{n \times n}$ such that $\mathbf{T} = \mathbf{T}_d \otimes \ldots \otimes \mathbf{T}_1$ and:*

$$\left\{ \begin{array}{lcl} \mathbf{A}_d^{(1)} \otimes \ldots \otimes \mathbf{A}_1^{(1)} & = & \mathbf{T}^{-1}(\mathbf{A}_d^{(2)} \otimes \ldots \otimes \mathbf{A}_1^{(2)})\mathbf{T} \\ \mathbf{B}_d^{(1)} \otimes \ldots \otimes \mathbf{B}_1^{(1)} & = & \mathbf{T}^{-1}(\mathbf{B}_d^{(2)} \otimes \ldots \otimes \mathbf{B}_1^{(2)}) \\ \mathbf{C}_d^{(1)} \otimes \ldots \otimes \mathbf{C}_1^{(1)} & = & (\mathbf{C}_d^{(2)} \otimes \ldots \otimes \mathbf{C}_1^{(2)})\mathbf{T} \end{array} \right. \tag{4.9}$$

Proof. If the similarity transformation $\mathbf{T}$ belongs to $\mathcal{K}_{d,1}$ and (4.9) hold, the input-output behaviour for the two sets is the same using classical results from unstructured system identification.

If the input-output behaviour of the two sets of matrices $\mathcal{S}^{(1)}$ and $\mathcal{S}^{(2)}$ is identical

for all time samples, then there exists a similarity transformation $\mathbf{T}$ (as in standard state-space models) such that $\mathbf{A}^{(1)} = \mathbf{T}^{-1}\mathbf{A}^{(2)}\mathbf{T}, \mathbf{B}^{(1)} = \mathbf{T}^{-1}\mathbf{B}^{(2)}, \mathbf{C}^{(1)} = \mathbf{C}^{(2)}\mathbf{T}$. Lemma B.1 allows us to write $\mathbf{T}$ as a sum of Kronecker products with d factors. If both $\mathbf{A}^{(1)}$ and $\mathbf{A}^{(2)}$ have Kronecker rank one, a proof by induction shows that $\mathbf{T}$ has Kronecker rank one, see Chapter 3 for further discussion. Moreover, there exists a matrix $\mathbf{P} \in \mathbb{R}^{I \times I}$ such that $\mathbf{u}(k) = \mathbf{P}\mathbf{u}(k)$. The matrix $\mathbf{P} = \mathbf{I}_I$ boils down to the identity as this equality holds for all inputs $\mathbf{u}(k)$. ■

The number of terms to represent a tensor with its CPD factors as in (4.3) is $rd\sum_{j=1}^{d} J_j I_j$ compared to JI in the unstructured case. The improvements in terms computational complexity with respect to an unstructured parametrization in vector form is shown in Figure 4.2. Computing a state-update cost n operations and is linear with respect to the size of the state-vector. We refer to Appendix B for a detailed count.

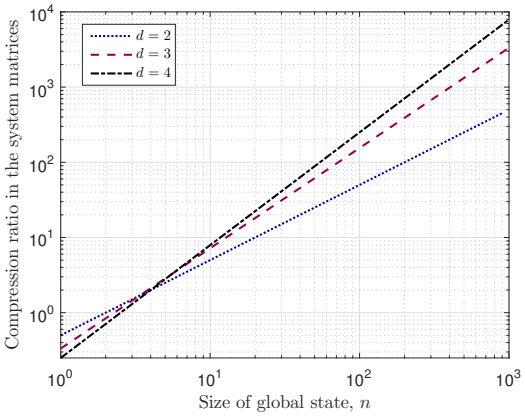


Figure 4.2: Compression ratio between the unstructured matrix $\mathbf{A}$ and its Kronecker parametrization for different tensor order.

Definition 4.4. *Farina (2002) A LTI system is said to be externally (strictly) positive if and only if for any (strictly) positive input and initial state vector, the output is (strictly) positive.*

Theorem 4.1. *Farina (2002) A LTI system is externally (strictly) positive if and only if its impulse response is (strictly) positive element-wise.*

Such a definition is different from the definition of internally positive system as studied in e.g Rantzer (2011). A system is said to be internally positive if and only if its state and output are non-negative when the input and initial state are non-negative. As a corollary, a system is internally positive if the matrices $\mathbf{B}, \mathbf{C}$ are non-negative and the non-diagonal entries in $\mathbf{A}$ are non-negative. Therefore, all internally positive systems are externally positive while the reverse is not true in general. An externally strictly positive system may not admit a decomposition with matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$ all strictly positive element-wise.

Lemma 4.5. *Let a LTI TSSM model as in* (4.6)-(4.7)*. If, for all* $j \in \{1, \ldots, d\}$*, the LTI system defined from the triplet* $(\mathbf{A}_j, \mathbf{B}_j, \mathbf{C}_j)$ *is externally strictly positive, then the LTI TSSM model as in* (4.6)-(4.7) *is externally strictly positive.*

Proof. Let $s \in \mathbb{N}$. We have, for all $(j, i) \in \{1, \ldots, d\} \times \{1, \ldots, s\}$, $\mathbf{C}_j \mathbf{A}_j^i \mathbf{B}_j > 0$. It implies $\mathbf{C}_d \mathbf{A}_d^i \mathbf{B}_d \otimes \ldots \otimes \mathbf{C}_1 \mathbf{A}_1^i \mathbf{B}_1 = \mathbf{C}\mathbf{A}^i\mathbf{B} > 0$. ■

In this chapter, we assume:

- **A1:** The pair $(\mathbf{A}_d \otimes \ldots \otimes \mathbf{A}_1, \mathbf{C}_d \otimes \ldots \otimes \mathbf{C}_1)$ is observable.
- **A2:** The pair $(\mathbf{A}_d \otimes \ldots \otimes \mathbf{A}_1, \mathbf{B}_d \otimes \ldots \otimes \mathbf{B}_1)$ is controllable.
- **A3:** For all $j \in \{1, \ldots, d\}$, the spectral radius of $\mathbf{A}_j$ is strictly inferior to 1.
- **A4:** The matrix $\begin{bmatrix} \mathbf{u}(1) & \ldots & \mathbf{u}(N_t) \end{bmatrix}$ is full row rank.
- **A5:** The measurement noise is zero-mean white noise with unknown covariance matrix.
- **A6:** The measurement noise is uncorrelated with all past inputs.
- **A7:** For all $j \in \{1, \ldots, d\}$, the LTI system defined from the triplet $(\mathbf{A}_j, \mathbf{B}_j, \mathbf{C}_j)$ is externally strictly positive.

4.3. Subspace-like algorithm, SEP-T4SID

The algorithm we now describe is referenced using the acronym SEP-T4SID standing for Tensor Structured State-Space System IDentification assuming Strictly Externally Positivity.

4.3.1. Identification of tensor auto-regressive models

High-order QUARKS

Let s_1 a positive integer and $s = 2s_1 - 1$. The parameter s is the length of the FIR filter in the QUARKS and should be chosen such that, first, for all $j \in \{1, \ldots, d\}, n_j < s_1 \min(I_j, J_j)$ to ensure that a block-Hankel matrix built from the factored Markov parameters is low-rank, and second, such that the approximation error in the QUARKS is small.

Let $\mathbf{M}_{j,i} := \mathbf{C}_j \mathbf{A}_j^i \mathbf{B}_j$. We estimate in this section $\widehat{\mathbf{M}}_{j,i}$ such that $\mathbf{M}_{j,i} = t_{j,i}\widehat{\mathbf{M}}_{j,i}$ and $t_{j,i}$ is an ambiguity factor necessarily non-zero. All factored Markov parameters are strictly positive element-wise. When identifying a QUARKS model, each least-squares is solved with strict positivity constraints on all the elements of each factor matrix.

A FIR model derived from (4.6)-(4.7) reads as:

$$\mathbf{y}(k) = \sum_{i=1}^{s} (\mathbf{M}_{d,i} \otimes \ldots \otimes \mathbf{M}_{1,i}) \mathbf{u}(k-i) + \mathbf{e}(k) \tag{4.10}$$

The sensor data is available for N_t temporal samples. Similarly as in Chapter 2 and Hoff (2015), we formulate an Alternating Least-Squares algorithm for estimating

the factor matrices. If we assume that the factor matrices $\mathbf{M}_{n,i}$ are known for all $(n,i) \in \{1,...,j-1,j+1,...,d\} \times \{1,...,s\}$, then we wish to identify the remaining ones from the cost function:

$$\begin{aligned} \min_{\{\mathbf{M}_{j,i}\}_{i=1..s}} \quad & \sum_{k=s+1}^{N_t} \|\mathbf{y}(k) - \sum_{i=1}^{s} (\mathbf{M}_{d,i} \otimes \ldots \otimes \mathbf{M}_{1,i})\mathbf{u}(k-i)\|_2^2 \\ \text{s.t} \quad & \forall i \in \{1,\ldots,s\}, \mathbf{M}_{j,i} > 0 \end{aligned} \tag{4.11}$$

Now forming the j-th mode reshuffling of $\mathcal{Y}(k)$ and $\mathcal{U}(k)$ denoted respectively with $\mathbf{Y}_{(j)}(k)$ and $\mathbf{U}_{(j)}(k)$, the cost function (4.11) equivalently reads, for $j = 1..d$:

$$\begin{aligned} \min_{\{\mathbf{M}_{j,i}\}_{i=1..s}} \quad & \sum_{k=s+1}^{N_t} \|\mathbf{Y}_{(j)}(k) - \sum_{i=1}^{s} \mathbf{M}_{j,i}\mathbf{F}_{i,j}(k-i)\|_F^2 \\ \text{s.t} \quad & \forall i \in \{1,\ldots,s\}, \mathbf{M}_{j,i} > 0 \end{aligned} \tag{4.12}$$

where the regression matrix is written as:

$$\mathbf{F}_{j,i}(k-i) = \mathbf{U}_{(j)}(k-i)\big(\mathbf{M}_{d,i} \otimes \ldots \otimes \mathbf{M}_{j+1,i} \otimes \mathbf{M}_{j-1,i} \otimes \ldots \otimes \mathbf{M}_{1,i}\big)^T \tag{4.13}$$

For a more compact notation, let:

$$\begin{aligned} \mathbf{M}_j &= \begin{bmatrix} \mathbf{M}_{j,1} & \ldots & \mathbf{M}_{j,s} \end{bmatrix} \\ \bar{\mathbf{F}}_j &= \begin{bmatrix} \mathbf{F}_{j,1}(s) & \ldots & \mathbf{F}_{j,1}(N_t-1) \\ \vdots & & \vdots \\ \mathbf{F}_{j,s}(1) & \ldots & \mathbf{F}_{j,s}(N_t-s) \end{bmatrix} \in \mathbb{R}^{sI_j \times (N_t-s)\prod_{i=1,i\neq j}^{d} J_i} \end{aligned}$$

Let μ be a small positive integer. The indicator function on the set $\mathcal{C}_j = \{\mathbf{M}_j \in \mathbb{R}^{J_j \times sI_j} : \mathbf{M}_j \geq \mu\}$ is denoted with $\mathcal{I}_j(.)$. The optimization (4.12) is then rewritten into:

$$\min_{\mathbf{M}_j} \quad \|\bar{\mathbf{Y}}_j - \mathbf{M}_j\bar{\mathbf{F}}_j\|_F^2 + \mathcal{I}_j(\mathbf{M}_j) \tag{4.14}$$

We assume $\bar{\mathbf{F}}_j$ is full row rank. To satisfy this condition, the number of time samples is such that $\bar{\mathbf{F}}_j$ is flat and the initial guesses are randomly chosen. The larger d, the less temporal samples N_t need to be acquired as the smallest dimension of $\mathbf{M}_j$ decreases.

A main idea to reduce the computational complexity for solving (4.14) is to avoid forming the Kronecker products in (4.13) but use rather the tensor form of FIR models which uses j-mode matrix products. Following Proposition B.1, each term $\mathbf{F}_{j,i}(k-i)$ should be computed using:

$$\mathcal{U}(k-i) \times_d \mathbf{M}_{d,i}^T \times \ldots \times_{j+1} \mathbf{M}_{j+1,i}^T \times_j \mathbf{I} \times_{j-1} \mathbf{M}_{j-1,i}^T \times \ldots \times_1 \mathbf{M}_{1,i}^T \tag{4.15}$$

The entries are eventually reshuffled to form $\mathbf{F}_{j,i}(k-i)$ subsequently concatenated into $\bar{\mathbf{F}}_j$.

The subproblem (4.14) is solved with ADMM. The algorithm is summarized in Algorithm 4.1.

Algorithm 4.1: QUARKS for tensorized data

Input : $\{\mathbf{u}(k), \mathbf{y}(k)\}_{k=1:N_t}, s, \{I_j, J_j\}_{j=1..d}$

Output : $\{\widehat{\mathbf{M}}_{j,i}\}_{(j,i)\in\{1,\dots,d\}\times\{1,\dots,s\}}$

`/* Prepare the data */`

for $j = 1..d$ **do**
  for $k = s+1..N_t$ **do**
    Form the unfoldings $\mathbf{Y}_{(j)}(k)$
  end
  Form $\bar{\mathbf{Y}}_j := \begin{bmatrix}\mathbf{Y}_{(j)}(s+1) & \dots & \mathbf{Y}_{(j)}(N_t)\end{bmatrix}$
  `/* Initial guesses */`
  for $i = 1..s$ **do**
    $\mathbf{M}_{j,i}^{(0)} = \text{rand}(J_j, I_j)$
  end
end

`/* Default values */`

$\ell = 0, \ell_{max} = 30, \epsilon = \infty, \epsilon_{min} = 10^{-3}$

`/* Start ALS */`

while $\ell < \ell_{max}$ *and* $\epsilon > \epsilon_{min}$ **do**
  for $j = 1..d$ **do**
    Form $\bar{\mathbf{F}}_j^{(\ell)}$, and compute its pseudo-inverse
    `/* Solve with ADMM */`
    $\mathbf{N} = (\bar{\mathbf{F}}_j^{(\ell)}\bar{\mathbf{F}}_j^{(\ell),T})^{-1}$ and $\mathbf{G} = \mathbf{N}\bar{\mathbf{F}}_j^{(\ell)}\bar{\mathbf{Y}}_j^T$
    $\kappa = 0$; if possible warm-start $\mathbf{Z}_j^{(\ell,0)}$ and $\mathbf{\Gamma}_j^{(\ell,0)}$
    while *stopping criterion not met* **do**
      $\mathbf{M}_j^{(\ell,\kappa+1)} = \mathbf{G} + \mathbf{N}(\mathbf{\Gamma}_j^{(\ell,\kappa)} - \mathbf{Z}_j^{(\ell,\kappa)})$
      Update the consensus variable:
      $\mathbf{\Gamma}_j^{(\ell,\kappa+1)} = \max(\mu, \mathbf{M}_j^{(\ell,\kappa+1)} + \mathbf{Z}_j^{(\ell,\kappa)})$
      Update the dual variable: $\mathbf{Z}_j^{(\ell,\kappa+1)} = \mathbf{Z}_j^{(\ell,\kappa)} + \mathbf{M}_j^{(\ell,\kappa+1)} - \mathbf{\Gamma}_j^{(\ell,\kappa+1)}$
      Check stopping criterion
      $\kappa = \kappa + 1$
    end
    Denote the optimal value with $\mathbf{M}_j^{(\ell+1)}$
  end
  $c(\ell) = ||\bar{\mathbf{Y}}_d - \mathbf{M}_d^{(\ell)}\bar{\mathbf{F}}_d||_F^2$
  $\epsilon = |c(\ell) - c(\ell-1)|$
  $\ell = \ell + 1$
end

Set $\widehat{\mathbf{M}}_{i,j}$ to the optimal values

Computational complexity
The computational complexity for the most expensive operations is summarized in the Table 4.1. The bottleneck is to form the matrix $\bar{\mathbf{F}}_i$ although the computational complexity is asymptotically in d linear with respect to the number of sensor nodes, I. Figure 4.3 displays the exponent on the total number of sensor nodes I as written in Table 4.1 as a function of the tensor order.

Operation	**Flops**	**With** $I_i = I_j, J_i = J_j, I_i = J_i$
Form $\bar{\mathbf{F}}_j$	$(N_t - s)sI\sum_{j=1}^{d} J_j$	$\mathcal{O}(N_t I^{(d+1)/d})$
Compute $\bar{\mathbf{F}}_j\bar{\mathbf{F}}_j^T$	$(I_j s)^2(N_t - s)\prod_{i=1,i\neq n}^{d} J_i$	$\mathcal{O}(N_t I^{(d+1)/d})$
Invert $\bar{\mathbf{F}}_j\bar{\mathbf{F}}_j^T$	$(I_j s)^3$	$\mathcal{O}(I^{3/d})$

Table 4.1: Computational complexity for the most expensive operations in the QUARKS

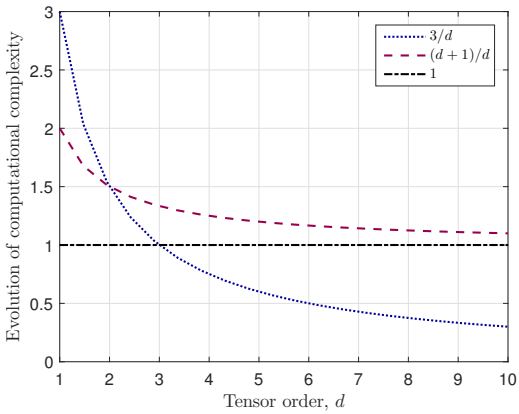


Figure 4.3: Exponent on the total number of nodes I referring to Table 4.1 as a function of the tensor order. The black line corresponds to a linear complexity with respect to the number of nodes in the array.

When $d = 1$, the cost for the identification is cubic with I which just corresponds to the unstructured identification. When the tensor order is larger than two, the bottlenecks are to form $\mathbf{F}_i$ and multiply it with its transpose. When dealing with two-dimensional sensor arrays, it is expected that d cannot be increased to very large values without losing much accuracy in the identification. However, most of the improvements are obtained within the first few tensor orders as $d = 4$ means a complexity of $\mathcal{O}(I^{1.25})$ instead of $\mathcal{O}(I^3)$ in the full unstructured case. Practically, such trends hold only from a given size of the sensor array N which depends on the implementation and the hardware.

Remark 4.1. *The complexity is cubic with s. Similarly, when generalizing the QUARKS for Kronecker rank strictly larger than one, the complexity is cubic with r which reminds of the cubic complexity with the system order of the underlying SSS matrix for linear algebra operations, Rice (2010).*

4.3.2. Low-rank optimization subject to multi-linear equality constraints

Problem formulation

We assume that, for all $(j,i) \in \{1,\dots,d\} \times \{1,\dots,s\}$, we have an estimate $\widehat{\mathbf{M}}_{j,i}$ with strictly positive entries for the Markov parameters $\mathbf{M}_{j,i}$ up to an unknown ambiguity parameter $t_{j,i}$. Only $\widehat{\mathbf{M}}_{j,i}$ is known in the equation $\mathbf{M}_{j,i} = t_{j,i}\widehat{\mathbf{M}}_{j,i}$.

Lemma 4.6. *Let $(j,i) \in \{1,\dots,d\} \times \{1,\dots,s\}$. If $\mathbf{M}_{j,i} > 0$ and $\widehat{\mathbf{M}}_{j,i} > 0$, then $t_{j,i} > 0$.*

We denote, for $j \in \{1,...,d\}, \mathbf{t}_j = \begin{bmatrix} t_{j,1} & \dots & t_{j,s} \end{bmatrix}$ and $\mathbf{t} = \begin{bmatrix} \mathbf{t}_1 & \dots & \mathbf{t}_d \end{bmatrix}$. The block-Hankel matrix built from the sequence $\{\mathbf{M}_{j,i}\}_{i\in\{1,\dots,s\}}$ is defined with:

$$\mathcal{H}_j(\mathbf{t}_j) := \begin{bmatrix} t_{j,1}\widehat{\mathbf{M}}_{j,1} & t_{j,2}\widehat{\mathbf{M}}_{j,2} & \dots & t_{j,s_1}\widehat{\mathbf{M}}_{j,s_1} \\ t_{j,2}\widehat{\mathbf{M}}_{j,2} & t_{j,3}\widehat{\mathbf{M}}_{j,3} & \dots & t_{j,s_1+1}\widehat{\mathbf{M}}_{j,s_1+1} \\ \vdots & & & \vdots \\ t_{j,s_1}\widehat{\mathbf{M}}_{j,s_1} & \dots & \dots & t_{j,s}\widehat{\mathbf{M}}_{j,s} \end{bmatrix}$$

To ensure that $\widehat{\mathbf{M}}_{d,i} \otimes \dots \otimes \widehat{\mathbf{M}}_{1,i} = \mathbf{M}_{d,i} \otimes \dots \otimes \mathbf{M}_{1,i}$, which corresponds to the non-unique decomposition of each Markov parameter with a Kronecker product of d matrices, the following condition necessarily holds:

$$\forall i \in \{1,...,s\}, \quad \prod_{j=1}^{d} t_{j,i} = 1 \tag{4.16}$$

In order to recover the terms $\mathbf{M}_{j,i}$, we wish to estimate the sequence $\mathbf{t}$ using the low-rank priors on $\mathcal{H}_j(\mathbf{t}_j)$ subject to a multi-linear equality constraint:

$$\begin{aligned} \min_{\mathbf{t}} \quad & \sum_{j=1}^{d} \operatorname{rank}(\mathcal{H}_j(\mathbf{t}_j)) \\ \text{s.t} \quad & \forall i \in \{1,...,s\}, \prod_{j=1}^{d} t_{j,i} = 1 \quad t_{j,i} > 0 \end{aligned} \tag{4.17}$$

The solution to the optimization problem (4.17) has been characterized in Chapter 3 when $d = 2$. We extend an important result.

Lemma 4.7. *The solution to the optimization problem* (4.17) *is not unique.*

Proof. Let $(j,i) \in \{1,...,d\} \times \{1,...,s\}$. The optimal solution of (4.17) is such that the ranks are all minimum, $\operatorname{rank}(\mathcal{H}_j(\mathbf{t}_j)) = n_j$. Let $\alpha_{j,i}$ non-zero such that $\widehat{\mathbf{M}}_{j,i} = \alpha_{j,i}\mathbf{M}_{j,i}$. Then, there exists scalars (a_j, c_j) such that $t_{j,i}\alpha_{j,i} = c_j a_j^i$ using the result that the Hankel matrix built from the sequence $\{t_{j,i}\alpha_{j,i}\}_{i=1..s}$ has rank one as shown in Chapter 4. We have:

$$\prod_{j=1}^{d} t_{j,i} = \prod_{j=1}^{d} \frac{c_j a_j^i}{\alpha_{j,i}} = 1$$

Using the fact that $\prod_{j=1}^{d}\alpha_{j,i}=1$, it follows $\prod_{j=1}^{d}c_j a_j^i=\underbrace{\left(\prod_{j=1}^{d}c_j\right)}_{c}\underbrace{\left(\prod_{j=1}^{d}a_j\right)}_{a}^{i}=1$, and then, both a,c are equal to one. The pair (a,c) is therefore unique which is not the case for the sequences built from the c_j and a_j. ■

We now operate the change of variable $v_{j,i}=\ln(t_{j,i})$ to transform the multi-linear product (4.16) into a sum and ease the numerical optimization. This step is the only reason why we have assumed externally strictly positive systems in this chapter. The logarithm function is a bijection from $\mathbb{R}^{+,*}$ to $\mathbb{R}$ and therefore, the corresponding optimal values for $t_{j,i}$ are obtained using the exponential function: $\widehat{t}_{j,i}=e^{\widehat{v}_{j,i}}$. After relaxing the rank constraint with the nuclear norm, the minimization (4.17) reads:

$$\begin{aligned}\min_{v_{j,i}}\quad & \sum_{j=1}^{d}\|\mathcal{H}_j(\{e^{v_{j,i}}\}_{i\in\{1,\dots,s\}})\|_\star \\ \text{s.t}\quad & \forall i\in\{1,\dots,s\},\sum_{j=1}^{d}v_{j,i}=0\end{aligned} \tag{4.18}$$

The optimization (4.18) is solved with ADMM to handle the non-differentiable term and a sequence of convex minimizations as in the second step of K4SID is not needed anymore. At each iteration, the update of the primal variable is obtained with a gradient descent accompanied with a backtracking line search. The iterations stop when the decrease in the cost function between two consecutive iterations is below a threshold.

Computational complexity

Each iteration involves linear algebra operations on matrices of size $s_1J_j\times s_1I_j$. Now assuming for all $i,j=1..d, I_i=I_j$ and $I_i=J_i$, the singular value soft-thresholding step requires $ds_1^3I^{3/d}$ operations.

4.3.3. Realization

The solution $\widehat{\mathbf{t}}$ obtained in the previous paragraph was characterized in Lemma 4.7:

$$t_{j,i}\widehat{\mathbf{M}}_{j,i}=t_{j,i}\alpha_{j,i}\mathbf{M}_{j,i}=c_j a_j^i\mathbf{M}_{j,i} \tag{4.19}$$

The scalar c_j is only a scalar and can be factorized as a product of two scalars e.g p_j,q_j. Let us denote the global optimum of the optimization (4.18) is $\{\mathbf{t}_{opt,j}\}_{j=1..d}$. The block-Hankel matrix $\mathcal{H}_j(\mathbf{t}_{opt,j})$ is then:

$$\mathcal{H}_j(\mathbf{t}_{opt,j})=\begin{bmatrix} p_jq_j\mathbf{M}_{j,1} & p_jq_ja_j\mathbf{M}_{j,2} & \dots & p_jq_ja_j^{s_1-1}\mathbf{M}_{j,s_1}\\ p_jq_ja_j\mathbf{M}_{j,2} & p_jq_ja_j^2\mathbf{M}_{j,3} & \dots & p_jq_ja_j^{s_1}\mathbf{M}_{j,s_1+1}\\ \vdots & & & \vdots\\ p_jq_ja_j^{s_1-1}\mathbf{M}_{j,s_1} & \dots & \dots & p_jq_ja_j^{s-1}\mathbf{M}_{j,s}\end{bmatrix} \tag{4.20}$$

whos rank is equal to n_j:

$$\mathcal{H}_j(\mathbf{t}_{opt,j}) = \begin{bmatrix} p_j\mathbf{C}_j \\ p_j\mathbf{C}_j(a_j\mathbf{A}_j) \\ \vdots \\ p_j\mathbf{C}_j(a_j\mathbf{A}_j)^{s_1-1} \end{bmatrix} \begin{bmatrix} q_j\mathbf{B}_j & (a_j\mathbf{A}_j)q_j\mathbf{B}_j & \dots & (a_j\mathbf{A}_j)^{s_1-1}q_j\mathbf{B}_j \end{bmatrix} \tag{4.21}$$

The vector $\mathbf{t}_{opt,j}$ is not equal to $\widehat{\mathbf{t}}_j$ because of the non-convex nature of (4.18). After selecting the system order $\widehat{n}_j$ from an SVD of $\mathcal{H}_j(\widehat{\mathbf{t}}_j)$, the matrices $\widehat{\mathbf{A}_j}, \widehat{\mathbf{B}_j}, \widehat{\mathbf{C}_j}$ are estimated via the standard realization steps derived in Algorithm 4.2.

Algorithm 4.2: Realization steps

Input : $\widehat{\mathbf{t}}, \{\widehat{\mathbf{M}}_{j,i}\}_{j=1..d,i=1..s}$
Output: $\{\widehat{\mathbf{A}}_j, \widehat{\mathbf{B}}_j, \widehat{\mathbf{C}}_j\}_{j=1..d}$

1 Compute a SVD, $\mathcal{H}_j(\widehat{\mathbf{t}}_j) = \mathbf{U}_j\mathbf{\Sigma}_j\mathbf{V}_j^T$
2 Select the system order, $\widehat{n}_j$
3 Denote: $\mathbf{U}_{j,\widehat{n}_j} = \mathbf{U}_j(:, 1:\widehat{n}_j)$ and $\mathbf{V}_{j,\widehat{n}_j} = \mathbf{\Sigma}_j(1:\widehat{n}_j, 1:\widehat{n}_j)\mathbf{V}_j(:, 1:\widehat{n}_j)^T$
/* Estimate $\mathbf{B}_j$ and $\mathbf{C}_j$ */
4 Extract $\widehat{\mathbf{B}_j} = \mathbf{V}_{j,\widehat{n}_j}(:, 1:I_j)$ and $\widehat{\mathbf{C}_j} = \mathbf{U}_{j,\widehat{n}_j}(1:J_j, :)$
/* Estimate $\mathbf{A}_j$ */
5 $\mu = 10^{-6}$, $\widehat{\mathbf{A}}_j = \mathbf{I}_{\widehat{n}_j}$
6 **while** $\widehat{\mathbf{A}}_j$ *is not strictly stable* **do**
7 Solve the regularized least-squares:

$$\min_{\mathbf{A}_j} \|\mathbf{U}_{j,\widehat{n}_j}(J_j+1:s_1J_j, :) - \mathbf{U}_{j,\widehat{n}_j}(1:(s_1-1)J_j, :)\mathbf{A}_j\|_F^2 + \mu\|\mathbf{A}_j\|_F^2$$

Denote the solution with $\widehat{\mathbf{A}}_j$
8 $\mu = 10 \cdot \mu$
9 **end**

Lemma 4.8. *Let $\mu = 0$ in Algorithm 4.2. The set of factored matrices estimated from Algorithm 4.2 is* not *an equivalent realization of the TSSM* (4.6)-(4.7).

Proof. There exists a similarity transformation $\mathbf{T}_j \in \mathbb{R}^{n_j \times n_j}$ such that:

$$\widehat{\mathbf{A}_j} = \mathbf{T}_j a_j \mathbf{A}_j \mathbf{T}_j^{-1}, \quad \widehat{\mathbf{B}_j} = \mathbf{T}_j q_j \mathbf{B}_j, \quad \widehat{\mathbf{C}_j} = p_j \mathbf{C}_j \mathbf{T}_j^{-1} \tag{4.22}$$

Using the fact that $\prod_{j=1}^{d} a_j = 1$ (see Lemma 4.7), we write:

$$\widehat{\mathbf{A}_d} \otimes \dots \otimes \widehat{\mathbf{A}_1} = \prod_{j=1}^{d} a_j \mathbf{T}\mathbf{A}\mathbf{T}^{-1} = \mathbf{T}\mathbf{A}\mathbf{T}^{-1} \tag{4.23}$$

However, it is not true that $\prod_{j=1}^{d} p_j = 1$ nor that $\prod_{j=1}^{d} q_j = 1$, and therefore, only

$$\widehat{\mathbf{C}_d} \otimes \ldots \otimes \widehat{\mathbf{C}_1} \neq \mathbf{C}\mathbf{T}^{-1}, \quad \widehat{\mathbf{B}_d} \otimes \ldots \otimes \widehat{\mathbf{B}_1} \neq \mathbf{T}\mathbf{B} \tag{4.24}$$

We have nonetheless $\prod_{j=1}^{d} p_j q_j = 1$ implying:

$$\widehat{\mathbf{C}_d}\widehat{\mathbf{B}_d} \otimes \ldots \otimes \widehat{\mathbf{C}_1}\widehat{\mathbf{B}_1} = \prod_{j=1}^{d} p_j q_j \mathbf{C}\mathbf{B} = \mathbf{C}\mathbf{B} \tag{4.25}$$

There are infinite possibilities for choosing p_j, q_j such that the Hankel matrix built from $\{p_j q_j a_j^i\}_{i\in\{1,\ldots,s\}}$ is rank one. ■

The remaining question is whether it matters or not. When the initial state is zero, the input-output relationship is given by an infinite impulse response and none of the terms $\mathbf{B}$ or $\mathbf{C}$ appears separately. In such case, the input-output relationship is identical as for the original system matrices. It is not the case when the initial state is not zero although its influence decays with time.

This lemma is also the reason why we have introduced the tensor $\boldsymbol{\mathcal{A}}$ in Chapter 3 in order to first estimate the state sequence, and second the factor matrices.

4.4. Numerical experiments

We present Monte-Carlo simulations based on randomly generated deterministic state-space models. We assume that the sensor (and input) array is such that $I_i = I^{1/d}$. The system order for each factor matrix is set to $I_i + 1$. The entries in the generators are randomly generated between 0 and 1 for $\mathbf{B}_i$ and $\mathbf{C}_i$. Each matrix $\mathbf{A}_i$ is set diagonal to control its eigenvalues. Especially, and to allow a fair comparison between the different tensor orders, $\mathbf{A}_i$ is equal to $0.96^{1/d}\mathbf{I}_{n_i}$ such that the eigenvalues of the global matrix $\mathbf{A}$ are all equal to 0.96 whatever the tensor order. The SNR is set to 20dB. Low SNR is mainly compensated by increasing the length of the identification dataset in the QUARKS or adding regularization and has already been investigated in Chapter 2, which justifies that we do not carry out a noise analysis in this section.

The length of the past temporal window s is 15 and there are $10dsI^{1/d}$ paramaters in both the identification and validation batches. The baseline is also a three-step algorithm exploiting the Kronecker structure solving first the QUARKS as in Algorithm 4.1, then assuming that $\widehat{\mathbf{t}} = 1$ and last, estimating the matrices with Algorithm 4.2. The system order is selected by grid search in a limited range. The second algorithm used for comparison solves a BCU update instead of (4.18). It is denoted with BCU-T4SID. It is especially of interest as none offer any theoretical guarantee of global convergence.

Accuracy

Let us denote the singular values of the matrix $\mathcal{H}_j(\widehat{\mathbf{t}}_j)$ with $\sigma_{j,i}$ for all $i \in \{1, \ldots, s_1 I^{1/d}\}$. Figure 4.4 compares the ratio defined as:

$$\frac{1}{d}\sum_{j=1}^{d} \frac{\sum_{i=1}^{\alpha} \sigma_{j,i}}{\sum_{i=1}^{s_1 I^{1/d}} \sigma_{j,i}} \tag{4.26}$$

as a function of α for all three methods. The quicker it reaches one, the more sparse the vector of the singular values, and the better the singular values have been separated from the noise contribution (the latter stemming from both the measurement noise, and the non-globally convergent behaviour of the second step). Figure 4.4 shows the significant improvement of both BCU and (4.18) with respect to the baseline $\widehat{\mathbf{t}} = \mathbf{1}$. The ratio has a smaller variance optimizing with (4.18) than with the BCU.

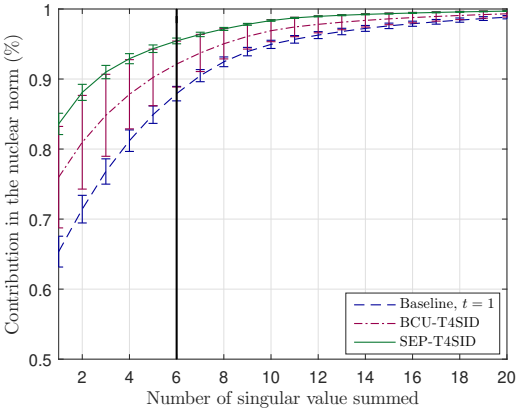


Figure 4.4: Ratio evaluating the sparsity of the singular values vector for the three methods: $\widehat{\mathbf{t}} = 1$, $\widehat{\mathbf{t}}$ obtained with BCU or (4.18). The black vertical line indicates the true rank of $\mathcal{H}_j(\mathbf{t}_{opt,j})$. The tensor order is equal to 4 and $I = 625$.

We now fix α in (4.26) to the system order of the factored matrices, i.e the minimum value for which the ratio is equal to one when the global minimum of (4.17) has been reached. A global trend observed in Figure 4.5 is that the sparsity increases with the size of the array, for d constant. When $d = 2$ and for the sizes considered, the differences between BCU and (4.18) are minor. It is no longer the case when increasing d as observed for example when $d = 3$ in Figure 4.5. Even when $\mathbf{t} = \mathbf{1}$, the sparsity increases with the size of the array. The ambiguity parameter (the true one that we can reconstruct solving a least squares as we know $\mathbf{M}_{i,j}$) gets closer to one when the size of the array increases. It may be particular to the datasets and this method of assuming $\mathbf{t} = \mathbf{1}$ is not reliable for all cases.

Figure 4.6 plots the VAF on validation data for respectively $d = 3$ and $d = 5$ as a function of the size of the array. SEP-T4SID improves the mean and reduces the variance w.r.t the BCU-T4SID especially for large d. Improvements in accuracy are negligible when $d = 2$ and therefore not shown here. These observations are similar as when evaluating the vector of singular values.

Computational time

Figure 4.7 shows the evolution of the computational time as a function of the size of the array for the particular case $d = 3$. Especially it shows the improvement of the SEP-T4SID over BCU-T4SID for all sizes although the slope of the line is similar

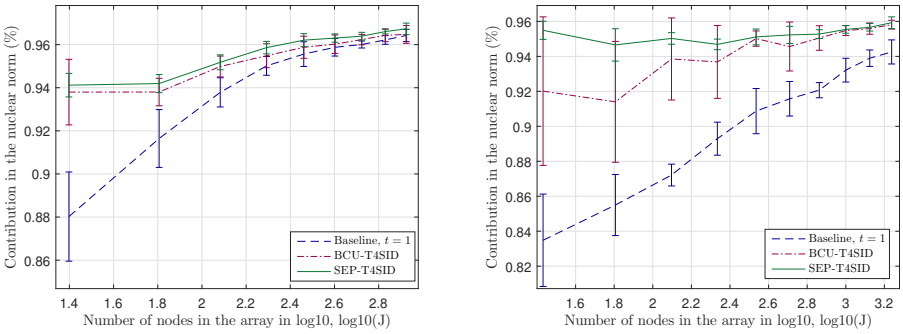


Figure 4.5: Ratio evaluating the sparsity of the singular values vector for the three methods: $\widehat{\mathbf{t}} = 1$, $\widehat{\mathbf{t}}$ obtained with BCU or (4.18). The y-axis corresponds to the contribution to the nuclear norm keeping only the first n_i singular values. The tensor order is equal to two (left) and three (right).

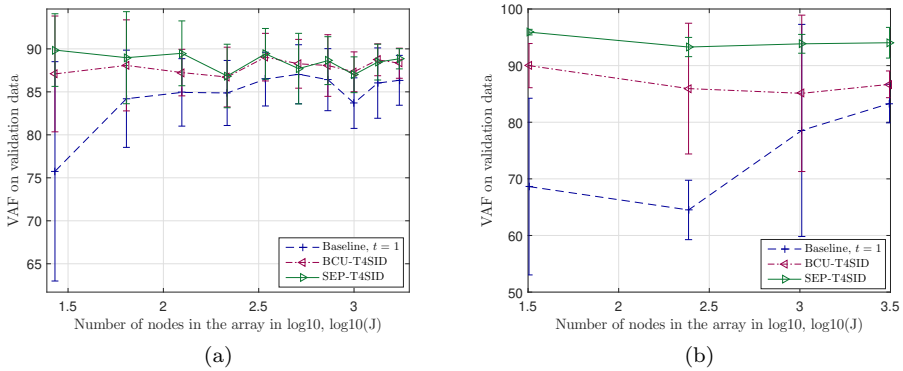


Figure 4.6: VAF on validation data as a function of the size of the array for SEP-T4SID and BCU-T4SID. Left: $d = 3$. Right: $d = 5$.

(as expected). We do not conclude directly from this figure that the QUARKS take more time as it strongly depends on the number of identification samples, but only mention it is however a general observation that it is the bottleneck for the equality (4.16) to hold.

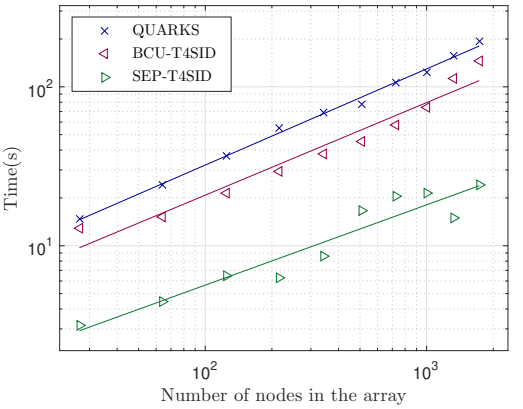


Figure 4.7: Evolution of the time for the QUARKS, the BCU-T4SID algorithm (QUARKS excluded), and the SEP-T4SID (QUARKS excluded) as a function of the size of the array for $d = 3$. The plot is in loglog scale.

We now compute the relative time improvement using SEP-T4SID and average over all sizes of array, for d constant. The values are summarized in Table 4.2. With increasing d, the relative improvement decreases although the accuracy of SEP-T4SID increases with respect to BCU-T4SID as observed in Figure 4.6.

Tensor order	$d = 2$	$d = 3$	$d = 4$
Mean	0.84	0.74	0.67
Standard deviation	0.14	0.07	0.13

Table 4.2: Relative improvement in computational time using SEP-T4SID over BCU-T4SID; both QUARKS excluded.

Figure 4.8 studies the scalability of both methods by showing the impact of increasing the tensor order d. When writing the complexity with aN^b for two scalars a, b, it is expected that a decreases when increasing d while b stays constant.
The trends in Table 4.3 are especially useful for extremely large sizes of sensor array. Tensor models should however not been used for small sizes of the sensor. It is moreover remarkable that the three lines seem to cross each other at the approximate same size of sensor. It is a particularity stemming from the choice of the length of the past window s and the number of identification samples. The computing installation that we use did not allow us to compute in reasonable time larger sizes of the array and therefore future improvements should go in the direction of decreasing the coefficient b of these models using e.g recursive methods.

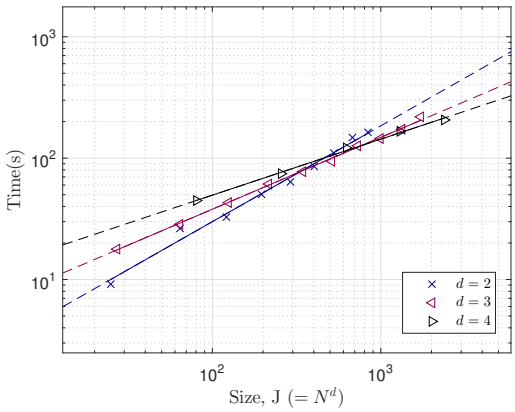


Figure 4.8: Total time of execution as a function of the total number of nodes in the array in loglog scale for the proposed algorithm exploiting the positivity constraint. The tensor order d ranges in the set $\{2, 3, 4\}$.

Tensor order	$d = 2$	$d = 3$	$d = 4$
Coef. a	0.79	0.59	0.46

Table 4.3: Coefficient a of linear model $log10(Time) = a \times log10(J) + b$ for different values of the tensor order d.

4.5. Conclusion

Conclusions

In this chapter, tensor state-space models have been introduced. Instead of the classical state-space model in vector form, we have introduced a multi-linear variant which recasts the input, state and output as tensors. The main weaknesses of K4SID as derived in Chapter 3 have been addressed by first, restricting to the class of Kronecker-structured systems whose factored Markov parameters are strictly positive element-wise, and the state-space matrices are now estimated from standard realization theory. The first variant introduces a different optimization strategy for estimating an admissible ambiguity sequence. The cost function includes the composition of the nuclear norm with the exponential component-wise function. Although theoretical properties have not been determined, numerical experiments have shown that this new formulation increases the accuracy of the estimates with respect to the Block-Coordinate Update. These tensor state-space models reveal to scale better for large sizes of the sensor array when increasing the order d.

Recommendations

There remains questions that we have not answered in this chapter and that are left open for future research. We highlight a few of them related to the assumption of strictly externally systems. More general recommendations related to the tensor state-space models are found in the concluding chapter.

When is it possible to model stochastic processes with internally or externally positive matrices?

What are the properties of the function that maps $\mathbf{v}_i$ *to* $\|\mathcal{H}_i(\{e^{v_{i,j}}\}_{j\in\{1,\dots,s\}})\|_\star$*?*

How to estimate the factored state-space matrices from a block-Hankel matrix when these are assumed strictly positive element-wise?
Scalable control methods have been derived in Rantzer (2011) for the class of internally positive systems although system identification is lagging behind. If the factors of a tensor state-space model are all strictly positive element-wise, then the state-space model in its vectorized form is internally positive. One main difficulty that arises concerns the realization from a block-Hankel of state-space matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$ with positive entries. Standard realizations do not guarantee that all three matrices have positive entries, simultaneously. Yu et al. (2018b) develop a gray-box identification method which allows an easy integration of such constraints. The drawback is that it is solved using a Difference-of-Convex programming algorithm which requires good initial guesses.

5
Solving Kronecker-structured discrete Lyapunov equations

We solve the discrete Lyapunov equation when the system matrices are low-Kronecker rank thereby reducing the computational cost with respect to the unstructured solution.

For an array with $N \times N$ nodes and such that the system order scales with N^2, standard methods used for solving this equation are no longer tractable. Two common methods for medium-size matrices consist of the Bartels-Steward and Schur-Hessenberg algorithm. Iterative algorithms are an alternative especially interesting for structured large matrices. The sign function was e.g used in Rice (2010) to carry out structure-preserving operations when the matrices are SSS. For discrete-time equations and low Kronecker rank matrices, we investigate how standard algorithms from the literature may efficiently exploit the structure.

As main contributions of this chapter, we adapt the squared Smith iterations to the class of Kronecker-structured matrices, and highlight the equivalence with a discrete Sylvester equation having a low-rank coefficient matrix which has received a large interest in the literature. From this observation, we adapt a factored Alternating Direction Implicit method for solving this equation when the state-transition matrix has Kronecker rank one. Both algorithms reduce the complexity from $\mathcal{O}(N^6)$ to $\mathcal{O}(N^3)$ for $d = 2$.

This chapter is published for the first time in this dissertation. The authors are grateful to Prof. Benner and Dr. Kürschner for providing a Matlab implementation of the algorithm in Benner and Kürschner (2014).

5.1. Introduction

When the state-space matrices of a stochastic LTI system are known, the minimum variance unbiased estimator of the predicted state requires the knowledge of the Kalman gain and essentially, to solve the Discrete Algebratic Riccati Equation (DARE). As a first step toward solving the DARE when the matrices are low-Kronecker rank, we propose a computationnaly efficient iterative algorithm for solving a Kronecker-structured discrete Lyapunov equation given as follows:

$$\mathbf{P} = \mathbf{A}\mathbf{P}\mathbf{A}^T + \mathbf{Q} \tag{5.1}$$

The latter is a critical step for the Newton's and Newton-Hewer's algorithm as highlighted in Benner and Faßbender (2011). Standard routines are the Bartels-Stewart algorithm, Bartels and Stewart (1972), and the Schur-Hessenberg method, Golub et al. (1978). Both scale with $\mathcal{O}(n^6)$ when $\mathbf{A} \in \mathbb{R}^{n^2 \times n^2}$ (assuming n in the same order of magnitude as N). Alternatives have been proposed for solving a large-scale discrete Lyapunov equation.

The first one relies on the matrix sign function, Gardiner and Laub (1985). The sign iteration was introduced in Roberts (1980) and is computed sequentially, each iteration involving a matrix inverse. It converges quadratically although it suffers from numerical problems as soon as an eigenvalue comes too close to the imaginary axis during the iterations. Although it adresses especially the continuous-time Lyapunov equation, it adapts to discrete-time systems when first transforming the latter into their equivalent continuous representation using the bilinear transform $\mathbf{C} = (\mathbf{D} + \mathbf{I})^{-1}(\mathbf{D} - \mathbf{I})$. The spectrum of $\mathbf{D}$ lies strictly within the unit circle if and only if the spectrum of $\mathbf{C}$ lies on the left-half plane excluding the imaginary axis. The bottleneck as it appears in general for structured-preserving operations is that inverses are required, e.g for the bilinear transform and when computing the matrix sign, implying more difficulties to maintain the structure in the solution without losing much accuracy. Rice (2010) solves the continuous-time Lyapunov equation using the sign iteration when the state-space matrices are SSS. Its main asset is the property that the inverse of a SSS matrix is SSS as was discussed in Chapter 1.

The second alternative is the squared Smith's iteration, Smith (1968). This doubling algorithm only involves matrix-matrix multiplication and squaring the transition matrix $\mathbf{A}$ at each iteration. When the matrix $\mathbf{A}$ is stable, the convergence is quadratic. The Kronecker rank of $\mathbf{A}$ increases when squaring unless it is equal to one from the first iteration. Errors due to truncation while keeping the Kronecker rank (or system order in the SSS representation) low may accumulate throughout the iterations at the expense of convergence, and then accuracy. It highlights the need for truncation algorithms to maintain the Kronecker rank within acceptable bounds and ensure computationally efficient operations.

While solving the DARE and when the system matrices are banded as obtained e.g from discretizing a PDE, sparsity is however gradually destroyed throughout the Smith's iterations and the low-rank structure of the solution is preferably exploited, Benner and Faßbender (2011). The third main class of algorithms targets especially the cases where $\mathbf{A}$ is large and sparse, and $\mathbf{Q}$ low-rank. The latter condition is shown to result in low-rank and not necessarily sparse solutions $\mathbf{P}$. Sabino (2006) derives

bounds on the decay of the singular values of $\mathbf{P}$ in the case that the coefficient matrix $\mathbf{A}$ is stable and real symmetric. The smaller the condition number of $\mathbf{A}$, the quicker the decay of the singular values of $\mathbf{P}$. There is however a lack of understanding for non-symmetric coefficient matrices. Nonetheless, this low-rank property was exploited to derive efficient algorithms which do not require to form the full solution matrix. Factored Alternating Direction Implicit method (fADI) have been proposed in Penzl (1999), Benner et al. (2008) and Benner and Kürschner (2014).

When vectorizing (5.1), the coefficient-matrix is a particular sum of Kronecker products. Demko et al. (2010) and Canuto et al. (2014) show that the inverse of symmetric, positive definite and banded matrices with this particular Kronecker structure are off-diagonally decaying matrices. Haber and Verhaegen (2016) exploit the decay in the entries of the solution to derive an algorithm with linear computational complexity with respect to the number of states. Major advantages of these assumptions are that first, theoretical decay rates have been derived to characterize the sparsity of the inverse in Canuto et al. (2014), second, that the relation between sparsity of the solution and the condition number of $\mathbf{A}$ are well-understood, Haber and Verhaegen (2016), and third, that a sparse banded solution paves the way for a structured Kalman gain and efficient online computations.

In adaptive optics for example, the wavefront covariance matrix is not sparse although low-Kronecker rank. The methods mentioned in the previous paragraphs thus do not apply. This provides a motivation for deriving a structured solution of the discrete Lyapunov equation in a scalable manner.

The main contributions of this chapter are twofold. First, we solve the discrete Lyapunov equation while preserving the low-Kronecker rank structure throughout the squared Smith's iterations in order to ensure the targeted computational complexity, $\mathcal{O}(n^3)$. As a building block, we derive an algorithm for truncating a matrix in $\mathcal{K}_{d,r_Y}$ by a matrix in $\mathcal{K}_{d,r_X}$ for r_X strictly smaller than r_Y. Theoretical conditions on when the discrete Lyapunov equation admit a low-Kronecker rank structure have however not been derived although it is shown to be equivalent to a widely studied and on-going research topic, that is, if the matrix $\mathbf{Q}$ appearing in the discrete Sylvester equation is low-rank, is the solution also low-rank. We write a factored ADI method adapted to the Kronecker structure when the matrix $\mathbf{A}$ has Kronecker rank one.

The chapter is organized as follows. Section 5.2 formulates the problem and introduces the Smith's iterations. In Section 5.3.1 we investigate linear algebra operations for low-Kronecker rank matrices. Section 5.3.4 adapts the doubling algorithm while preserving the structure and mentions the pitfalls. Section 5.4 presents an alternative which consists of rewriting the discrete Lyapunov into a large though structured discrete Sylvester equation with low-rank matrices. Section 5.5 discusses numerical experiments for randomly generated LTI systems.
Notations. $\bar{x}$ is the complex conjugated of the complex scalar x and $\mathbf{X}^H = \bar{\mathbf{X}}^T$ is the complex conjugate transpose of $\mathbf{X}$. The determinant of the square matrix $\mathbf{A}$ is denoted with $det(\mathbf{A})$. For two square matrices $\mathbf{A}, \mathbf{E}$, the generalized eigenvalues $\Lambda(\mathbf{A}, \mathbf{E})$ are equal to the set $\{\lambda \in \mathbb{C} : det(\mathbf{A} - \lambda\mathbf{E}) = 0\}$.

5.2. Problem formulation

Let $d \in \mathbb{N}, (n_1, \ldots, n_d) \in \mathbb{N}^d$ and $n = \prod_{i=1}^{d} n_i$. Let $\underline{n} = \min(\{n_i\}_{i=1..d})$. If n_i is independent of i, we denote the one-dimensional size of the sensor array with $\bar{n} = n^{1/d}$. For $\mathbf{x}(k) \in \mathbb{R}^n$, let the state equation for a stochastic LTI model be written as:

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{w}(k) \tag{5.2}$$

The assumptions are the following. The matrix $\mathbf{A}$ is strictly stable. The process noise $\mathbf{w}(k)$ is zero mean white Gaussian and with semi-positive definite covariance matrix $\mathbf{Q}$. Let r a positive integer such that $r \ll \underline{n}$. We assume that both $\mathbf{A}$ and $\mathbf{Q}$ belong to $\mathcal{K}_{d,r}$.

Whether the solution of the discrete Lyapunov equation admits a low-Kronecker rank decomposition depends on whether the inverse of a particular matrix may be well approximated within this class. Writing $\mathbf{A} = \sum_{i=1}^{r} \mathbf{A}_{d,i} \otimes \ldots \otimes \mathbf{A}_{1,i}$ and $\mathbf{Q} = \sum_{i=1}^{r} \mathbf{Q}_{d,i} \otimes \ldots \otimes \mathbf{Q}_{1,i}$, and vectorizing (5.1), we have:

$$\underbrace{(\mathbf{I} - \mathbf{A} \otimes \mathbf{A})}_{\mathbf{M}} \mathrm{vec}(\mathbf{P}) = \mathrm{vec}(\mathbf{Q}) \tag{5.3}$$

The matrix $\mathbf{M}$ is a low-Kronecker rank matrix. We denote its factors with $\mathbf{M}_{j,i}$ for $(i,j) \in \{1, \ldots, r+1\} \times \{1, \ldots, 2d\}$. Let us denote the inverse of $\mathbf{M}$ (assuming it exists) with $\mathbf{N}$ and decompose it with a sum of Kronecker products as well. The solution is then given by:

$$\mathrm{vec}(\mathbf{P}) = \big(\sum_{i=1}^{r_N} \mathbf{N}_{2d,i} \otimes \ldots \otimes \mathbf{N}_{1,i}\big)\mathrm{vec}(\mathbf{Q}) \tag{5.4}$$

which is also rewritten using the ivec operator as:

$$\mathbf{P} = \sum_{i=1}^{r_N} \sum_{j=1}^{r} \mathbf{N}_{d,i}\mathbf{Q}_{d,j}\mathbf{N}_{2d,i}^T \otimes \ldots \otimes \mathbf{N}_{1,i}\mathbf{Q}_{1,j}\mathbf{N}_{d+1,i}^T \tag{5.5}$$

The Kronecker rank of $\mathbf{P}$ is equal to $\min(r_N r, n^4)$. The lower r_N, the better. We refer to Varnai (2017) for a first discussion on whether inverses of low-Kronecker rank matrices with random factors admit a decomposition with a sum of few terms. In this chapter, we first derive a structure-preserving algorithm and discuss sufficient properties on the factor matrices for the discrete Lyapunov equation to be approximated within this class. The problem is now formulated.

From the known factor matrices of the Kronecker-structured $(\mathbf{A}, \mathbf{Q})$ *in the class* $\mathcal{K}_{d,r}$*, and assuming* $\mathbf{A}$ *is stable and* $\mathbf{Q}$ *is symmetric positive definite, determine the state covariance matrix* $\widehat{\mathbf{P}}$ *in* $\mathcal{K}_{d,r}$ *solving the discrete Lyapunov equation with* $\mathcal{O}(\bar{n}^3)$ *if* $d = 2$ *and* $\mathcal{O}(\bar{n}^{2(d-1)})$ *for* $d \geq 3$ *such that the relative error between the unstructured solution and* $\widehat{\mathbf{P}}$ *is small.*

5.3. The squared Smith's method

We first discuss existence and uniqueness of the solution of the discrete Lyapunov equation.

Theorem 5.1. *Let* $\mathbf{A}, \mathbf{Q} \in \mathbb{R}^{n \times n}$ *matrices and such that* $\mathbf{A}$ *stable and* $\mathbf{Q}$ *is symmetric positive definite. The discrete Lyapunov equation*

$$\mathbf{A}\mathbf{P}\mathbf{A}^T - \mathbf{P} = -\mathbf{Q} \tag{5.6}$$

has a unique solution when $\lambda\mu \neq 1$ *for every pair of eigenvalues* λ, μ *of* $\mathbf{A}$*. The solution is then given from the convergent serie:*

$$\mathbf{P} = \sum_{i=0}^{\infty} \mathbf{A}^i \mathbf{Q} \mathbf{A}^{iT} \tag{5.7}$$

Smith (1968) first truncates the infinite sum and derives the iterations:

$$\mathbf{P}^{(0)} = \mathbf{0}, \qquad \mathbf{P}^{(\kappa+1)} = \mathbf{Q} + \mathbf{A}\mathbf{P}^{(\kappa)}\mathbf{A}^T \tag{5.8}$$

where κ is the iteration counter. He then proposes a doubling algorithm to speed up the convergence. Instead of computing all consecutive updates of (5.8), the distance in terms of κ between two consecutive updates doubles at each iteration. Starting from $\mathbf{U}^{(0)} = \mathbf{A}$ and $\mathbf{P}^{(0)} = \mathbf{Q}$, the iterations are as follows:

$$\mathbf{P}^{(\kappa+1)} := \mathbf{P}^{(\kappa)} + \mathbf{U}^{(\kappa)}\mathbf{P}^{(\kappa)}{\mathbf{U}^{(\kappa)}}^T, \qquad \mathbf{U}^{(\kappa+1)} := {\mathbf{U}^{(\kappa)}}^2 \tag{5.9}$$

Or, to illustrate its quadratic convergence rate:

$$\mathbf{P}^{(\kappa)} = \sum_{i=0}^{2^\kappa} \mathbf{A}^i \mathbf{Q} \mathbf{A}^{iT} \tag{5.10}$$

Solving the discrete Lyapunov equation boils down to computing iteratively matrix-matrix multiplications.

5.3.1. Structure-preserving operations

Assuming that the system matrices are in $\mathcal{K}_{d,r}$, simply plugging them into (5.9) would not improve the computational complexity unless the structure is exploited and most importantly, maintained, at each iteration allowing to rewrite the routines in matrix (or tensor) form. Linear algebra operations such as adding, multiplying and transposing low-Kronecker rank matrices should therefore not destroy the structure. We first review a few results on the computational complexity of elementary operations when the matrix belongs to $\mathcal{K}_{d,r}$ before deriving algorithms for approximating a matrix in $\mathcal{K}_{d,r_Y}$ by a matrix in $\mathcal{K}_{d,r_X}$ with r_X smaller than r_Y.

5.3.2. Adding, multiplying and transposing

Lemma 5.1. Let $(\mathbf{X}, \mathbf{Y}) \in \mathcal{K}_{d,r_X} \times \mathcal{K}_{d,r_Y}$ and with factors matrices of same size. The following properties hold:

- $\mathbf{X} + \mathbf{Y} \in \mathcal{K}_{d,r_X+r_Y}$
- $\mathbf{X}\mathbf{Y} \in \mathcal{K}_{d,r_X r_Y}$

- $\forall i \in \mathbb{N}, \mathbf{X}^i \in \mathcal{K}_{d,r_X^i}$
- $\mathbf{X}^T \in \mathcal{K}_{d,r_X}$

Proof.

- The result readily follows from expressing $\mathbf{X}+\mathbf{Y}$ in Kronecker format with:

$$\begin{aligned}\mathbf{X}+\mathbf{Y} &= \sum_{i=1}^{r_X} \mathbf{X}_{d,i} \otimes \ldots \otimes \mathbf{X}_{1,i} + \sum_{i=1}^{r_Y} \mathbf{Y}_{d,i} \otimes \ldots \otimes \mathbf{Y}_{1,i} \\ &= \sum_{i=1}^{r_X+r_Y} \mathbf{Z}_{d,i} \otimes \ldots \otimes \mathbf{Z}_{1,i}\end{aligned}$$

  where, for all $j \in \{1,\ldots,d\}, \mathbf{Z}_{j,i}$ is equal to $\mathbf{X}_{j,i}$ if $i \leq r_X$ and $\mathbf{Y}_{j,i}$ otherwise.

- The product $\mathbf{XY}$ reads:

$$\begin{aligned}\mathbf{XY} &= \sum_{i=1}^{r_X} \mathbf{X}_{d,i} \otimes \ldots \otimes \mathbf{X}_{1,i} \sum_{j=1}^{r_Y} \mathbf{Y}_{d,j} \otimes \ldots \otimes \mathbf{Y}_{1,j} \\ &= \sum_{i=1}^{r_X}\sum_{j=1}^{r_Y} \mathbf{X}_{d,i}\mathbf{Y}_{d,j} \otimes \ldots \otimes \mathbf{X}_{1,i}\mathbf{Y}_{1,j} \qquad (5.11)\end{aligned}$$

  There is a maximum of $r_X r_Y$ terms in the sum.

- Let $i \in \mathbb{N}$. From the previous point, we infer $\mathbf{X}^2 \in \mathcal{K}_{d,r_X^2}$. A reasoning by induction gives $\mathbf{X}^i \in \mathcal{K}_{d,r_X^i}$.

- For $\mathbf{X} \in \mathcal{K}_{d,r_X}$, we have $\mathbf{X}^T = \sum_{i=1}^{r_X} \mathbf{X}_{d,i}^T \otimes \ldots \otimes \mathbf{X}_{1,i}^T$ using the linearity of the transpose operator and the property that it applies to each factor matrix on each single Kronecker product.

■

These operations are denoted in the sequel with `SOK_add`, `SOK_multiply`, `SOK_transpose`, (SOK standing for Sums-Of-Kronecker).

5.3.3. Truncating the Kronecker rank of matrices

Adding and multiplying low-Kronecker rank matrices increase the Kronecker rank. In this paragraph, we approximate a matrix $\mathbf{Y}$ belonging to $\mathcal{K}_{d,r_Y}$ by a matrix $\mathbf{X}$ in $\mathcal{K}_{d,r_X}$ such that r_X is smaller than r_Y. This operation is necessary for iterative algorithms to ensure that the complexity does not explode throughout the iterations.

There are two options depending on whether the factor matrices of $\mathbf{Y}$ are known. If the factor matrices are not known, then estimating the factor matrices in $\mathbf{X}$ from the tensorised $\mathbf{Y}$ is solved in general with a CPD. If only the factor matrices

are known (which is the case we are interested in as we never form the large matrices for storage reasons), we aim at estimating $\mathbf{X}$ such that:

$$\min_{\{\mathbf{X}_{i,j}\}_{i=1..d;j=1..r_X}} \quad \|\sum_{j=1}^{r_Y} \mathbf{Y}_{d,j} \otimes \ldots \otimes \mathbf{Y}_{1,j} - \sum_{j=1}^{r_X} \mathbf{X}_{d,j} \otimes \ldots \otimes \mathbf{X}_{1,j}\|_F^2 \tag{5.12}$$

where $\mathbf{X}_{i,j}, \mathbf{Y}_{i,j} \in \mathbb{R}^{n_i \times n_i}$. The minimization problem (5.12) is multi-convex. For m in the set $\{1, \ldots, d\}$, fixing all variables but $\{\mathbf{X}_{m,j}\}_{j=1..r_X}$ yields a convex problem. Similarly to Chapter 2 and 4 where we have dealt with autoregressive models, we propose an Alternating Least-Squares algorithm to estimate the factor matrices.

Remark 5.1. *Gauss-Newton algorithms may be proposed as well to benefit from the quadratic convergence rate instead of linear for ALS, Vervliet and De Lathauwer (2018). Key steps for deriving efficient updates is to exploit structure in the Hessian and solve the associated linear equation with e.g conjugate gradient such that no inversion is required. Such derivations are not presented here.*

To exhibit the factor matrices $\mathbf{X}_{m,j}$ and formulate a tractable optimization problem, we transform (5.12) reshuffling the matrix along the m-th mode. In addition, we add a regularization on the variable to minimize the Frobenius norm of $\mathbf{X}_{m,j}$ to avoid numerical issues due to terms diverging while the cost function still decreases (this case is possible because of the ambiguity transformation inherent to the multi-linear parametrization of $\mathbf{X}$). The optimization (5.12) is rewritten,

$$\min_{\mathbf{U}_{X,m}} \quad \|\mathbf{U}_{Y,m}\mathbf{V}_{Y,m}^T - \mathbf{U}_{X,m}\mathbf{V}_{X,m}^T\|_F^2 + \lambda\|\mathbf{U}_{X,m}\|_F^2 \tag{5.13}$$

where λ is a weighting parameter, and:

$$\begin{aligned}
\mathbf{U}_{Y,m} &= \begin{bmatrix}\text{vec}(\mathbf{Y}_{m,1}) & \ldots & \text{vec}(\mathbf{Y}_{m,r_Y})\end{bmatrix} \in \mathbb{R}^{n_m^2 \times r_Y} \\
\mathbf{V}_{Y,m} &= \begin{bmatrix}\text{vec}(\tilde{\mathbf{Y}}_{m,1}) & \ldots & \text{vec}(\tilde{\mathbf{Y}}_{m,r_Y})\end{bmatrix} \in \mathbb{R}^{\prod_{i=1,i\neq n}^{d} n_i^2 \times r_Y} \\
\tilde{\mathbf{Y}}_{m,j} &= \mathbf{Y}_{d,j} \otimes \ldots \otimes \mathbf{Y}_{m+1,j} \otimes \mathbf{Y}_{m-1,j} \otimes \ldots \otimes \mathbf{Y}_{1,j}
\end{aligned}$$

The terms $\mathbf{U}_{X,n}$ and $\mathbf{V}_{X,n}$ are defined similarly from the factors $\mathbf{X}_{i,j}$. To stop the iterations, we need to evaluate the cost function (5.12). The latter is evaluated with $(r_Y^2 + r_X r_Y + r_X^2)dn^{3/d}$ operations as follows.

$$\begin{aligned}
\|\mathbf{Y} - \mathbf{X}\|_F^2 = &\sum_{j_1=1}^{r_Y}\sum_{j_2=1}^{r_Y}\prod_{\ell=1}^{d}\text{Trace}(\mathbf{Y}_{\ell,j_1}^T\mathbf{Y}_{\ell,j_2}) - 2\sum_{j=1}^{r_Y}\sum_{i=1}^{r_X}\prod_{\ell=1}^{d}\text{Trace}(\mathbf{Y}_{\ell,j}^T\mathbf{X}_{\ell,i}) \\
&+ \sum_{i_1=1}^{r_X}\sum_{i_2=1}^{r_X}\prod_{\ell=1}^{d}\text{Trace}(\mathbf{X}_{\ell,i_1}^T\mathbf{X}_{\ell,i_2})
\end{aligned} \tag{5.14}$$

For simplicity, assume that the dimensions of the factor matrices are equal. Denote the value of the cost function at iteration κ with $c^{(\kappa)}$. Once $|c^{(\kappa)} - c^{(\kappa-1)}| < \epsilon_{max}$ for some given threshold ϵ_{max}, we stop iterating.

The algorithm entitled `SOK_truncate` is summarized in Algorithm 5.1. No guarantee for achieving a global minimum of the approximation error is provided. As discussed in the musings, Mohlenkamp (2013), the convergence to a global minimum of ALS is not well understood in general. When $d = 2$, the algorithm is very similar to a PCA problem and a proof of convergence to a set of stationary points is given in Udell et al. (2016).

Algorithm 5.1: `SOK_truncate(Y, r_X, lambda)`

Input : $\{\mathbf{Y}_{i,j}\}_{i=1..d,j=1..r_Y}, r_X, \lambda$
Output: $\{\mathbf{X}_{i,j}\}_{i=1..d,j=1..r_X}$

`/* Default values */`
$\kappa_{max} = 30, \epsilon_{max} = 10^{-3}$
`/* Initial guesses */`
$\kappa \leftarrow 0$
foreach $i \leq d$ **do**
  foreach $j \leq r$ **do**
    $\mathbf{X}_{i,j}^{(\kappa)} \leftarrow$ `rand(n_i, n_i)`
  end
end
`/* ALS iterations */`
while $\kappa \leq \kappa_{max}$ and $\epsilon > \epsilon_{max}$ **do**
  foreach $n = 1..d$ **do**
    Form $\mathbf{V}_{X,n}^{(\kappa)}$ from $\{\mathbf{X}_{i,j}^{(\kappa+1)}\}_{i=1..n-1,j=1..r_X}, \{\mathbf{X}_{i,j}^{(\kappa)}\}_{i=n+1..d,j=1..r_X}$
    Compute $\mathbf{V}_{Y,n}^T \mathbf{V}_{X,n}^{(\kappa)}$ and denote with $\mathbf{c}$
    Compute $\mathbf{U}_{Y,n}\mathbf{c}$ and denote with $\mathbf{d}$
    Compute ${\mathbf{V}_{X,n}^{(\kappa)}}^T \mathbf{V}_{X,n}^{(\kappa)} + \lambda\mathbf{I}$ and invert, denote with $\mathbf{R}$
    Set $\mathbf{U}_X^{(\kappa+1)}$ to $\mathbf{dR}$
    Update $\{\mathbf{X}_{i,j}^{(\kappa+1)}\}_{j=1..r_X}$ from $\mathbf{U}_X^{(\kappa+1)}$
  end
  `/* Check stopping criterion */`
  Evaluate the residual using (5.14) and denote with $c^{(\kappa)}$
  $\epsilon \leftarrow |c^{(\kappa)} - c^{(\kappa-1)}|$
  $\kappa \leftarrow \kappa + 1$
end
$\mathbf{X}_{i,j} \leftarrow \mathbf{X}_{i,j}^{(\kappa-1)}$

Computational complexity

Storing $\mathbf{U}_{Y,m}, \mathbf{V}_{Y,m}$ for all $m \in \{1, \ldots, d\}$ scales with $rd(n^{2\frac{d-1}{d}} + n^{2/d})$. If all variables are fixed but $\mathbf{U}_{X,m} \in \mathbb{R}^{n_m^2 \times r_X}$, the optimization (5.13) is a standard regularized least-squares whose complexity is dominated by the cost for computing

products such as $\mathbf{V}_{X,m}^T\mathbf{V}_{X,m}$, i.e:

$$\mathbf{V}_{X,m}^T\mathbf{V}_{X,m} = \begin{bmatrix} \mathrm{vec}(\tilde{\mathbf{X}}_{m,1})^T\mathrm{vec}(\tilde{\mathbf{X}}_{m,1}) & \dots & \mathrm{vec}(\tilde{\mathbf{X}}_{m,1})^T\mathrm{vec}(\tilde{\mathbf{X}}_{m,r_X}) \\ \vdots & \ddots & \vdots \\ \mathrm{vec}(\tilde{\mathbf{X}}_{m,r_X})^T\mathrm{vec}(\tilde{\mathbf{X}}_{m,1}) & \dots & \mathrm{vec}(\tilde{\mathbf{X}}_{m,r_X})^T\mathrm{vec}(\tilde{\mathbf{X}}_{m,r_X}) \end{bmatrix} \tag{5.15}$$

where, for $i, j \in \{1, \dots, r_X\}$:

$$\begin{aligned} \mathrm{vec}(\tilde{\mathbf{X}}_{m,i})^T\mathrm{vec}(\tilde{\mathbf{X}}_{m,j}) &= \mathbf{1}^T\mathrm{vec}\big(\mathbf{A}_{d,i,j} \otimes \dots \otimes \mathbf{A}_{m+1,i,j} \otimes \mathbf{A}_{m-1,i,j} \otimes \dots \otimes \mathbf{A}_{1,i,j}\big) \\ \mathbf{A}_{\ell,i,j} &= \mathbf{X}_{\ell,i} \circ_H \mathbf{X}_{\ell,j}, \quad \ell = 1..d \end{aligned} \tag{5.16}$$

where $\circ_H$ denotes the Hadamard product. The cost for computing $\mathbf{A}_{\ell,i,j}$ is $n^{2/d}$. Computing $\mathrm{vec}(\tilde{\mathbf{X}}_{m,i})^T\mathrm{vec}(\tilde{\mathbf{X}}_{m,j})$ requires $n^{2(d-1)/d}$ flops: forming this large matrix with Kronecker products is devastating for the computational efficiency especially as soon as d is larger than two. It is the bottleneck of this algorithm. This type of operation is repeated $(r_X + 1)/2$ times to form (5.15). Computing the inverse costs only r_X^3. Computing $\mathbf{V}_{Y,i}^T\mathbf{V}_{X,i}^{(\kappa)}$ costs $r_Y r_X n^{2(d-1)/d}$. As an illustration with the notations in an array of size $N \times N$ and $d = 2$, we have $n = N^2$ and the algorithm scales with $\mathcal{O}(N^2)$. For arrays of size $N \times N \times N$, the complexity reaches $\mathcal{O}(N^4)$. Asymptotically with d, the complexity is not linear with the number of nodes in the array but scales quadratically with the latter. It is a main difference to all previous algorithms studied in this thesis where increasing the tensor order decreases the scalability coefficient.

5.3.4. Pitfalls

The algorithm for solving the discrete Lyapunov equation via the squared Smith iterations is first summarized in Algorithm 5.2. When $\mathbf{U}^{(\kappa)}$ is stable, the sum of the norms (e.g Frobenius) of the factor matrices decays to 0, at least from a certain iteration, and is used as stopping criterion.

The difficulty in this algorithm is therefore not only to keep the Kronecker structure, but to ensure simultaneously that the truncated (global) $\mathbf{U}^{(\kappa)}$ is stable. The pitfalls relate essentially to the convergence of the doubling algorithm when the matrix $\mathbf{A}$ has eigenvalues close to the unit circle. Truncating while iterating may cause the eigenvalues of $\mathbf{U}^{(\kappa)}$ to jump outside the unit circle and consequently, cause the residual to diverge. In this case, the Kronecker rank should be increased. Computing in a scalable manner the eigenvalues of a Kronecker-structured matrix from its factors would allow to integrate a stopping criterion such that the algorithm is restarted with a larger Kronecker rank, r_P. Indeed, there is no characterization in terms of the factors (or their respective eigenvalues) such that a sum of Kronecker products is stable. For example with the case $d = 2$, let a matrix whose reshuffling is rank r strictly larger than one. It is not true in general that if the spectral radius of each factor matrix is strictly smaller than one, then the matrix written of a sum of r Kronecker terms has its spectral radius strictly smaller than one.

Another concern (minor compared to the previous one) here is related to the non-uniqueness of each factor due to the multi-linear representation, i.e. the trivial

Algorithm 5.2: `SOK_dlyap(A,Q,r_P)`

Input : $\{\mathbf{A}_{i,j}\}_{i=1..d,j=1..r_A}, \{\mathbf{Q}_{i,j}\}_{i=1..d,j=1..r_Q}$
Output: $\{\mathbf{P}_{i,j}\}_{i=1..d,j=1..r_P}$

```
   /* Default values                                                  */
 1 kappa_max = 30; epsilon_max = 1e-6; lambda = 1e-9
   /* Initialization                                                 */
 2 kappa = 0
 3 Ur = A
 4 Pr = Q
   /* Doubling algorithm                                             */
 5 while kappa < kappa_max-1 and epsilon > epsilon_max do
     /* P^(κ+1) := P^(κ) + U^(κ) P^(κ) U^(κ)^T                       */
 6   UP = SOK_multiply(Ur,Pr)
 7   UPr = SOK_truncate(UP,r,lambda)
 8   UT = SOK_transpose(U)
 9   UPU = SOK_multiply(UPr,UT)
10   UPUr = SOK_truncate(UPU,r,lambda)
11   P = SOK_add(P,UPUr)
12   Pr = SOK_truncate(P,r,lambda)
     /* U^(κ+1) := U^(κ)^2                                           */
13   U2 = SOK_multiply(U,U)
14   Ur = SOK_truncate(U2,r,lambda)
     /* Check stopping criterion                                     */
15   epsilon = SOK_getNormFactors(Ur)
16   kappa = kappa+1
17 end
```

indeterminacies which are scaling and permutation. For example, when $d = 2$, some factors might take very large values when the other counter-balance by getting closer to 0 which leads to numerical issues when squaring again in the next iteration. The regularization with μ introduced in (5.13) plays a key role in mitigating this effect. The accuracy of the truncation would suffer from setting μ large, though.

5.4. Using a factored Alternating Direction Implicit method

We now assume d is equal to two and the Kronecker rank of $\mathbf{A}$ equal to one. If not stated otherwise, the norm used is the spectral norm. The discrete Lyapunov equation (5.6) with a Kronecker rank one representation of $\mathbf{A}$ may be reshuffled applying the operator $\mathcal{R}$:

$$\mathcal{R}(\mathbf{A}\mathbf{P}\mathbf{A}^T - \mathbf{P} + \mathbf{Q}) = \mathbf{0} \tag{5.17}$$

We expand the product $\mathbf{A}\mathbf{P}\mathbf{A}^T$ and use the linearity of the reshuffling operator to write

$$\sum_{j=1}^{r} \mathcal{R}(\mathbf{A}_2\mathbf{P}_{2,j}\mathbf{A}_2^T \otimes \mathbf{A}_1\mathbf{P}_{1,j}\mathbf{A}_1^T) - \mathcal{R}(\mathbf{P}) + \mathcal{R}(\mathbf{Q}) = \mathbf{0} \tag{5.18}$$

Using the fact that the reshuffling of a Kronecker matrix is rank one,

$$\sum_{j=1}^{r} \mathrm{vec}(\mathbf{A}_2\mathbf{P}_{2,j}\mathbf{A}_2^T)\mathrm{vec}(\mathbf{A}_1\mathbf{P}_{1,j}\mathbf{A}_1^T)^T - \sum_{j=1}^{r} \mathrm{vec}(\mathbf{P}_{2,j})\mathrm{vec}(\mathbf{P}_{1,j})^T + \sum_{j=1}^{r_Q} \mathrm{vec}(\mathbf{Q}_{2,j})\mathrm{vec}(\mathbf{Q}_{1,j})^T = \mathbf{0} \tag{5.19}$$

and isolating the factors from $\mathbf{P}$ using the rule (A.10),

$$\underbrace{(\mathbf{A}_2 \otimes \mathbf{A}_2)}_{\tilde{\mathbf{A}}} \tilde{\mathbf{P}} \underbrace{(\mathbf{A}_1 \otimes \mathbf{A}_1)}_{\tilde{\mathbf{B}}}^T - \tilde{\mathbf{P}} + \tilde{\mathbf{Q}} = \mathbf{0} \tag{5.20}$$

where $\tilde{\mathbf{P}} = \sum_{j=1}^{r} \mathrm{vec}(\mathbf{P}_{2,j})\mathrm{vec}(\mathbf{P}_{1,j})^T$ and $\tilde{\mathbf{Q}} = \sum_{j=1}^{r_Q} \mathrm{vec}(\mathbf{Q}_{2,j})\mathrm{vec}(\mathbf{Q}_{1,j})^T$. The equation (5.20) is a discrete Sylvester equation.

Lemma 5.2. *Let $\lambda_{1,i}$ and $\lambda_{2,j}$ represent the eigenvalues of $\mathbf{A}_1$ and $\mathbf{A}_2$ respectively. The discrete Sylvester equation* (5.20) *has a unique solution for every $\tilde{\mathbf{Q}}$ if and only if $\lambda_{1,i}\lambda_{1,j}\lambda_{2,k}\lambda_{2,\ell} \neq 1$ for all $i, j \in \{1, \ldots, n_1\}, k, \ell \in \{1, \ldots, n_2\}$.*

Proof. The solution is unique if and only if both spectrum $\Lambda(\tilde{\mathbf{A}}, \mathbf{I}) = \{\lambda_{1,i}\lambda_{1,j}\}_{i,j=1..n_1}$ and $\Lambda(\mathbf{I}, \tilde{\mathbf{B}}) = \{\frac{1}{\lambda_{2,i}\lambda_{2,j}}\}_{i,j=1..n_2}$ are disjoint. The equation (5.20) has a unique solution if and only if $\lambda_{1,i}\lambda_{1,j}\lambda_{2,k}\lambda_{2,\ell} \neq 1$. ■

Solving a Kronecker-structured discrete Lyapunov equation is equivalent to solving a discrete Sylvester equation with low-rank right-hand side and Kronecker matrices. This connection is especially helpful because the case when the right-hand side matrix is low-rank has been already throughly investigated in the literature. Solving large-scale discrete Sylvester equation was made amenable when the coefficient matrices are sparse and the matrix $\tilde{\mathbf{Q}}$ is low-rank, say $r \ll n$. The latter condition is

shown to result in low-rank solutions. Sabino (2006) derives bounds on the decay of the singular values of $\tilde{\mathbf{P}}$ in the case that the coefficient matrices $\tilde{\mathbf{A}}, \tilde{\mathbf{B}}$ are stable and real symmetric. The low-rank structure has been exploited in Benner et al. (2009) and Benner and Kürschner (2014) and have relied on previous work carried out for solving Lyapunov equations with similar properties, see e.g Penzl (1999) and Benner et al. (2008).

The standard fADI algorithm iteratively forms factor matrices $\mathbf{Z}_k, \mathbf{Y}_k, \mathbf{D}_k$ of the low-rank solution $\tilde{\mathbf{P}}$. At each iteration, a different pair α_k, β_k is used and r new columns are added to the low-rank factor. The shifts parameters α_k and β_k are chosen e.g following Penzl (1999) to maximize the convergence speed. These parameters are complex and Benner and Kürschner (2014) adapt Algorithm 5.3 in order to carry out the calculations in $\mathbb{R}$ rather than in $\mathbb{C}$, which alleviates the computations and storage.

Algorithm 5.3: Factored ADI algorithm solving $\tilde{\mathbf{A}}\tilde{\mathbf{P}}\tilde{\mathbf{B}} - \tilde{\mathbf{P}} = \mathbf{F}\mathbf{G}^T$

Input : $\tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \mathbf{F}, \mathbf{G}$ and shift parameters $\{\alpha_1, \ldots, \alpha_{k_{max}}\}, \{\beta_1, \ldots, \beta_{k_{max}}\}$, tolerance $0 < \tau \leq 1$.

Output : $\mathbf{Z}_{k_{max}} \in \mathbb{C}^{n^2 \times rk_{max}}, \mathbf{Y}_{k_{max}} \in \mathbb{C}^{n^2 \times rk_{max}}, \mathbf{D}_{k_{max}} \in \mathbb{C}^{rk_{max} \times rk_{max}}$ such that $\mathbf{Z}_{k_{max}}\mathbf{D}_{k_{max}}\mathbf{Y}_{k_{max}}^H \approx \tilde{\mathbf{P}}$

1 $\widehat{\mathbf{V}}_0 = \mathbf{F}, \widehat{\mathbf{W}}_0 = \mathbf{G}, k = 1$

2 **while** $\widehat{\mathbf{V}}_{k-1}\widehat{\mathbf{W}}_{k-1}^H \geq \tau \|\mathbf{F}\mathbf{G}^T\|$ **do**

3 $\quad \mathbf{V}_k = (\tilde{\mathbf{A}} - \beta_k \mathbf{I}_{n^2})^{-1}\widehat{\mathbf{V}}_{k-1}, \mathbf{W}_k = (\mathbf{I}_{n^2} - \alpha_k \tilde{\mathbf{B}})^{-H}\widehat{\mathbf{W}}_{k-1}$

4 $\quad \widehat{\mathbf{V}}_{k-1} = \widehat{\mathbf{V}}_{k-1} + \gamma_k \mathbf{V}_k, \widehat{\mathbf{W}}_{k-1} = \widehat{\mathbf{W}}_{k-1} - \bar{\gamma_k}\tilde{\mathbf{B}}^T\mathbf{W}_k, \gamma_k = \beta_k - \alpha_k$

5 $\quad$ Update the low rank solution factors

$\quad \mathbf{Z}_k = \begin{bmatrix}\mathbf{Z}_{k-1} & \mathbf{V}_k\end{bmatrix}, \mathbf{Y}_k = \begin{bmatrix}\mathbf{Y}_{k-1} & \mathbf{W}_k\end{bmatrix}, \mathbf{D}_k = diag(\mathbf{D}_{k-1}, \gamma_k \mathbf{I}_r)$

6 $\quad k = k + 1$

7 **end**

In this work, we have used Algorithm 2 in Benner and Kürschner (2014) adapting the selection of the shift parameters, the solution of the linear system of equations, and the matrix-vector multiplications (e.g $\tilde{\mathbf{B}}^T\mathbf{W}_k$) which should also exploit the Kronecker structure using the ivec operator. The algorithm is not repeated here to highlight the novelty of our contribution based on Algorithm 5.3.

The computational bottleneck is in solving the linear system of equations in line 3 of Algorithm 5.3. When the coefficient-matrices are sparse, it was overcome using iterative optimization such as Krylov subspaces or e.g conjugate gradient. Rather than using iterative solvers when the matrices are Kronecker-structured, we observe that these linear systems of equations are nothing more than discrete Sylvester equations with smaller coefficient matrices:

$$\begin{cases} \mathbf{A}_1 \text{ivec}(\mathbf{V}_k)\mathbf{A}_1^T - \beta_k \text{ivec}(\mathbf{V}_k) & = \text{ivec}(\widehat{\mathbf{V}}_{k-1}) \\ \text{ivec}(\mathbf{W}_k) - \bar{\alpha}_k \mathbf{A}_2^T \text{ivec}(\mathbf{W}_k)\mathbf{A}_2 & = \text{ivec}(\widehat{\mathbf{W}}_{k-1}) \end{cases} \tag{5.21}$$

It follows:

$$\begin{cases} \frac{1}{\beta_k}\mathbf{A}_1 \text{ivec}(\mathbf{V}_k)\mathbf{A}_1^T - \text{ivec}(\mathbf{V}_k) & = \frac{1}{\beta_k}\text{ivec}(\widehat{\mathbf{V}}_{k-1}) \\ \bar{\alpha}_k \mathbf{A}_2^T \text{ivec}(\mathbf{W}_k)\mathbf{A}_2 - \text{ivec}(\mathbf{W}_k) & = -\text{ivec}(\widehat{\mathbf{W}}_{k-1}) \end{cases} \tag{5.22}$$

Both discrete Sylvester equations can be solved at each iteration using standard methods such as Bartels-Stewart algorithm or the Hessenberg-Schur method.

As for the estimation of optimal shift parameters, Penzl (1999) derives a minimax problem:

$$\min_{\substack{\alpha_j \in \mathbb{C} \\ \beta_j \in \mathbb{C}}} \max_{\substack{\lambda \in \Lambda(\tilde{\mathbf{A}}, \mathbf{I}) \\ \mu \in \Lambda(\mathbf{I}, \tilde{\mathbf{B}})}} \prod_{j=1}^{k} \left| \frac{(\lambda - \alpha_j)(\mu - \beta_j)}{(\lambda - \beta_j)(\mu - \alpha_j)} \right| \tag{5.23}$$

We follow these guidelines later adapted in Benner et al. (2009) for Sylvester equations and restrict α_j, β_j to $\Lambda(\tilde{\mathbf{A}}, \mathbf{I})$ and $\Lambda(\mathbf{I}, \tilde{\mathbf{B}})$ as commonly done. The shifts are here selected from the elements in $\Lambda(\tilde{\mathbf{A}}, \mathbf{I})$ and $\Lambda(\mathbf{I}, \tilde{\mathbf{B}})$. The good news when the matrices $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{B}}$ are Kronecker rank one is that the spectra in (5.23) are cheaply evaluated. The elements in $\Lambda(\tilde{\mathbf{A}}, \mathbf{I})$ are all $\lambda_{1,i}\lambda_{1,j}$ for $i, j = 1..n$. The elements in $\Lambda(\mathbf{I}, \tilde{\mathbf{B}})$ are all $1/(\lambda_{2,i}\lambda_{2,j})$. It is preferable for numerical reasons to avoid computing inverses of the eigenvalues all the more as some of them may be close to the origin. Although it cannot be avoided for computing e.g γ_k in Line 4, only the shifts $\frac{1}{\beta_k}$ (which are only a permuted version of $\lambda_{2,i}\lambda_{2,j}$) are necessary in (5.22).

This fADI method determines the global solution and hopes that the algorithm converges after very few iterations. Truncating the Kronecker rank is applied in the very last step. The solution is then a best rank-r approximation of the unstructured $\mathbf{P}$ contrary to all algorithms developed in this thesis so far including the squared Smith's iteration. Instead of parametrizing the variable with a low-Kronecker rank decomposition, the factors $\mathbf{Z}_k, \mathbf{D}_k, \mathbf{Y}_k$ are expanded until convergence and a low-Kronecker rank structure is not parametrized in the solution before starting. This is a key difference with the rest of the algorithms that we wish to insist upon. As a consequence, the stopping criteria are adjusted: Algorithm 5.2 stops when the residual in the factors $\mathbf{U}^{(\kappa)}$ are zero while the fADI method converges globally and stops when the residual of the discrete Lyapunov is below some threshold.

Remark 5.2. *We comment the more general case where $\mathbf{A} \in \mathcal{K}_{d,1}$. Line 3 of Algorithm 5.3 is then of the form:*

$$(\mathbf{A}_d \otimes \ldots \otimes \mathbf{A}_1 - \alpha\mathbf{I})\mathbf{x} = \mathbf{y} \tag{5.24}$$

The Bartels-Stewart algorithm extends by first, computing Schur decompositions of all $\mathbf{A}_i = \mathbf{Z}_i\mathbf{U}_i\mathbf{Z}_i^T$ where $\mathbf{U}_i$ is upper triangular and $\mathbf{Z}_i$ is orthogonal, and second, introduce the variable $\tilde{\mathbf{x}} = (\mathbf{Z}_d \otimes \ldots \otimes \mathbf{Z}_1)^T\mathbf{x}$ to transform (5.24) into:

$$(\mathbf{U}_d \otimes \ldots \otimes \mathbf{U}_1)\tilde{\mathbf{x}} - \tilde{\mathbf{x}} = \tilde{\mathbf{y}} \tag{5.25}$$

Back substitution finally exploits the Kronecker structure.

5.5. Numerical analysis

5.5.1. Sufficient conditions for a low-Kronecker rank solution

First, we detail how the matrix $\mathbf{A}$ is generated before solving (5.6). The doubling algorithm (5.9) converges if the spectral radius of $\mathbf{A}$ is strictly smaller than one. Such condition is easily ensured when $r_A = 1$ if the factor matrices respect themselves such a condition. If $r_A > 1$, it is more difficult to form a stable global matrix only working on the factor matrices, i.e it is not sufficient that each factor matrix is stable. After generating these factors stable, the global matrix is formed and if it is unstable, one factor matrix is randomly chosen and its entries are divided by a certain value larger than one. We repeat this step until the global matrix is stable. This step is only for generating the simulation data, although it highlights the difficulties to compute in a scalable manner the eigenvalues of the global matrix when the Kronecker rank is larger than one. The spectral radius of $\mathbf{A}$ is approximately equal to 0.99 for all examples considered. We choose the amplitude of $\mathbf{Q}$ such that the Frobenius norm of $\mathbf{A}, \mathbf{Q}$ are similar. The factors have a SSS structure, although this is not further exploited. The factors are in $\mathbb{R}^{50\times 50}$.

5

Squared Smith's iteration The maximum number of iterations is 20, unless the relative difference between two iterates is less than 10^{-6} and the iterations would then stop. The regularization parameter when truncating is 10^{-9}. There are 20 simulations, and the mean and variance of the residual error with the unstructured solution are collected. In Figure 5.1 we display the influence of the Kronecker rank r_P of the solution on the accuracy. The lower the Kronecker rank r of $\mathbf{A}, \mathbf{Q}$, the more accurate the solution for fixed r_P. It is especially remarkable when r is equal to one. We explain this by relating to the equation (5.5): the lower r, the better the approximation of the matrix inverse with a sum of few terms. The compression ratio for all these solutions is significiant: $\frac{2500^2}{r_P \cdot 2 \cdot 50^2} = \frac{1250}{r_P}$.

fADI method The Kronecker rank of $\mathbf{A}$ is now equal to one. The maximum number of iterations for the fADI method is 150, unless the residual as evaluated in Benner and Kürschner (2014) is below 10^{-6} in which case the iterations stop. There is a minimum of 5 shifts parameters. The exact number is adjusted so that the sets $\{\alpha_1, \ldots, \alpha_{k_{\tilde{\mathbf{A}}}}\}$, and $\{\beta_1, \ldots, \beta_{k_{\tilde{\mathbf{B}}}}\}$ are closed under conjugation. 20 Monte-Carlo simulations are computed. The equations (5.22) are solved with the Matlab function, `dlyap`. We have noticed numerical unstabilities when the absolute value of the imaginary part of the eigenvalues of either $\tilde{\mathbf{A}}$ or $\tilde{\mathbf{B}}$ is in the range $]0, \epsilon]$, $\epsilon \approx 10^{-15}$. We insist that real eigenvalues are not contained in this set. The solution of the Sylvester equation using `dlyap` (which uses the Schur-Hessenberg algorithm, and backpropagation) is then slightly different from the one using the backslash operator solving via a LU decomposition which leads to the propagation of numerical errors. These cases are discarded from the analysis. Figure 5.2 compare the squared Smith's iteration and the fADI method when the matrix $\mathbf{A}$ is dense with randomly generated factors using `rand`. Maintaining the structure is all the easier in both algorithms as the Kronecker rank of $\mathbf{A}$ is equal to one.

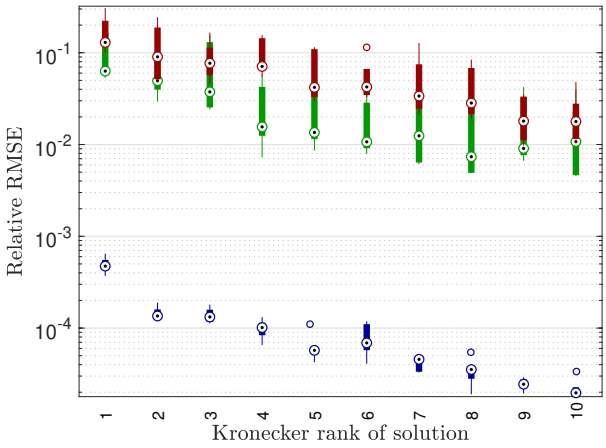


Figure 5.1: Relative RMSE between the unstructured matrix and the Kronecker-structured solution as a function of its Kronecker rank. For each configuration, 25 different matrices $\mathbf{A}, \mathbf{Q}$ are generated. Blue: $r = 1$; Green: $r = 2$; Red: $r = 3$. The vertical bars display the standard deviation.

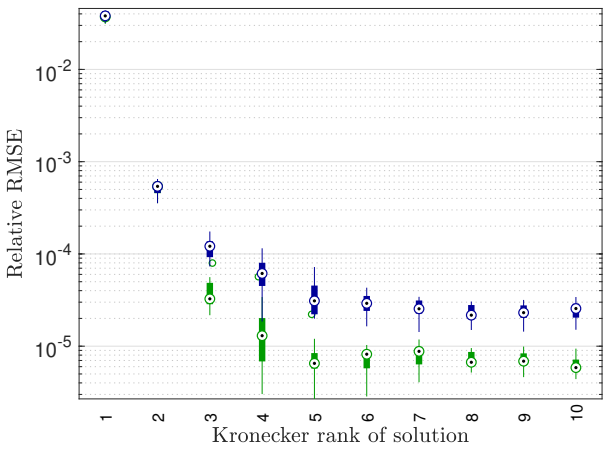


Figure 5.2: Relative RMSE between the unstructured matrix and the Kronecker-structured solution as a function of its Kronecker rank. The Kronecker rank of $\mathbf{A}$ is one and the one of $\mathbf{Q}$ is two. The vertical bars display the standard deviation. Blue refers to the squared Smith's iteration and green refers to the fADI method.

5.5.2. Scalability

The scalability of the proposed algorithm is now compared to the one of Matlab `dlyap` and to the one of the squared Smith's iteration which does not exploit the Kronecker structure. We vary the total number of nodes in the array by increasing

n. Figure 5.3 plots the time as a function of n for unstructured and structured methods and the impact is significant. The variance for the fADI method is larger as the points are more scattered around the fitting line, leading to less confidence in the regression coefficient obtained. Nonetheless, the trends are similar for both structured methods.

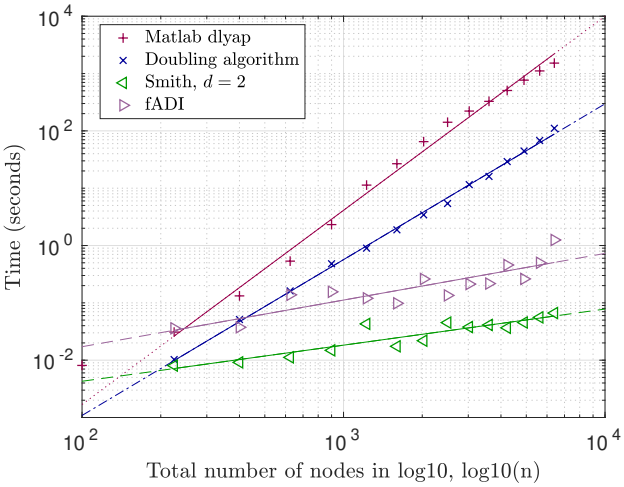


Figure 5.3: Time for solving the discrete Lyapunov equation with the Matlab function dlyap, the doubling algorithm without exploiting the structure, the proposed Algorithm 5.2 with $d = 2$ and the fADI method. The Kronecker rank r_P is equal to two in Figure 5.3. The regression coefficients are equal to 3.35, 2.73, 0.62 and 0.93 for respectively, `dlyap`, the unstructured squared-Smith iteration, Algorithm 5.2 and the factored ADI tailored for low-Kronecker rank matrices.

5.6. Conclusion

In this chapter, we derived computational tools for writing structure-preserving iterative algorithms for solving the discrete Lyapunov equation within the class $\mathcal{K}_{d,r}$. The squared Smith's iteration requires a sequence of matrix-matrix multiplications which also multiplies the Kronecker rank. An Alternating Least Squares algorithm is used to truncate the Kronecker rank of matrices and avoid that it explodes while iterating and destroys all computational improvements. As an alternative, we have shown the equivalence between a Kronecker structured discrete Lyapunov equation and a Sylvester equation with a low-rank solution. This observation was exploited to adapt an existing factored ADI algorithm to the Kronecker structure. Numerical examples have highlighted that the solution of the discrete Lyapunov equation is well-approximated with a sum of few Kronecker terms when the coefficient matrices have such structure. A scalability analysis insisted on the computational improvements with respect to the `dlyap` function from Matlab.

6

Tensor-based predictive control for large-scale SCAO

We propose a data-driven predictive control algorithm for large-scale single conjugate adaptive optics systems. At each time sample, the Shack-Hartmann wavefront sensor signal sampled on a spatial grid of size $N \times N$ is reshuffled into a tensor of order d, not necessarily equal to 2. Its spatial-temporal dynamics are modelled with an autoregressive model of temporal order p where each tensor storing past data undergoes a multi-linear transformation by factor matrices of small sizes. Equivalently, the vector form of this autoregressive model features coefficient matrices parametrized with a sum of Kronecker products between d factor matrices.

When parametrizing each coefficient matrix with a sum of r terms, the computational complexity for updating the sensor prediction online reduces from $\mathcal{O}(pN^4)$ in the unstructured matrix case to $\mathcal{O}(prdN^{\frac{2(d+1)}{d}})$. Most importantly, this model structure breaks away from assuming any prior spatial-temporal coupling as it is discovered from data.

The algorithm is validated on a laboratory test-bed that demonstrates the ability to decompose accurately the coefficient matrices of large-scale autoregressive models with a tensor-based representation, hence achieving high data compression rates and reducing the temporal error especially for large Greenwood per sample frequency ratio.

A similar version of this chapter dealing with only one turbulence disk only previously appeared in: B. Sinquin and M. Verhaegen, "Tensor-based predictive control for extremely large-scale single conjugate adaptive optics," in Journal of the Optical Society of America A, Vol. 35, Issue 9, pp. 1612-1626 (2018). Further experiments have been added, and the section on tensor auto-regressive models removed to avoid repetitions in the thesis.

6.1. Introduction

With a focus on control of Single-Conjugate Adaptive Optics systems, we lay our interest in deriving in a *scalable and data-driven* manner a prediction of the Shack-Hartmann sensor data using a tensor autoregressive model of temporal order p, with $p \geq 1$. Tensorizing the sensor measurements for identifying the spatial-temporal dynamics of the turbulence from data and applying subsequently predictive control is most advantageous when the seeing conditions are such that the quasi-static assumption is not valid and when the sensor array has many lenslets.

The contribution of this chapter is twofold. First, we propose a scalable control law relying on an estimation computed online with $\mathcal{O}(prdN^{2(d+1)/d})$ complexity. Second, we validate the approach on a laboratory testbed to illustrate the decrease of the temporal error over standard non-predictive methods, especially for increasing wind speed all else unchanged.

The outline is as follows. Section 6.2 introduces the AO modelling and presents an overview of the data-driven control algorithm that we propose. We discuss the tensor auto-regressive modelling presented in Chapter 4 in the context of AO in Section 6.3. We describe the laboratory test-bed and detail the calibration procedure in Section 6.5. We evaluate the prediction capabilities of the model structure in an open-loop setting in Section 6.6 and present closed-loop experiments with one turbulence disk in Section 6.7 and a non-pure frozen flow in Section 6.8. Conclusions are drawn in Section 6.9.

Notations specific to this chapter. Most AO-related notations were introduced in Chapter 1, from Section 1.4 on.

6.2. Predictive control in the time domain for adaptive optics

As derived in Chapter 1, the LQG cost function in SCAO trades between minimizing the residual wavefront and the control effort:

$$\min_{\mathbf{u}(k)} \quad \|\widehat{\boldsymbol{\phi}^{tur}}(k+1|k) + \mathbf{H}\mathbf{u}(k)\|_2^2 + \mathbf{u}(k)^T\mathbf{Q}\mathbf{u}(k) \tag{6.1}$$

where $\mathbf{Q}$ is a positive definite matrix. We propose an alternative to (6.1) which consists of minimizing the 2-norm of the residual slopes:

$$\min_{\mathbf{u}(k)} \quad \|\widehat{\boldsymbol{s}^{tur}}(k+1|k) + \mathbf{B}\mathbf{u}(k)\|_2^2 + \mathbf{u}(k)^T\mathbf{Q}\mathbf{u}(k) \tag{6.2}$$

where $\mathbf{B}$ is an interaction matrix relating the slopes induced by the mirror only and the control inputs, $\mathbf{s}^m(k) = \mathbf{B}\mathbf{u}(k-1)$. The limitations of minimizing the residual slopes rather than the residual wavefront have been pointed out in Kulcsár et al. (2012) and are of prime importance in Multi-Conjugate AO especially.

Remark 6.1. *Minimizing the 2-norm of the predicted slopes is equivalent to a weighted least-squares of the wavefront modes. To see this, we write a singular value*

decomposition for $\mathbf{G}$ *with*

$$\mathbf{G} = \begin{bmatrix}\mathbf{U}_1 & \mathbf{U}_2\end{bmatrix} \begin{bmatrix}\mathbf{\Sigma} & 0 \\ 0 & 0\end{bmatrix} \begin{bmatrix}\mathbf{V}_1^T \\ \mathbf{V}_2^T\end{bmatrix} \tag{6.3}$$

The observable part of the wavefront is $\boldsymbol{\varphi}(k) := \mathbf{V}_1^T \boldsymbol{\phi}(k)$*, Hinnen (2007), and then,* $\|\boldsymbol{\phi}(k)\|_2^2 = \|\boldsymbol{\varphi}(k)\|_2^2$*. It yields:*

$$\begin{aligned} \|\widehat{\mathbf{s}}(k+1|k)\|_2^2 &= \|\mathbf{\Sigma}\mathbf{V}_1^T \widehat{\boldsymbol{\phi}}(k+1|k)\|_2^2 \\ &= \|\mathbf{\Sigma}\widehat{\boldsymbol{\varphi}}(k+1|k)\|_2^2 \end{aligned} \tag{6.4}$$

The temporal dynamics of the sensor signals induced by the turbulence only are approximated with a VAR model of temporal order p,

$$\widehat{\mathbf{s}^{tur}}(k+1|k) \approx \sum_{i=0}^{p-1} \mathbf{M}_i \mathbf{s}^{tur}(k-i) \tag{6.5}$$

We investigate the case where the coefficient matrices are parametrized with $\mathbf{M}_i = \sum_{j=1}^{r} \mathbf{M}_{i,j,d} \otimes \ldots \otimes \mathbf{M}_{i,j,1}$ for two integers r and d. The spatial dynamics of the turbulence are embedded within the structure of $\mathbf{M}_i$. By collecting sensor data in open-loop before closing the loop, the factor matrices are estimated offline using the QUARKS algorithm from Chapter 4 (ignoring the assumption of strict positivity). During closed-loop operation, it is possible to reconstruct the signal $\mathbf{s}^{tur}(k)$ at each time sample by subtracting the influence of the previous inputs. It uses the sparse structure of the interaction matrix $\mathbf{B}$ to achieve $\mathcal{O}(N^2)$ complexity. Tensorizing the vector of measurements scales with $\mathcal{O}(N^2)$ whereas computing the prediction with the n-mode matrix product scales with $prdN^{\frac{2(d+1)}{d}}$. Projecting the predicted wavefront on the mirror shall exploit the sparsity in $\mathbf{B}$ and algorithms such as the conjugate gradient are relevant. The control loop is summarized in Algorithm 6.1.

Algorithm 6.1: Control algorithm minimizing the residual sensor measurement with a tensor-based wavefront prediction

Input : $\{\mathbf{M}_{i,j,n}\}_{i=1..p,j=1..r,n=1..d}, \mathbf{B}, \{\mathbf{s}^{tur}(k-i)\}_{i=1..p-1}, \mathbf{s}(k), \mathbf{u}(k-1)$
Output : $\mathbf{u}(k)$

1. $\mathbf{s}^{tur}(k) = \mathbf{s}(k) - \mathbf{B}\mathbf{u}(k-1)$
2. Reshuffle $\mathbf{s}^{tur}(k)$ into $\boldsymbol{\mathcal{S}}^{tur}(k)$
3. $\widehat{\boldsymbol{\mathcal{S}}^{tur}}(k+1|k) = \sum_{i=0}^{p-1} \sum_{j=1}^{r} \boldsymbol{\mathcal{S}}^{tur}(k-i) \times_1 \mathbf{M}_{i,j,1} \times_2 \ldots \times_d \mathbf{M}_{i,j,d}$
4. Reshuffle $\widehat{\boldsymbol{\mathcal{S}}^{tur}}(k+1|k)$ into $\widehat{\mathbf{s}^{tur}}(k+1|k)$
5. Solve the sparse least-squares (6.2) to get $\mathbf{u}(k)$

Remark 6.2. *A possible extension consists of formulating the tensor-based VAR model on the wavefront data reconstructed with e.g de Visser et al. (2016) which provides an estimate of* $\boldsymbol{\phi}^{tur}(k+1)$*. The prediction* $\widehat{\boldsymbol{\phi}^{tur}}(k+1|k)$ *is then used in*

(6.1) *to compute the control inputs. It would however not be a stand-alone method as would be a Kalman filter doing both the wavefront reconstruction and the prediction in one step.*

6.3. Tensorizing the sensor data

The measurement signal at time instant k is available on a regular 2D grid of size $N \times N$. From an input-output data perspective, tensorizing the sensor data corresponds to partitioning the 2D sensor array. The vector $\mathbf{s}^{tur}(k)$ is reshaped into a tensor denoted as $\mathcal{S}^{tur}(k) \in \mathbb{R}^{J_1 \times \ldots \times J_d}$. Each sensor signal at node i, j is re-indexed with a tuple of size d rather than with two position indices. The dimensions $\{J_i\}_{i \in \{1,\ldots,d\}}$ correspond to the size of the partition. Figure 6.1 illustrates a possible partition for $d = 4$ and $N = 32$.

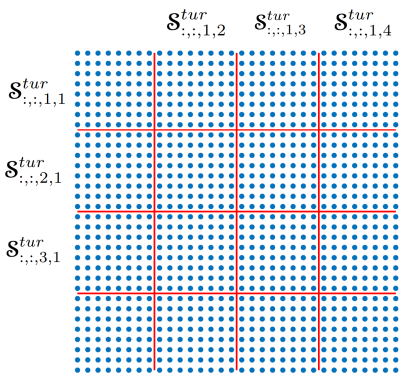


Figure 6.1: Partitioning a 2D array of sensor data with 32×32 nodes (in blue) with a 4th order tensor $\mathcal{S}^{tur}(k) \in \mathbb{R}^{8\times8\times4\times4}$. The red lines indicate the partition into blocks of 8×8 matrices. For example, the data inside the block on the upper-left side is placed into $\mathcal{S}^{tur}_{:,:,1,1}(k)$.

The choice of a partition is highly not unique as there are different partitions that result in the same number of stored parameters (as counted from the number of entries in the factor matrices). An optimal partition is such that it minimizes the data storage while minimizing the prediction-error of the VAR model for a given temporal order. If d is fixed, a general idea for minimizing the number of parameters is to construct factor matrices of similar sizes for all $n \in \{1, \ldots, d\}$, i.e J_i close to $N^{2/d}$. As further indication, we note that *in general*, choosing $d = 2$, with $J_1 = 2N, J_2 = N$ to take advantage of the two-level block structure maximizes the accuracy. When increasing d, less data is stored and the accuracy decreases w.r.t the case $d = 2$ which is partly compensated by increasing r. In the context of adaptive optics, the choice also depends on the geometry of the sensor and the location of available measurements.

Two similarities can be drawn between such tensor models and the splined-based wavefront reconstruction method, D-SABRE de Visser et al. (2016). First, both methods partition the sensor array into zonal parts. Second, and although the wavefront reconstruction is performed separately for each local zone, the splines are

merged into one global wavefront assuming a continuity order. The latter shares similar consequences with the Kronecker rank: the larger, the lower the error w.r.t the original matrix although the less data compression.

The sensor data is available on a circular array and therefore, the rectangular embedding is considered by padding with 0 at the edges. This modification is likely to introduce errors at the boundaries that are all the more important as the coupling with far-away nodes is significant.

Example 6.1. *We illustrate the role of the Kronecker rank on tensor decompositions for modelling two-level matrices for a static input-output system. Let* $N = 32, m = 1, \sigma \in \mathbb{R}$ *and* $(i, j, k, l) \in \{1, ..., N\}^4$. *The matrix* $\mathbf{M} \in \mathbb{R}^{N^2 \times N^2}$ *is set as follows. For* $x = (i-1)N + j$ *and* $y = (k-1)N + \ell$,

$$m_{x,y} = e^{-\frac{(i-k)^2+(j-\ell)^2}{\sigma^2}} \tag{6.6}$$

where $\sigma = 0.54$. *The matrix* $\mathbf{M}$ *is representative of an influence function matrix between the wavefront and the deformable mirror commands although the parameter* σ *is set to a large value, hence showing that the Kronecker decomposition does not rely on sparsity in the entries. Using randomly generated data* $\mathbf{u}(k), \mathbf{y}(k)$ *with* $N_t = 500$ *temporal samples, we solve the optimization problem with the ALS presented in Chapter 4 (without strict positivity constraints),*

$$\min_{\mathbf{M}_{i,j,n}} \quad \sum_{k=1}^{N_t} \|\mathbf{y}(k) - \sum_{j=1}^{r} (\mathbf{M}_{i,j,d} \otimes \ldots \otimes \mathbf{M}_{i,j,1})\mathbf{u}(k)\|_2^2 \tag{6.7}$$

The number of iterations in the ALS is set to 20. *The estimates for the factor matrices are denoted with* $\widehat{\mathbf{M}}_{i,j,n}$, *and for the global matrix with* $\widehat{\mathbf{M}}$.

The relative root-mean-square error is computed between vec$(\mathbf{M})$ *and vec*$(\widehat{\mathbf{M}})$ *and is shown in Table 6.1. The exponential function is separable in both the horizontal and vertical coordinates, and therefore* $\mathbf{M}$ *is exactly decomposed with a single Kronecker product of two matrices. This illustrates the global convergence of the algorithm. The large discrepancy between the different configuration does not happen in practice because of noise and non-exact separability of the functions. When increasing the tensor order, approximations are made and we are looking for a trade-off between model accuracy and its complexity. Partitioning the matrix* $\mathbf{M}$ *requires a sum of* r *Kronecker matrices, with* $r > 1$. *The compromise is illustrated in Table 6.2 with the Akaike information criteria (AIC) used for selecting a model structure, Akaike (1974):*

$$AIC = \frac{1}{N_t}(R + dr \sum_{i=1}^{d} J_i^2) \tag{6.8}$$

where R *is the residual of the cost function* (6.7). *The lower the criteria, the more information the model structure provides with respect to the alternatives. Owing to the separability of* (6.6), *the optimal model structure corresponds to* $(r, d) = (1, 2)$. *Among the decompositions with* $d > 2$, *the configuration* $(r, d) = (3, 3)$ *is optimal*

from the AIC criteria. In Figure 6.2 and Figure 6.3, we display the identified matrix $\widehat{\mathbf{M}}$ *obtained by parameterizing with a high-order QUARKS structure and optimizing from noise-free input-output data. A particular patch-wise structure is observed for all the matrices when* $d > 2$ *and* $r = 1$*, which disappears gradually when increasing the Kronecker rank.*

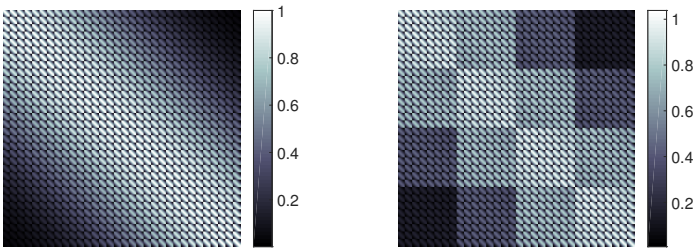


Figure 6.2: Entries of the estimated matrix $\widehat{\mathbf{M}}$ with: (left) $d = 2$ and $\mathcal{Y}(k) \in \mathbb{R}^{32\times32}$, (right) $d = 3$ and $\mathcal{Y}(k) \in \mathbb{R}^{32\times8\times4}$.

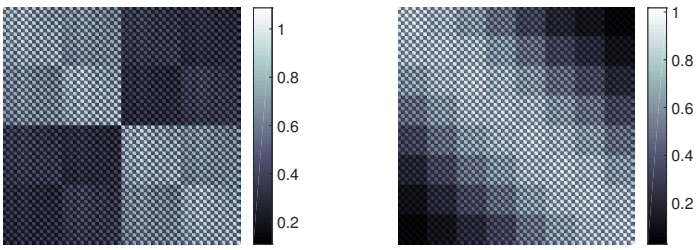


Figure 6.3: Entries of the estimated matrix $\widehat{\mathbf{M}}$ with: (left) $d = 4$ and $\mathcal{Y}(k) \in \mathbb{R}^{16\times16\times2\times2}$, (right) $d = 4$ and $\mathcal{Y}(k) \in \mathbb{R}^{8\times8\times4\times4}$.

Kronecker rank, r	1	3	5
32×32	$7.07 \cdot 10^{-15}$	$1.34 \cdot 10^{-12}$	$2.66 \cdot 10^{-13}$
$32 \times 8 \times 4$	$1.50 \cdot 10^{-1}$	$1.5 \cdot 10^{-1}$	$1.95 \cdot 10^{-3}$
$16 \times 16 \times 2 \times 2$	$3.64 \cdot 10^{-1}$	$2.78 \cdot 10^{-1}$	$2.67 \cdot 10^{-1}$
$8 \times 8 \times 4 \times 4$	$2.70 \cdot 10^{-1}$	$1.18 \cdot 10^{-1}$	$9.18 \cdot 10^{-2}$

Table 6.1: Influence of the Kronecker rank r on the relative Root Mean Square Error between $\text{vec}(\mathbf{M})$ and $\text{vec}(\widehat{\mathbf{M}})$.

Kronecker rank, r	1	2	3
32×32	0.612	0.913	1.09
$32 \times 8 \times 4$	3.84	2.09	1.05
$16 \times 16 \times 2 \times 2$	4.61	4.37	3.86
$8 \times 8 \times 4 \times 4$	4.35	3.62	3.31

Table 6.2: Akaike information criteria for the different model structures (in $log10$).

6.4. Computational gains

In Figure 6.4 is plotted the ratio of improvement in computational complexity for obtaining a prediction with the tensor model with respect to the unstructured counterpart. Assuming a sensor grid of 200×200, the improvement in computational complexity with respect to the unstructured approach is summarized in Table 6.3 for different tensor orders.

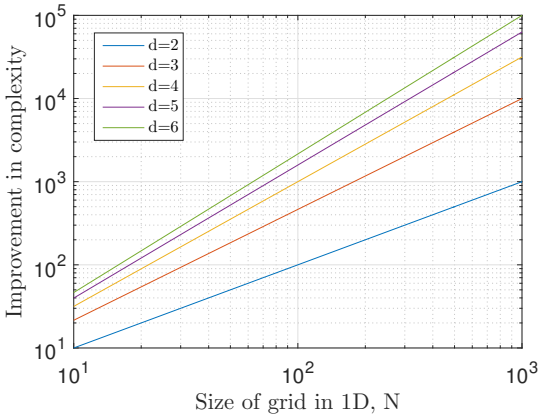


Figure 6.4: Ratio $\frac{N^4}{N^{2(d+1)/d}}$ which reflects the improvement in the computational complexity w.r.t the unstructured case for computing online a prediction as a function of the size of the array.

Tensor order, d	2	3	4	5	6
Improvement ($\times 10^3$)	0.067	0.39	0.94	1.6	2.3

Table 6.3: Improvement in complexity with $N = 200, r = 3$

6.5. The experimental testbed

6.5.1. Description of the system

A schematic represents the testbed in Figure 6.5 and a picture shows the hardware in Figure 6.6.

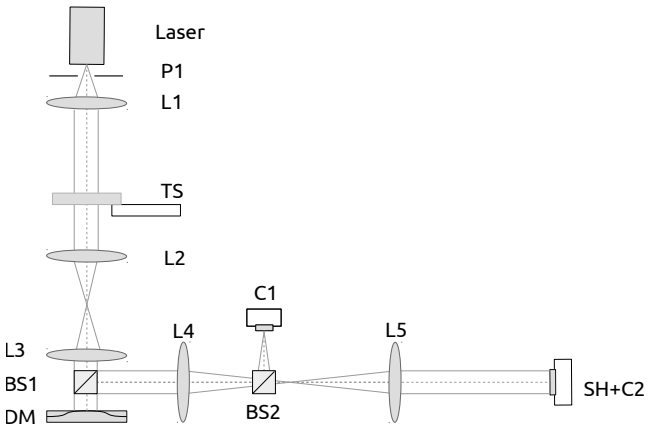


Figure 6.5: Schematic view of the laboratory testbed. P1 is a pin-hole, L1 till L5 are lenses, TS is a rotating disk for simulating the turbulence, BS1 and BS2 are beam splitters, DM is the kilo-DM, C1 is the Point-Spread-Function camera, SH+C2 is the wavefront sensor.

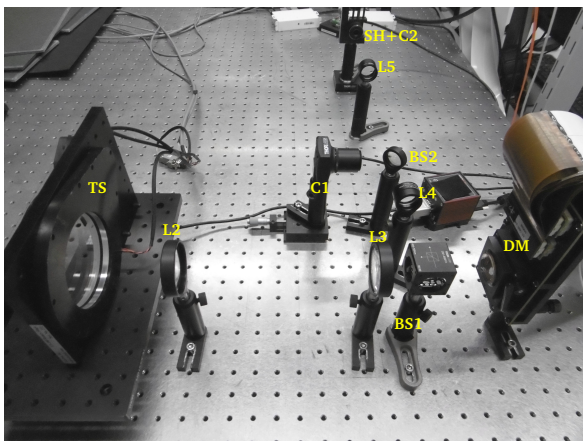


Figure 6.6: Annotated picture of the laboratory testbed.

The light is emitted from a laser source at a wavelength $\lambda = 635$nm and is collimated into a beam of size $D = 9$mm with the lens L_1. The beam then goes through a turbulence disk TS placed at the focal plane of the lens L2 of length $f_2 = 10$cm. The atmospheric turbulence for a single frozen layer is generated using a pseudo-random phase plate machined by Lexitek, Inc. The phase functions consist of a sandwich of acrylic and an optical polymer which is itself sandwiched between two glass windows, 10mm thick with a broadband visible AR coating ($< 0.6\%, \lambda = 425 - 675$nm). The optical path difference is defined as follows. A phase design that follows the spatial

Kolmogorov distribution is generated and is then multiplied by $(1 + 1/5 \cdot \sin\theta)^{-5/6}$, where θ is the central angle. The local value of the Fried parameter r_0 varies as $(1 + 1/5 \cdot \sin\theta)$, i.e from 1.2 to 1.8mm, and simulates temporally-varying seeing conditions. The scaling factor along with the phase profile for TS is displayed in Figure 6.7. The ratio D/r_0 ranges between 5 and 7.5. The Greenwood frequency is a linear function of the wind speed which is adjusted by rotating the disk at different rounds per minute. In this work, open- and closed-loop experiments are carried out for $\bar{f} = f_G/f_S$ sampled between 0.026 and 0.43. This range is such that the ratio remains below the Nyquist criteria, and hence no temporal aliasing occurs. Most AO systems whose control algorithm does not include a prediction are likely to operate in the lower end of this range.

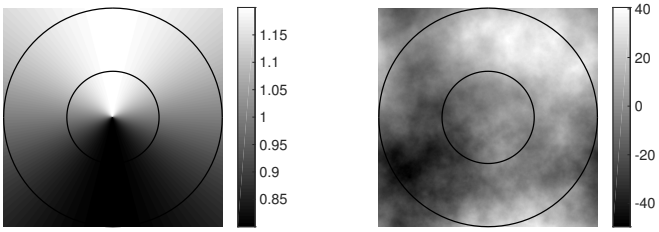


Figure 6.7: Left: 2D profile for the Fried parameter r_0 on the turbulence disk TS inside the annular aperture. Right: turbulence profile in radian for TS. The phase values are radians of phase at the wavelength λ.

The lenses L2 and L3 of focal length equal to 10mm conjugate the plane of the disturbance TS with the deformable mirror kilo-DM from Boston Micromachines Corp. A number of 952 actuators (among which 5 of them are failing) are positioned within the circular aperture of a regular grid of size 34×34. We denote $N_u = 34$, $n_u = 952$. The stroke is 1.8μm, the actuator pitch is 300μm and there is no hysteresis. The response time from 10% to 90% of the mechanical response is below 20μs. The coupling between the actuators is 15% with an error margin of 5%. The electrostatic actuation is such that the actuators can only be pulled, and hence the control commands are positive only. We define the working point of the mirror u_{bias} as the middle value of the linear range. This sets a new 0 and allows the control inputs $\mathbf{u}(k)$ to be both positive and negative. Hence, they are related to the mirror commands $\mathbf{u}_{DM}(k)$ (in voltages) with, for all $i = 1..n_u$:

$$u_{DM,i}(k) = u_i(k)^2 + u_{bias} \tag{6.9}$$

The mirror is poked to the offset value u_{bias} for calibration, collecting open-loop data and for closed-loop operation so as to make sure that there is no additional defocus introduced on the sensors between the calibration and the closed-loop. Moreover, at each iteration of the control loop, the mean of the squared control inputs is substracted and the bias is added. Due to the failing actuators, it however does not correspond to a perfect piston mode.

The light beam is reflected by the mirror to the lens L4 of focal length 150mm and divided in two using the beam splitter BS2. One part of the beam is focused on C1, a Thorlabs-CMOS camera DCC1545M with 1280×1024 pixels, which records the Point-Spread-Function (PSF). The exposure time is 3ms. The second part travels to the lens L5 of focal length 150mm and reaches the wavefront sensor denoted with SH+C2 in Figure 6.5. An OKOtech Shack-Hartmann wavefront sensor, 1-inch optical format, with a lenslet array pitch of 300μm and focal length 18.6mm, is placed perpendicular to the optical beam path at the focal point of L5. An array of 30×30 lenslets is selected among which 689 are illuminated within a circular aperture and considered as active, hence providing with 1378 slopes measurements. The set of active lenslets is denoted with Ω_{SH}. The exposure time is 1.3ms. Both cameras collect images with 8 bits. The distance between two micro-lenses and the actuator pitch are both equal to 300μm. Use is not made of all available actuators n_u: the illuminated pupil on the DM is slightly smaller than the actuated area of the DM. The set of actuators that are active on the SH array is indexed and used for defining the vector of control inputs, $\mathbf{u}(k) \in \mathbb{R}^{706}$. There are approximately 5 actuators and lenslets per r_0. The wavefront is hence well spatially sampled.

The operating system is Ubuntu 14.04 LTS. The offline operations for identifying models are carried out with Matlab R2016b while the online algorithms are implemented in C along with CUDA. The loop runs at a sampling frequency $f_S = 10$Hz. Within $1/f_S$ seconds, the following operations are carried out sequentially. The DM commands are first sent, a PSF image and a WFS image are then taken, the slopes are calculated and Algorithm 6.1 is computed. A region of interest in the PSF image centered around the maximum value is stored in a Matlab file at each time sample. The PSF and WFS camera do not record simultaneously. Data-driven predictive control methods require the sampling frequency used for control to be exactly set to the one used for identification and as little jitter as possible is allowed. Due to the sequential nature of the operations presented here, the image recorded on the PSF camera does not correspond to an optimal correction as would be obtained when the inputs are applied simultaneously with the acquisition of a SH image every $1/f_S$ seconds. It is nonetheless less prominent when the exposure time of the PSF camera is small w.r.t the sampling period.

6.5.2. Control approach used for comparison

Algorithm 6.1 is compared to a non-predictive controller commonly used in AO. This controller relies on the quasi-static assumption,

$$\widehat{\boldsymbol{\phi}^{tur}}(k+1|k) = \boldsymbol{\phi}^{tur}(k) \tag{6.10}$$

for modeling the turbulence temporal dynamics, which ignores all spatial-temporal coupling. The control law is a Proportional-Integrator, Ellerbroek (2002):

$$\mathbf{u}(k) = \frac{c_1}{1 - c_2 z^{-1}} \mathbf{R}\mathbf{s}(k) \tag{6.11}$$

for (c_1, c_2) two real scalars carefully tuned and z the forward shift-operator. Let $\mathbf{C}_e$ be the covariance matrix of the measurement noise. The gain matrix $\mathbf{R}$ is expressed

with,

$$\mathbf{R} = \underbrace{(\mathbf{H}^T\mathbf{H} + \mathbf{Q})^{-1}\mathbf{H}^T}_{\text{Mapping to the mirror inputs}} \quad \underbrace{\mathbf{C}_\phi\mathbf{G}^T(\mathbf{G}\mathbf{C}_\phi\mathbf{G}^T + \mathbf{C}_e)^{-1}}_{\text{Wavefront reconstruction}} \tag{6.12}$$

The matrix $\mathbf{R}$ is dense, and therefore the equation (6.11) is unpractical for large-scale AO as the online cost scales with $\mathcal{O}(N^4)$.

6.5.3. Calibrating the system

The grid of the Shack-Hartmann pattern and the spacing between each spot is determined with a Fast-Fourier Transform of a calibration image. In each box allocated to a single lenslet, the center of gravity is computed after truncating all values below 5% of the maximum. Once the position of all centroids has been obtained for one image, the procedure is repeated so as to average the centroids location over 50 reference images. The noise and wavefront covariance matrices are required for the common control approach. The diagonal covariance matrix $\mathbf{C}_e$ in (5.2) is determined by collecting $2 \cdot 10^3$ WFS samples with no aberration. The wavefront covariance matrix $\mathbf{C}_{\boldsymbol{\phi}}$ is computed assuming Kolmogorov turbulence. Moreover, we determine the Signal-to-Noise Ratio (SNR) of the system with a static turbulence layer by collecting again $2 \cdot 10^3$ temporal samples of the slopes. The SNR w.r.t the noisy samples with no aberration has a mean of 6.7. It varies as a function of r_0.

A zone without turbulence in the middle part of the disk is used for calibration. No defocus is hence introduced on the WFS camera compared to the case when the beam goes through the turbulence-printed area of the disk TS. In this case, $\mathbf{s}^m(k) = \mathbf{s}(k)$ and $\boldsymbol{\phi}^m(k) = \boldsymbol{\phi}(k)$. First, we poke all actuators of the mirror with the maximum stroke allowed within the linear range. The SH spots nonetheless remain within the box used for computing the centroids. This data enables to retrieve the approximate position of each actuator on the SH grid and to determine a neighborhood $\Omega_{u_i,s}$ in terms of lenslets that it influences. For each actuator u_i poked, every lenslet whose value is above $0.05 \times \max(|\mathbf{s}(k)|)$ is included within the neighborhood $\Omega_{u_i,s}$. The neighborhood of each lenslet in terms of actuators, $\Omega_{s_i,u}$, is determined from $\Omega_{u_i,s}$ for all i.

Second, $2.5 \cdot 10^3$ temporal samples of slope measurements are collected with a random poking of all actuators following a uniform distribution. The first $1.5 \cdot 10^3$ temporal samples are used for identifying a model while the remaining points are used for validation. We denote N_{ide} as the length of the identification batch, similarly N_{val} for the validation set. A sparse least-squares to estimate $\mathbf{B}$ is solved using the knowledge of the neighborhood $\Omega_{s_i,u}$. For all $i \in \{1, ..., \text{card}(\Omega_{SH})\}$:

$$\min_{\mathbf{B}(i,\Omega_{s_i,u})} \sum_{k=2}^{N_{ide}} \| \underbrace{\begin{bmatrix} \mathrm{s}_{x_i}(k) \\ \mathrm{s}_{y_i}(k) \end{bmatrix}}_{\mathbf{s}_i(k)} - \mathbf{B}(i, \Omega_{s_i,u})\mathbf{u}_{\Omega_{s_i,u}}(k-1)\|_2^2 \tag{6.13}$$

where $\mathrm{s}_{x_i}(k), \mathrm{s}_{y_i}(k)$ represent respectively the horizontal and vertical slopes for the lenslet i. The estimated slopes $\widehat{\mathbf{s}}_i(k)$ are computed from the estimate $\widehat{\mathbf{B}}$, and the

Variance Accounted For (VAF) is computed for each sensor output on the validation dataset and then averaged for all the channels. The VAF on validation data for estimating $\mathbf{B}$ with (6.13) reaches 96.84%. There remains unmodeled spatial dynamics that are due to misalignment of the optics, sensor noise and the approximated linear range of the DM. The influence matrix from the input to the wavefront $\mathbf{H}$ is required for computing the PI controller in (6.11). The wavefront within a circular grid Ω_ϕ and induced by the mirror is reconstructed with a stochastic least-squares,

$$\widehat{\boldsymbol{\phi}^m}(k) = \mathbf{C}_\phi \mathbf{G}^T (\mathbf{G}\mathbf{C}_\phi \mathbf{G}^T + \mathbf{C}_e)^{-1} \mathbf{s}^m(k) \tag{6.14}$$

A sparse matrix $\mathbf{H}$ is subsequently obtained solving a sparse least-squares:

$$\min_{\mathbf{H}(\Omega_{\phi_i,u},i)} \|\widehat{\mathbf{B}}(:,i) - \mathbf{G}(:,\Omega_{\phi_i,u})\mathbf{H}(\Omega_{\phi_i,u},i)\|_2^2 \tag{6.15}$$

where $\Omega_{\phi_i,u}$ denotes the neighborhood of each wavefront node in terms of actuators indices. The VAF between vec($\widehat{\mathbf{B}}$) and vec($\mathbf{G}\widehat{\mathbf{H}}$) is 94.86%.

6.6. Analysis of predictive algorithms using open-loop data

We have mentioned in Chapter 1 that the LQG criteria is addressed in two steps, first estimating $\boldsymbol{\phi}^{tur}(k+1)$ from previous data, then solving the LQR optimization. The performances of Algorithm 6.1 are then assessed in two steps, first we evaluate the predictive performances of the tensor autoregressive model using open-loop data, and closed-loop performances are then studied. Separating the analysis allows to investigate the performance of the prediction only, without adding fitting errors and additional noise that appear in closed-loop, e.g when computing $\mathbf{s}^{tur}(k)$.

Sensor data is first collected while actuating the mirror to a constant offset voltage (simulating the virtual 0 position for the actuators). Second, an autoregressive model is computed using the first 2/3 of the dataset. Third, the last part of the dataset is used for checking the quality of the model and whether the prediction is valuable. This validation step is as follows: the matrices estimated are plugged into (6.5) to compute $\widehat{\mathbf{s}^{tur}}(k+1|k)$. The wavefront $\widehat{\boldsymbol{\phi}^{tur}}(k+1|k)$ is then reconstructed from $\widehat{\mathbf{s}^{tur}}(k+1|k)$ following (6.14). The actual wavefront at time $k+1$ is similarly estimated from the actual value $\mathbf{s}^{tur}(k+1)$. The temporal error is defined with a mean-square error (MSE) criteria between the prediction $\widehat{\boldsymbol{\phi}^{tur}}(k+1|k)$ and the actual value $\boldsymbol{\phi}^{tur}(k+1)$:

$$\sigma^2 = \frac{1}{N_{val}} \sum_{k=0}^{N_{val}-1} \|\widehat{\boldsymbol{\phi}^{tur}}(k+1|k) - \boldsymbol{\phi}^{tur}(k+1)\|_2^2 \tag{6.16}$$

where N_{val} is the number of temporal samples in the validation dataset. It is the variance of the residual wavefront over the full phase screen.

The different model structures used for predicting $\mathbf{s}^{tur}(k+1)$ (and hence, $\boldsymbol{\phi}^{tur}(k+1)$) are:

1. Using the autoregressive model in (6.5) with $p = 1, \mathbf{M}_0 = \mathbf{I}_{m \times \text{card}_{\Omega_{SH}}}$. It corresponds to the assumption of quasi-static turbulence, $\widehat{\mathbf{s}^{tur}}(k+1|k) = \mathbf{s}^{tur}(k)$ and is labeled with *Static* in Figure 6.8.

2. Using the autoregressive model in (6.5) with $p = 5$ and a localized pattern. The prediction is written with:

$$\widehat{\mathbf{s}_j^{tur}}(k+1|k) = \sum_{i=1}^{p} \sum_{\bar{j} \in \mathcal{N}_j} \mathbf{A}_{j,i,\bar{j}} \mathbf{s}_{\bar{j}}^{tur}(k-i) \tag{6.17}$$

   where $\mathbf{A}_{j,i,\bar{j}}$ is a 2×2 matrix identified in least-squares sense using the identification data batch. For each lenslet, every other lenslet located closer than 120 pixels on the SH camera belongs to the neighborhood, including a maximum of 17 neighbours per lenslet.

3. Using the autoregressive model in (6.5) with a sums-of-Kronecker parametrization. The three tensor decompositions analyzed are displayed in Table 6.4. The factor matrices are identified using the ALS for QUARKS following the lines presented in Chapter 4 although without strict positivity constraints.

Tensor order, d	**Size of factor matrices**, J
2	$(60, 30)$
3	$(12, 6, 25)$
4	$(12, 5, 6, 5)$

Table 6.4: Partitions in the SH sensor associated with the QUARKS model

A number of $3 \cdot 10^3$ temporal samples are collected in open-loop at the frequency f_S. The first $2 \cdot 10^3$ of them are used for identification while the remainder is used for validation. Each channel is detrended before Algorithm 4.1 is applied. In the forthcoming figures, ξ is the relative RMSE between the interpolation with a second order polynomial and the dataset.

The MSE as a function of the Greenwood per sample ratio is shown in Figure 6.8. For low values of $\bar{f}$, the turbulence is quasi-static and all data-driven models result in small residual errors. When increasing $\bar{f}$, the quasi-static assumption is not valid anymore and the prediction performances for the static model worsen gradually. The banded pattern on the coefficient-matrices is well suited for values of $\bar{f}$ below 0.1, after which the MSE increases significantly. Increasing the bandwidth of the coefficient-matrices in the sparse model would have resulted in denser models and thus, more coefficients to estimate. Importantly, the tensor representation breaks away from assuming a given ratio $\bar{f}$ as is required for the sparse structure to determine the width of the band. The triplet (p, r, d) tunes the trade-off between data compression (and computational efficiency for online purposes) and prediction error. The triplet

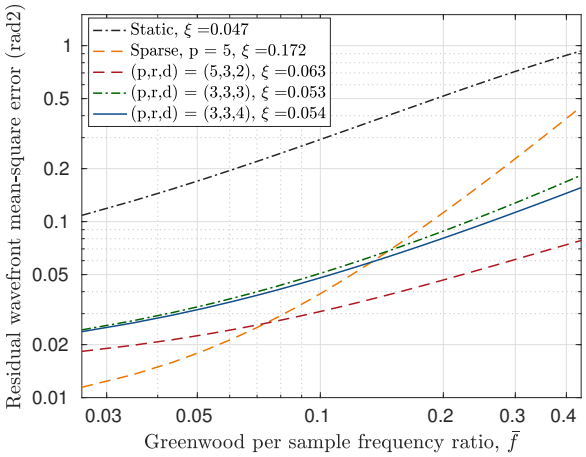


Figure 6.8: Comparison of different model structures. MSE on validation data as a function of the Greenwood per sample frequency ratio.

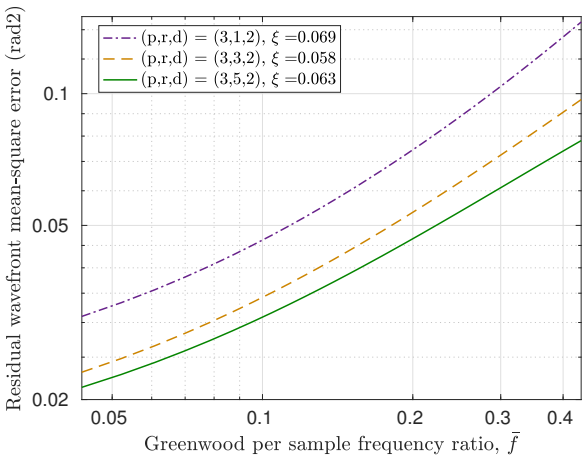


Figure 6.9: MSE on validation data as a function of the Greenwood per sample frequency ratio, $\bar{f}$, when increasing the Kronecker rank. The turbulence consists of one single disk.

is however dependent on the model structure chosen, the latter implying a different trade-off between the bias and the variance in the least-squares fit. For example, when the tensor order d increases from $d = 2$ to larger d all else unchanged, the bias increases as the fitting capability of the structure decreases all the more as we are identifying dynamics on a 2D grid. For example, a tensor representation with $d = 4$ reaches a prediction error slightly larger than the case $d = 2$ while the number of stored entries in the factor matrices is $pr \times 122$ contrary to $pr \times 4500$. For a given choice of the parameter d and the associated tuple $(J_1, \ldots, J_d)$, increasing the parameter r decreases the prediction error as shown in Figure 6.9: the fitting capabilities of the model increase while the data compression is less efficient.

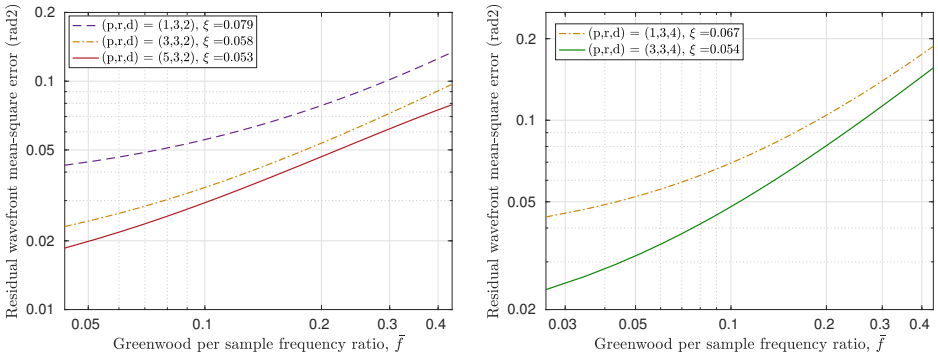


Figure 6.10: MSE on validation data as a function of the Greenwood per sample frequency ratio, $\bar{f}$. The turbulence consists of one single disk. (a) and (b): Increasing temporal order for respectively $d = 2$ and $d = 4$.

In Table 6.5 we show the relative improvements that occur when increasing the Kronecker rank in the case $d = 2$. It is averaged in different ranges of $\bar{f}$ to highlight the main trends. The larger $\bar{f}$, the more useful it is to increase r for accurate estimations. The relative improvement is less significant from $r = 3$ to $r = 5$ than from $r = 3$ to $r = 1$. It highlights that approximating the spatial-temporal dynamics of the sensor data with a function separable in d-dimensions is a valid assumption, even for low-values of the Kronecker rank.

$(r_a, p_a) \to (r_b, p_b)$	$\bar{f} \in [0.026, 0.061]$	$\bar{f} \in [0.069, 0.10]$	$\bar{f} \in [0.11, 0.15]$	$\bar{f} \in [0.15, 0.22]$	$\bar{f} \in [0.24, 0.31]$	$\bar{f} \in [0.33, 0.40]$
$(3,1) \to (3,3)$	0.23	0.22	0.29	0.27	0.29	0.28
$(3,3) \to (3,5)$	0.067	0.10	0.11	0.12	0.13	0.14
$(1,3) \to (3,3)$	0.48	0.30	0.35	0.30	0.28	0.24
$(3,3) \to (5,3)$	0.16	0.14	0.13	0.12	0.10	0.11

Table 6.5: Relative improvement on σ^2 when increasing either the temporal order or the Kronecker rank while $d = 2$. $(r_a, p_a) \to (r_b, p_b) := |\sigma^2_{(p,r)=(p_a,r_a)} - \sigma^2_{(p,r)=(p_b,r_b)}|/\sigma^2_{(p,r)=(p_a,r_a)}$

Increasing the temporal order p only leads to decreased prediction-error before reaching a plateau as shown in Figure 6.9 and in Table 6.5. Classical identification of autoregressive models recast this estimation problem as a structure selection, Ljung (1999).

6.7. Closed-loop performances

In this section, the performances of Algorithm 6.1 are evaluated in closed-loop. The experiment duration is $5 \cdot 10^2 \times f_S$ seconds. The matrix $\mathbf{Q}$ weighting the control inputs is set to $\lambda \mathbf{I}$ with $\lambda = 2 \cdot 10^{-3}$. A first run in closed-loop is carried out by setting the inputs to the reference u_{bias}. It is referenced with a *No control* label. The disk is then set back to the home position. The MVM control in (6.11) plays the role of a baseline. The pair (c_1, c_2) is tuned at each Greenwood per sampling frequency ratio. Algorithm 6.1 is tested for different values of the triplet (p, r, d).

Figure 6.14 displays the long-exposure PSF for $\bar{f} = 0.069$ and $\bar{f} = 0.24$. The PSF is much sharper using the predictive algorithm than with the MVM as expected.

The Strehl ratio is computed based on the PSF image obtained by averaging 250 frames of the camera C1. The Strehl ratio is computed following the guidelines in Hinnen (2007). Let $\bar{I}(\mathbf{p})$ denote the intensity of the long-exposure image at the pixel coordinate $\mathbf{p}$ in the CCD frame, and $\underline{I}(\mathbf{p})$ the intensity of the diffraction-limited PSF image obtained with no turbulence at position $\mathbf{p}$. All values below 2% of the CCD range are first thresholded to 0. Let the maximum intensity be denoted with $\bar{I}_0$ and its position with $\mathbf{p}_0$. Let q_0 denote the radius between the center of the theoretical Airy disk and the position where the first minimum occurs. Let $\mathcal{B}_r(\mathbf{p}_0) = \{\mathbf{p} \in \mathcal{I}, \|\mathbf{p} - \mathbf{p}_0\|_2 < r\}$. Summing the intensities in the neighborhood $\mathcal{B}_{2q_0}(\mathbf{p}_0)$ yields the flux $\bar{I}_{2q_0}$. The Strehl ratio is computed with:

$$S = \frac{\bar{I}_0}{\bar{I}_{2q_0}} \frac{\underline{I}_{2q_0}}{\underline{I}_0} \tag{6.18}$$

Figure 6.15 depict the Strehl ratio as a function of $\bar{f}$. The MVM performances degrade with increasing $\bar{f}$ while Algorithm 6.1 reduces the temporal error. The relative performances for different tensor orders are different than for the open-loop case: we notice better disturbance rejection when $d = 4$ especially for large $\bar{f}$. One reason is the robustness to noise of a parametrization with fewer coefficients. Moreover, the differences between the random walk assumption and the prediction obtained using a tensor AR model in Figure 6.8 are attenuated in Figure 6.15. One

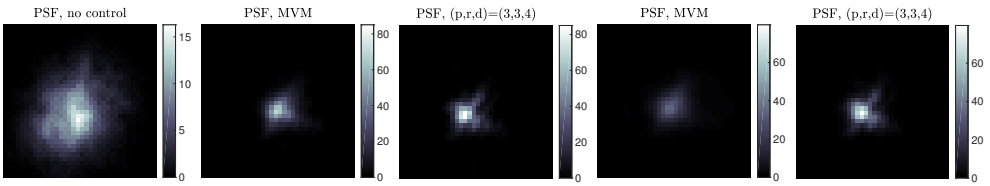


Figure 6.14: Long-exposure PSF for the different methods. Second row: PSD of the residual error between the turbulence-free PSF and the long-exposure PSF. First column: there is no control and the mirror is flat. The data corresponding to the MVM is represented in the second and fourth columns, and the data corresponding to Algorithm 6.1 in the third and fifth columns. The Greenwood per sample frequency ratio $\bar{f}$ is 0.0696 in columns 2 and 3, and 0.2435 in columns 4 and 5. The scale for the columns 2 and 3 is identical. Similarly for the columns 4 and 5.

reason is that the DM cannot take any arbitrary shape that is predicted and the high spatial frequencies of the prediction vector are filtered out.

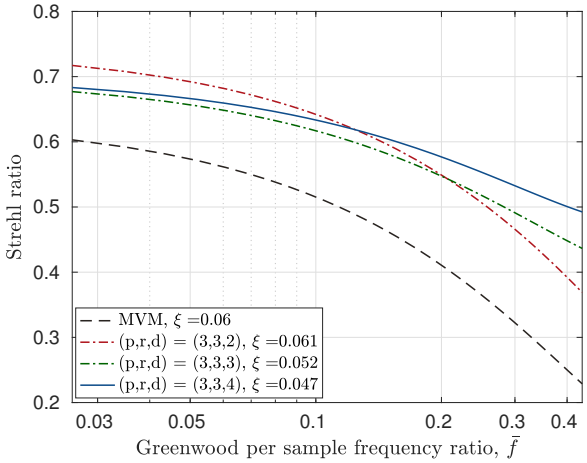


Figure 6.15: Strehl ratio as a function of the Greenwood per sample frequency ratio. The turbulence consists of a single disk.

When increasing gradually the radius for computing the encircled energy, the normalized value converges to one. In Figure 6.16 is shown the relative improvement that the model with $(p, r, d) = (3, 3, 4)$ brings over the structure $(p, r, d) = (1, 3, 4)$ when increasing the radius used for computing the encircled energy. For $\bar{f}$ larger than 0.086, the improvement is larger than 15%. For quasi-static turbulence however, large temporal orders marginally increase the performances over a simple tensor model with $p = 1$. Similarly as highlighted in Section 6.6 with open-loop identification results, increasing the temporal order sharpens the PSF. In closed-loop however, we have observed in the case $d = 2$ that the closed-loop performances may decrease when the temporal order is set too large. As the wind speed is equal to the ones used for identification, such behaviour may be due to the low SNR requiring larger identification batches.

6.8. Two disks rotating in conjugated planes

We now place two disks in conjugated planes rotating at different speeds: one disk rotating at a constant speed of 0.33RPM while the other one varies. The turbulence flow is no longer frozen, and the identification and validation sets are different. The Fried parameter is approximated as equal to half the value of one disk such that D/r_0 is between 10 and 15. It is used to compute the Greenwood frequency based on (1.34). The PI controller now sets $\mathbf{R} = \mathbf{B}^\dagger$ using a truncated SVD. Especially the interaction matrix is not decomposed into the product $\mathbf{GH}$. Figure 6.17 displays the long-exposure Strehl ratio and the Euclidean norm of the residual PSF. The trends observed are very similar to the ones when using a single disk. We note the following differences. The PI controller reaches superior performances compared to

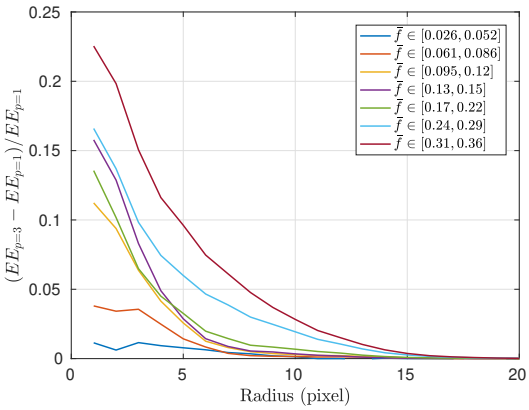


Figure 6.16: Encircled energy as a function of the Greenwood per sample frequency ratio. The relative improvement brought by the case $(p, r, d) = (3, 3, 4)$ over $(p, r, d) = (1, 3, 4)$ is shown.

the proposed method when the Greenwood per sample frequency is low as it does not rely on any wavefront reconstruction, and the performances do no suffer from this static error. Moreover, the least performing model structure is with a tensor order equal to 4. This model structure is less suited to this scenario of turbulence without increasing the Kronecker rank to larger values.

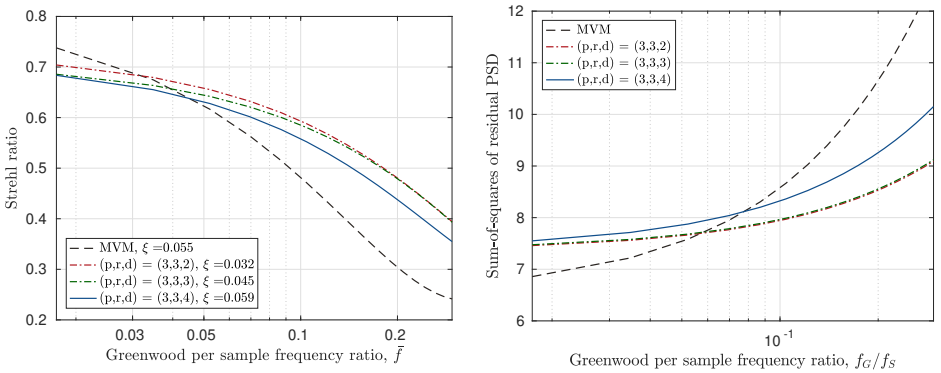


Figure 6.17: PSF measures as a function of the Greenwood per sample frequency ratio. The turbulence consists of two disks in conjugated planes. (a): Strehl ratio. (b): Euclidean norm of the residual PSF as a function of the Greenwood per sample frequency ratio.

6.9. Conclusion

Conclusions

We have proposed a scalable identification algorithm for large-scale adaptive optics. A Vector Autoregressive model on the sensor data is used rather than a state-space model which overcomes the difficulties for handling subspace identification of large-scale systems. The sensor data is shuffled into a tensor, and the coefficient matrices are modelled with a sum of few matrices modelled with a Kronecker product with d terms. Although the case $d = 2$ is optimal for accuracy purposes, larger tensor orders imply larger data compression rates although the larger the prediction error on identification data. Increasing the Kronecker rank however mitigates this latter effect as shown on open-loop sensor data. The impact of prediction on closed-loop performances has been demonstrated on a laboratory testbed.

Regularization for enforcing stable QUARKS models has been presented in Chapter 2 in the case $d = 2$. However, calibration errors to obtain the turbulence-only contribution may add up and deteriorate the performances. It is not particular to the method presented in this chapter but is more general to predictive control methods as pointed out in Guyon and Males (2017).

Recommendations

Further tests

The laboratory testbed experiments demonstrates the basic performance of the algorithm, and not all the possible nuances, and further work could be done to characterize performance in simulation or field conditions under various atmospheric settings. It is especially interesting to compare with Massioni et al. (2011) both in terms of accuracy for the prediction and online computational complexity.

Handling more general noise covariance matrices in the QUARKS

When identifying autoregressive models, the covariance matrix $\mathbf{C}_w$ was assumed equal to a multiple of the identity. However, it is in general not the case. The minimum-variance control problem reads:

$$\min_{\mathbf{u}(k)} \quad \|\mathbf{C}_w^{-1/2}\left(\widehat{\mathbf{s}^{tur}}(k+1|k) + \mathbf{B}\mathbf{u}(k)\right)\|_2^2 \tag{6.19}$$

Standard deformable mirrors have a coupling of $15-20\%$, and hence the matrix $\mathbf{B}$ is multi-banded. The Kronecker structure is all the more interesting with respect to the sparse structure when the original matrix is dense. However, the product $\mathbf{C}_w^{-1/2}\mathbf{B}$ should be highly structured to solve (6.19) with $\mathcal{O}(N^3)$ at most. The problem of estimating the matrix $\mathbf{C}_w^{-1/2}$ is formulated as follows. Let the sequence $\mathbf{w}(k)$ be the residual of the least-squares used for estimating the QUARKS in Chapter 2 when the regularization parameters are set to zero. Let a signal $\mathbf{e}(k)$ be zero-mean white Gaussian with identity covariance matrix. The sequence $\{\mathbf{e}(k)\}_{k\in\{1,\dots,N_t\}}$ is unknown. For all $j \in \{1,\dots,r\}$, let $\mathbf{A}_{1,j}, \mathbf{A}_{2,j} \in \mathbb{R}^{N\times N}$. The estimation of the

whitening-noise matrix boils down to finding $\mathbf{A}_{2,j}$ and $\mathbf{A}_{1,j}$ such that:

$$\sum_{j=1}^{r}(\mathbf{A}_{2,j}\otimes\mathbf{A}_{1,j})\begin{bmatrix}\mathbf{w}(1) & \dots & \mathbf{w}(N_t)\end{bmatrix}=\begin{bmatrix}\mathbf{e}(1) & \dots & \mathbf{e}(N_t)\end{bmatrix} \tag{6.20}$$

Handling more general geometries than the square aperture
In adaptive optics, the circular aperture as well as a central obscuration hampers a practical applicability of the Kronecker models which rely on regular rectangular grids. In the laboratory experiments, the sensor measurements are available on a circular array rather than on a square array. We have proposed to consider the rectangular embedding and set all outputs corresponding to non-illuminated lenslets to zero. It is worth exploring when targeting practical applications how to modify the sums-of-Kronecker structure to avoid that the edges deteriorate the overall estimation. The standard approach is to solve the stochastic least-squares,

$$\min_{\boldsymbol{\phi}(k)} \quad \|\mathbf{M}\big(\mathbf{s}(k)-\mathbf{G}\boldsymbol{\phi}(k)\big)\|_2^2+\lambda\boldsymbol{\phi}(k)^T\mathbf{C}_{\phi}^{-1}\boldsymbol{\phi}(k) \tag{6.21}$$

where the matrix $\mathbf{M}$ selects the valid lenslets actually providing measurements and λ is a regularization parameter depending on the sensor noise.

Let Ω denote the set of known entries in the rectangular embedding $\mathcal{R}$ of the pupil aperture. Let assume that the two-dimensional wavefront (e.g ivec($\boldsymbol{\phi}(k)$), denoted with $\mathbf{W}$) is low-rank. A standard matrix completion problem is formulated as follows:

$$\begin{aligned}\min_{\boldsymbol{\Phi}(k)(i,j)_{(i,j)\in\mathcal{R}\setminus\Omega}} \quad & \|\boldsymbol{\Phi}(k)\|_\star \\ \text{s.t} \quad & \forall(i,j)\in\Omega,\boldsymbol{\Phi}(k)(i,j)=\mathbf{W}(i,j)\end{aligned} \tag{6.22}$$

To cope with the large dimensions, only the missing data is parametrized as variable. An alternative idea is to stack the reconstructed wavefront for N_t temporal samples collected in open-loop in a third-order tensor and decompose the latter with e.g a CPD. The missing entries are estimated based on this sum of rank-one terms decomposition.

7
Conclusions and recommendations

For deriving a prediction for the spatial-temporal dynamics of large and multi-dimensional stochastic systems, we have proposed a data-sparse though multi-linear parametrization of the system matrices and have used it to derive computationally efficient algorithms.

Whether for identifying the coefficient matrices of autoregressive models or state-space matrices, input and output data were reshuffled from vectors to matrices or tensors, which allowed to formulate multi-convex cost functions solved iteratively. The proposed Alternating Least Squares algorithm for identifying QUARKS models showed empirically a globally convergent behaviour although we lack theoretical guarantees. Matrix state space models were introduced and identified from data in a scalable manner. Further assumptions on the system matrices such as strict positivity of the coefficients allowed to scale to larger sizes and with a simpler algorithm. Structure-preserving iterations were presented to solve the discrete Lyapunov equation with $\mathcal{O}(N^3)$ complexity rather than $\mathcal{O}(N^6)$. It is a first step toward estimating a low Kronecker rank Kalman gain solving the DARE.

In the context of adaptive optics, we have proposed to identify offline an autoregressive model to predict online and in a scalable manner the sensor measurements and have validated the approach in a laboratory testbed. Further tests should be carried out on realistic ELT simulations to continue the validation process and on-sky to investigate its ability to adapt to temporally varying conditions.

We now conclude and propose suggestions for further developing the ideas and algorithms presented in this thesis, and deepening the theoretical understanding of certain aspects.

7.1. Conclusions

We have introduced a multi-linear parametrization of the system matrices to model the spatial and temporal dynamics of stochastic LTI systems that are spatially distributed on multiple dimensions. The decomposition into Kronecker products is especially suitable when the underlying multi-variable function is separable in its coordinates. When this property does not hold exactly, the Kronecker rank makes a balance between the accuracy of the representation (which is maximized in an unstructured setting), the data compression and the scalability of the algorithms. Its role is comparable to the one of the underlying system order in SSS matrices. This trade-off is a fortiori competitive with respect to the sparse multi-banded structure when the true matrix is dense. Spatial invariance is not assumed.

We have started the analysis with a focus on the identification of large-scale auto-regressive models. The standard vector of measurements is reshuffled into a matrix and the cost function is a bilinear least-squares with rN^2 variables instead of N^4 for a sensor array of size $N \times N$. Regularization to enforce temporal stability or a decay in the factor matrices was incorporated without altering the convergence to the global minimum of the Alternating Least Squares as observed numerically. Recursive algorithms can be combined with the QUARKS without compromising on the accuracy of the solution hence enabling first, a reduced memory storage, and second, to deal with non-stationary data.

When state-space matrices are written with a single Kronecker product, a class of matrix state-space models was introduced. After having analyzed the observability and controllability, and characterized similarly equivalent systems, we have proposed a subspace-like algorithm to identify the factor matrices. It consists of three steps, one of which consists of minimizing a low-rank cost function subject to bilinear constraints. The methodology shares similarities with the PBSID framework. Although its reduced computational performances allows handling much larger dimensions than the standard algorithms, a decrease in accuracy with respect to the identification of unstructured matrices is observed and is caused by the non-globally convergent algorithm proposed.

For all standard linear algebra operations, assuming a decomposition of the coefficient matrices with a single Kronecker product of two terms implies a lower-bound on the achievable computational complexity, equal to $\mathcal{O}(N^3)$ for an array of size $N \times N$. A linear computational complexity with respect to the number of nodes as e.g would be obtained when the nodes are decoupled could not be reached this way, and a parametrization with a product of more Kronecker products than only two was studied. Its close relationship with tensors allowed to derive more efficient algorithms reaching asymptotically with the tensor order $\mathcal{O}(N^2)$ complexity. The first tensor orders provide already with most of the computational improvements without significantly losing accuracy as demonstrated in the laboratory experiments. The subclass of externally strictly positive and Kronecker-structured systems was assumed in order to improve the accuracy and computational cost of a critical step in K4SID.

In some applications such as adaptive optics where the state has a physical meaning and can be estimated - the wavefront-, it is common to derive the system

matrices without resorting to subspace identification and subsequently solve the DARE to compute the Kalman gain. We have made a first step toward solving the DARE when the matrices are low-Kronecker rank which would allow to relax some convervative assumptions used in Massioni et al. (2011). For example, we would like to assume $\mathbf{A} = \mathbf{A}_2 \otimes \mathbf{A}_1$ as in Yu and Verhaegen (2018b) (or a sum of few terms) instead of $\mathbf{A} = a\mathbf{I}_{N^2}$ in order to account for some spatial-temporal coupling. The factors would be identified from data, and the DARE subsequently solved. Inspired by the structure-preserving iterations derived with the SSS structure in Rice (2010), we adapted a doubling algorithm, the squared Smith iteration, for solving discrete Lyapunov equations tailored for maintaining the low-Kronecker rank parametrization. The larger the Kronecker rank of the solution, the less the errors accumulate while iterating and the better the final approximation although the computational assets then deteriorates cubically with the Kronecker rank. Reshuffling the Lyapunov equation yields a discrete Sylvester equation with low-rank factors, which allowed to derive a variant of an existing factored ADI method. Many works have observed that if the matrix $\mathbf{Q}$ is low-rank, the solution is also low-rank, remarkably leading to exactly the same conclusions for low-Kronecker rank matrices. If the matrix $\mathbf{Q}$ is low-Kronecker rank, the solution is low-Kronecker rank, and we have exploited a structured $\mathbf{A}$ to solve efficiently linear systems of equations in the fADI.

The QUARKS models were implemented on a laboratory testbed to demonstrate its applicability to large-scale adaptive optics. Especially, we have shown that despite losing performance because of structuring the matrix, it reduces significantly the temporal error for large Greenwood per sample frequency ratio compared to the non-predictive methods. Deriving a prediction from autoregressive models solving a simple least-squares from open-loop data is the most robust method of the three described above (including subspace identification and estimating a Kalman gain), especially since it does not require wavefront reconstruction and because the proposed algorithm converges to the global minimum.

Table 7.1 summarizes the contributions relative to the study of the low-Kronecker rank structure.

Table 7.2 focuses on their assets and downsides. Some comments are particularly related to adaptive optics. The process and measurement noises are zero mean white Gaussian, uncorrelated and with covariances respectively $\mathbf{Q}$ and $\sigma_e^2\mathbf{I}$.

Identification of autoregressive models [**Chapter 2**]	Solved using ALS starting from random initial guesses. Complexity of $\mathcal{O}(N^3 N_t)$ compared to $\mathcal{O}(N^6)$ when unstructured. A recursive alternative is proposed to handle time-varying behaviours.
Identification of state-space models [**Chapters 3,4**]	1. Identify the Markov parameters using the QUARKS 2. Solve a low-rank minimization problem subject to bilinear constraints 3. Either estimate the state sequence with two SVDs and subsequently the factored state-space matrices, or form low-rank block-Hankel matrices and do the estimation using classical realization theory.
Solve the discrete Lyapunov equation [**Chapter 5**]	1. With the squared Smith's iteration, and preserving the structure at each iteration by truncating the Kronecker rank. 2. Reshuffling the equation to form a discrete Sylvester equation and derive a tailored factored ADI algorithm to exploit the low-rank structure.
Approximate the inverse Varnai (2017)	Proposed using ALS. May be combined with Newton's iteration to refine the estimates.

Table 7.1: Summary of the algorithms that have been derived to benefit from the low-Kronecker rank structure.

Assuming spatial invariance and infinite supports, Bamieh et al. (2002), Massioni et al. (2011)	Exploits the localizability of the Kalman gain for spatially-invariant systems and relies on a Fourier decomposition of an infinitely large state and sensor to decouple the modes. Highly parallelizable. The quality of the estimation decreases at the edges.
QUARKS [**Chapters 2,4,6**]	$\mathbf{y}(k) = \sum_{i=1}^{p} \sum_{j=1}^{r} \big(\mathbf{M}_{i,j,d} \otimes \ldots \otimes \mathbf{M}_{i,j,1}\big)\mathbf{y}(k-i) + \mathbf{v}(k)$ Minimizes the residual slopes. Autoregressive model with coefficient matrices estimated from data, and possibly updated online. Global convergence and the Kronecker structure is more robust to outliers. The covariance of $\mathbf{v}(k)$ is assumed equal to the identity. Measurements available on a square aperture are assumed.
Subspace identification [**Chapters 3,4**]	$\begin{aligned} \mathbf{X}(k+1) &= \mathbf{A}_1 \mathbf{X}(k) \mathbf{A}_2^T + \mathbf{K}_1 \mathbf{E}(k) \mathbf{K}_2^T \\ \mathbf{Y}(k) &= \mathbf{C}_1 \mathbf{X}(k) \mathbf{C}_2^T \end{aligned}$ Minimizes the residual wavefront if written as in Hinnen (2007). Data-driven method of computing the Kalman gain. Assumption of Kronecker rank one.
Kronecker-structured DARE [**Chapter 5**]	$\begin{aligned} \mathbf{\Phi}^{tur}(k+1) &= \mathbf{A}_1 \ \mathbf{\Phi}^{tur}(k) \mathbf{A}_2^T + \mathbf{W}(k) \\ \mathbf{Y}(k) &= \sum_{j=1}^{2} \mathbf{G}_{1,j} \mathbf{\Phi}^{tur}(k) \mathbf{G}_{2,j}^T + \mathbf{E}(k) \end{aligned}$ where the covariance of $\mathrm{vec}(\mathbf{W}(k))$ is low-Kronecker rank. Models spatial-temporal coupling and better suited to large Greenwood per sample frequency ratio. Efficient algorithm for computing discrete Lyapunov equations and approximate inverses. Lack of understanding when the structure is preserved via Newton-Hewer's iterations solving the DARE.

Table 7.2: Assets and drawbacks of the low-Kronecker rank predictive algorithms for large-scale systems.

7.2. Recommendations

In this section, we highlight research questions which have not been answered in this dissertation although for which the material presented in previous chapters is a starting point. These are summarized in Table 7.3. More specific comments that are very much chapter-related have already been made after each chapter.

QUARKS [**Chapters 2,4,6**]	Compare with Massioni et al. (2011) on AO simulators dedicated to large sensors. Test on-sky for the analysis of temporal variations. Extend to Laser Tomography AO.
Subspace identification [**Chapters 3,4**]	Extend to larger Kronecker ranks: $\mathbf{X}(k+1) = \mathbf{A}_1\mathbf{X}(k)\mathbf{A}_2^T + \mathbf{K}_1\mathbf{E}(k)\mathbf{K}_2^T$ $\mathbf{Y}(k) = \sum_{j=1}^{r} \mathbf{C}_{j,1}\mathbf{X}(k)\mathbf{C}_{j,2}^T + \mathbf{E}(k)$ Combine with an output error algorithm with $\mathcal{O}(N^3 N_t)$ complexity. Test on laboratory data.
Kronecker-structured DARE [**Chapter 5**]	Understand when the Kalman gain is low-Kronecker rank. Assume spatial-invariance and translate into structural properties to obtain further speed up.

Table 7.3: Recommendations.

7.2.1. The question of circular apertures

A Kronecker formulation of the models is best suited for systems with a rectangular regular sensor grid, which is not the case for AO systems for extremely large telescopes that have circular apertures and a significant central obscuration. It is not yet quite clear what precisely the impact of the edges is on the global performance and extrapolating the data on a square embedding as suggested at the end of Chapter 6 might impact the performance. Rather than working with a Kronecker product, which is only suited to model block matrices with blocks of equal size, it appears possible to *define* a tailored product between factor matrices that would adapt to the particular block structure of the matrices, see Figure 7.1. The only required assumption is the separability of the underlying function in the horizontal and vertical directions, and such a new product would be essentially defined in two steps: first the sums of Kronecker parametrization as studied in this thesis, and second,

the selection of the valid indices corresponding to active measurements. It would be interesting to investigate the structure-preserving capabilities of such models.

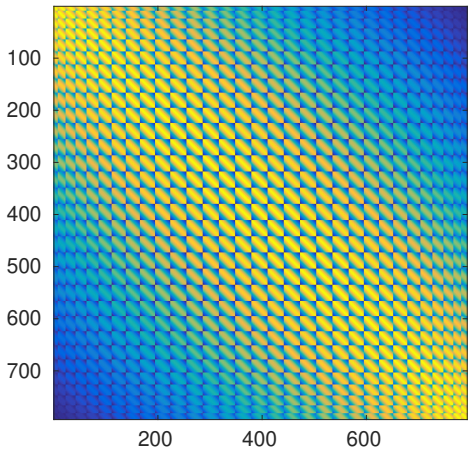


Figure 7.1: Example of block matrix when the actuators and sensors available only on a circular aperture.

7.2.2. Identification algorithms for state-space models

We have laid some foundations for susbpace identification of tensor state-space models in Chapter 3 and 4, but there is room for improvement in the performances as we have seen that the optimization problems at stake are challenging (at least as currently formulated). It cannot be applied in a satisfactory manner to real-life data, both because a Kronecker-rank one structure was assumed and because K4SID is not globally convergent. We have proposed an implementation of the low-rank problem with bilinear constraints following the lines in Doelman and Verhaegen (2016) in the toolbox. It is the most promising approach to extend the current work for larger Kronecker ranks. A non-linear output-error algorithm to refine the estimates obtained from K4SID would be relevant only if its complexity is less or equal than $\mathcal{O}(N^3 N_t)$.

7.2.3. The Kronecker-structured DARE

When the system matrices are low-Kronecker rank, a structured Kalman gain may be derived maintaining the structure throughout the iterations. The assets in an AO context are not only that the $\mathbf{A}$ matrix is allowed to be dense (although possibly banded) and hence, express better the spatial-temporal dynamics, but most importantly, there is no assumption of spatial invariance.

Solving the DARE using the Newton's or Newton-Hewer's iterations involve mainly solving a discrete Lyapunov equation and inverting a matrix. Approximate inverses for low-Kronecker rank matrices have been studied in Varnai (2017). All

ingredients for structure-preserving Newton-Hewer's iterations now allow a computationally efficient computation of the Kalman gain. Nonetheless, Varnai (2017) illustrates that the Kronecker rank is likely to increase significantly when the factor matrices are random. Conditions on the factor matrices such that the inverse is well-approximated with few Kronecker terms is essential to have some understanding on the required properties that the system should have. This question is very much related to the approximation of the Kalman gain with a low-Kronecker rank matrix. An alternative for computing the inverse is the Newton's method, which was applied in the same context for preserving a structure in Olshevsky et al. (2008) and Haber and Verhaegen (2018).

When the Kronecker rank is set too low, the observer might be made unstable. Gradually increasing it yields first stability, then improved performance. It is of interest to quantify the Kronecker rank yielding a stable observer, or a given performance level. In practice, it appears when the control frequency only allows a maximum computational complexity. This question relates in its methodology to finding the lowest order of the SSS representation yielding a stable observer/controller.

For dealing with possibly non-stationnary disturbances flowing over a fourty-meter wide aperture at a control frequency of 500Hz as for large telescopes, we believe the matrices describing the plant and the disturbance should be continuously updated from data while the loop is closed. We have proposed in this thesis to update recursively autoregressive models. In this direction, a Kronecker-structured Kalman gain recursively updated would adapt to the current dynamics above the telescope.

7.2.4. Parametrizing the factors

The dimension of a system (i.e the dimension of the sensor array) was allowed to be different from the dimension of the model, d. In other words, a sensor array of size $N \times N$ could be modelled with matrices belonging to $\mathcal{K}_{d,r}$ to reduce the computational complexity of the algorithms from $\mathcal{O}(N^3)$ to $\mathcal{O}(N^2)$ when increasing d. Instead, we should think of structuring the Kronecker factors. For spatial-invariant systems, the factors of the Kronecker sums would be Toeplitz. When the size of the sensor is finite, a most adequate representation is rather the Almost Toeplitz SSS structure as introduced in Rice (2010) to deal with the boundary conditions. It is clear how to use both of these for matrix-matrix multiplications and, when $r = 1$, for matrix inversions. The question whether the Kronecker and SSS structure could be *simultaneously* maintained when approximating the inverse or throughout Newton's iterations for computing the DARE has not been addressed.

Parametrizing the factors with further structure means that we study the dynamics of the subsystems at a local scale. An illustration is given for autonomous systems. Let $\mathbf{x}(0) = \mathbf{x}^h(0) \otimes \mathbf{x}^v(0)$, and the state update equation

$$\mathbf{x}(k+1) = \left(\bar{\mathbf{A}}^h \otimes \bar{\mathbf{A}}^v\right)\mathbf{x}(k) \tag{7.1}$$

such that the global state is, $\mathbf{x}(k) = \mathbf{x}^h(k) \otimes \mathbf{x}^v(k)$. The equation (7.1) is decoupled into two independent *virtual* autonomous systems that are associated with the

horizontal and vertical direction,

$$\begin{cases} \mathbf{x}^h(k+1) &= \bar{\mathbf{A}}^h \mathbf{x}^h(k) \\ \mathbf{x}^v(k+1) &= \bar{\mathbf{A}}^v \mathbf{x}^v(k) \end{cases} \tag{7.2}$$

The Kronecker decomposition breaks away from a local view of the subsystems, and viewing the 2D grid with local subsystems on each node is erroneous. This differs from the 1D SSS, and its extensions to multi-level SSS for which it was proposed to model the dynamics of each subsystems with its own set of generators, Yu et al. (2018b), or to consider each horizontal string as a subsystem of the vertical string, Rice (2010). It is as if there are two virtual strings of subsystems, each associated with a direction of the grid. The disturbance, control inputs and measurement are located on the grid, and computing the state associated to a given node of the grid is done by propagating the horizontal and vertical states along its respective SSS string with a forward-backward pass, as illustrated in Figure 7.2.

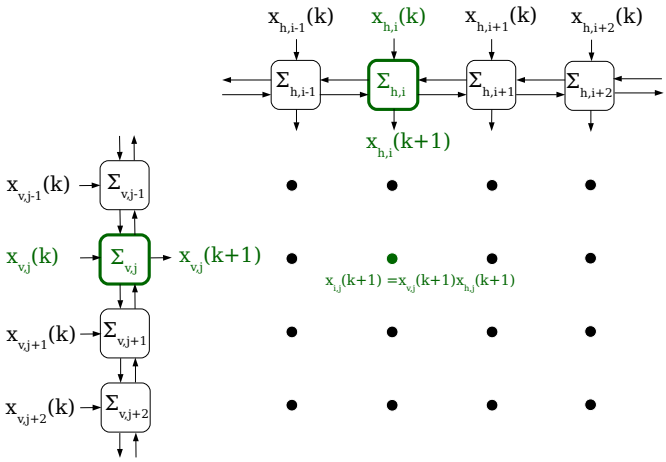


Figure 7.2: Schematic illustrating the virtual strings of interconnected systems each associated with a direction of the grid for performing the matrix-vector multiplication $\mathbf{x}(k+1) = (\bar{\mathbf{A}}_h \otimes \bar{\mathbf{A}}_v)\mathbf{x}(k)$. Only the simplest cast $r = 1$ is depicted. The state at each position of the grid is obtained by multiplying the horizontal and vertical (virtual) states obtained with a forward and backward pass on the SSS string.

7.2.5. Assuming another tensor decomposition of the reshuffled matrix

We conclude this dissertation by pondering on the structure chosen to parametrize the global system matrices which may not be optimal for deriving structure-preserving algorithms. Low-Kronecker rank matrices were shown to have an underlying tensor

representation in Chapter 4. We have assumed a parametrization of the system matrices and then, have related to the reshuffled version. What about starting from a tensor corresponding to the same reshuffling of the data that we have introduced, and then decomposing it using a Tensor Train (TT) decomposition, which have been introduced in Oseledets and Dolgov (2012), to finally discover the equivalent formulation of the global matrix? A particular rank is defined for such matrices. Particularly, the rank of the sum of two TT matrix is the sum of the rank of each matrix, the rank is multiplied when multiplying TT matrices. It is however still an open question whether an efficient representation of such structure is maintained after inversion.

7.2.6. Communication scheme and the implementation

At each time sample, computing a state-update with low-Kronecker system matrices requires to collect the full vector of sensor measurements on a central computational unit. Especially for large-scale Kalman filtering, the model structure used is such that the communication may be a bottleneck in the closed-loop. For example in AO, the Shack-Hartmann camera transfers the image to the host device (CPU memory) and the slopes measurements may then be loaded on a GPU. It is an open question to choose the computing platform best suited to the Kronecker-structured Kalman filters. The overall architecture remains to be studied in the context of the structured matrices proposed in this thesis, especially to investigate how the computational improvements theoretically obtained would translate in practice.

Appendices

A

The Kronecker product

In this appendix, we recall the main computational rules associated with the Kronecker product. The Kronecker product between two matrices $\mathbf{B} \in \mathbb{R}^{m \times n}$ and $\mathbf{C} \in \mathbb{R}^{p \times q}$ is defined as:

$$\mathbf{A} = \mathbf{B} \otimes \mathbf{C} = \begin{bmatrix} b_{11}\mathbf{C} & b_{12}\mathbf{C} & \dots & b_{1n}\mathbf{C} \\ b_{21}\mathbf{C} & b_{22}\mathbf{C} & \dots & b_{2n}\mathbf{C} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1}\mathbf{C} & b_{m2}\mathbf{C} & \dots & b_{mn}\mathbf{C} \end{bmatrix} \tag{A.1}$$

The result is a matrix $\mathbf{A} \in \mathbb{R}^{mp \times nq}$. Decomposing $\mathbf{A}$ with the factors $\mathbf{B}$ and $\mathbf{C}$ is not unique. There exists a non-zero scalar γ such that:

$$\mathbf{A} = \mathbf{B} \otimes \mathbf{C} = \gamma\mathbf{B} \otimes \frac{1}{\gamma}\mathbf{C} \tag{A.2}$$

The Kronecker product is distributive and associative:

$$\mathbf{D} \otimes (\mathbf{B} + \mathbf{C}) = \mathbf{D} \otimes \mathbf{B} + \mathbf{D} \otimes \mathbf{C} \tag{A.3}$$

$$\mathbf{D} \otimes (\mathbf{B} \otimes \mathbf{C}) = (\mathbf{D} \otimes \mathbf{B}) \otimes \mathbf{C} \tag{A.4}$$

As a direct consequence of associativity, the following equality holds:

$$(\mathbf{B}_1 \otimes \mathbf{C}_1)(\mathbf{B}_2 \otimes \mathbf{C}_2) = (\mathbf{B}_1\mathbf{B}_2) \otimes (\mathbf{C}_1\mathbf{C}_2) \tag{A.5}$$

The transpose is computed from its factors:

$$(\mathbf{B} \otimes \mathbf{C})^T = \mathbf{B}^T \otimes \mathbf{C}^T \tag{A.6}$$

The trace of a product of Kronecker is equal to the product of the traces:

$$\text{Trace}(\mathbf{A} \otimes \mathbf{B}) = \text{Trace}(\mathbf{A})\text{Trace}(\mathbf{B}) \tag{A.7}$$

If the factor matrices are square and invertible, the inverse of $\mathbf{A}$ exists and is given by:

$$(\mathbf{B} \otimes \mathbf{C})^{-1} = \mathbf{B}^{-1} \otimes \mathbf{C}^{-1} \tag{A.8}$$

The same relation holds with the pseudo-inverse when the factor matrices are not square.
Inner products also boil down to operations on the factor matrices:

$$\langle \mathbf{B}_1 \otimes \mathbf{C}_1, \mathbf{B}_2 \otimes \mathbf{C}_2 \rangle = \langle \mathbf{B}_1, \mathbf{B}_2 \rangle \langle \mathbf{C}_1, \mathbf{C}_2 \rangle \tag{A.9}$$

This property is shown from (A.7). The more general Kronecker product between d factor matrices has similar properties. The proof is done by induction.

The Kronecker product behaves nicely when it comes to preserving the matrix structure from the factors on the global matrix. A few properties are highlighted in the table A.1. However, the properties are often one-sided and the reverse side is in general not true.

if B and C are both ...	**then $\mathbf{B} \otimes \mathbf{C}$ is ...**
banded	multi-banded
Toeplitz	two-level Toeplitz
SSS	two-level SSS
symmetric	symmetric
positive definite	positive definite
orthogonal	orthogonal

Table A.1: Preserving the structures from the factors to the global matrix

Given the matrices $\mathbf{A} \in \mathbb{R}^{m \times n}, \mathbf{B} \in \mathbb{R}^{n \times p}, \mathbf{C} \in \mathbb{R}^{p \times q}$, we have the following relation:

$$\text{vec}(\mathbf{ABC}) = (\mathbf{C}^T \otimes \mathbf{A})\text{vec}(\mathbf{B}) \tag{A.10}$$

The Khatri-Rao product $\odot$ between two matrices with equal number of columns n is the column-wise Kronecker product:

$$\mathbf{A} = \mathbf{B} \odot \mathbf{C} = \begin{bmatrix} \mathbf{b}_1 \otimes \mathbf{c}_1 & \dots & \mathbf{b}_n \otimes \mathbf{c}_n \end{bmatrix} \tag{A.11}$$

More details on the computational rules related to the Kronecker product are found in van Loan (2000).

B
Fundaments on tensors

Let d, I, J three integers. Let $(I_1, \ldots, I_d)$ and $(J_1, \ldots, J_d)$ be two tuples of integers such that $\prod_{j=1}^{d} I_j = I$ and $\prod_{j=1}^{d} J_j = J$.

Definition B.1. *For $j \in \{1, \ldots, d\}$, let i_j an integer such that $1 \leq i_j \leq I_j$. A vector $\mathbf{x} \in \mathbb{R}^I$ is tensorized into $\mathcal{X} \in \mathbb{R}^{I_1 \times \ldots \times I_d}$ using the elementwise relationship:*

$$x_{i_1,\ldots,i_d} = \mathrm{x}_{\overline{i_1 \ldots i_d}} \tag{B.1}$$

where $\overline{i_1 \ldots i_n} = i_1 + (i_2 - 1)I_1 + \ldots + (i_N - 1)I_1 \ldots I_d$.

This reshuffling operation consists of re-indexing the elements in $\mathbf{x}$ with a tuple $i_1, \ldots, i_d$ instead of one index.

Example B.1. *We illustrate the reshuffling operation with a small-scale example and choose $I = 16, d = 3$. Let a vector $\mathbf{x} \in \mathbb{R}^{16}$ be defined with $x_i = i$ for $i \in \{1, \ldots, 16\}$. Its reshuffling into a tensor $\mathcal{X} \in \mathbb{R}^{2 \times 4 \times 2}$ is such that:*

$$\mathcal{X}_{:,:,1} = \begin{bmatrix} 1 & 3 & 5 & 7 \\ 2 & 4 & 6 & 8 \end{bmatrix}, \quad \mathcal{X}_{:,:,2} = \begin{bmatrix} 9 & 11 & 13 & 15 \\ 10 & 12 & 14 & 16 \end{bmatrix} \tag{B.2}$$

The reshuffling into a tensor of order 3 is not unique: the tensor $\mathcal{X} \in \mathbb{R}^{4 \times 2 \times 2}$ can also be formed:

$$\mathcal{X}_{:,:,1} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{bmatrix}^T, \quad \mathcal{X}_{:,:,2} = \begin{bmatrix} 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix}^T \tag{B.3}$$

Tensors may be transformed into matrices to help carry out efficient linear algebra operations.

Definition B.2. *The n-mode matricization of a tensor $\mathcal{X} \in \mathbb{R}^{J_1 \times \ldots \times J_d}$ is denoted with $\mathbf{X}_{(n)}$ and belongs to $\mathbb{R}^{J_n \times J_1 \ldots J_{n-1} J_{n+1} \ldots J_d}$. It is equal to:*

$$\begin{bmatrix} x_{1,\ldots,1,1,1,\ldots,1} & x_{2,\ldots,1,1,1,\ldots,1} & \cdots & x_{J_1,\ldots,J_{n-1},1,J_{n+1},\ldots,J_d} \\ x_{1,\ldots,1,2,1,\ldots,1} & x_{2,\ldots,1,2,1,\ldots,1} & & x_{J_1,\ldots,J_{n-1},2,J_{n+1},\ldots,J_d} \\ \vdots & \vdots & & \vdots \\ x_{1,\ldots,1,J_n,1,\ldots,1} & x_{2,\ldots,1,J_n,1,\ldots,1} & \cdots & x_{J_1,\ldots,J_{n-1},J_n,J_{n+1},\ldots,J_d} \end{bmatrix} \tag{B.4}$$

A more intuitive way of matricizing a tensor is to introduce the notation of fibers.

Definition B.3. *Using the notations used above, a n-mode fiber is a column vector that contains the elements* $\mathcal{X}_{j_1,\ldots,j_{n-1},:,j_{n+1},\ldots,j_d}$.

For a second order tensor, i.e a matrix, a 1-mode fibre is a column while a 2-mode fibre is a row. The n-mode matricization is formed by reshuffling the n-mode fibers to be the columns of the matrix $\mathbf{X}_{(n)}$ and is illustrated in Figure B.1.

Example B.2. *For the tensor* $\mathcal{X} \in \mathbb{R}^{2\times4\times2}$ *mentioned in Example 1.1, we have:*

$$\mathbf{X}_{(1)} = \begin{bmatrix} 1 & 3 & 5 & 7 & 9 & 11 & 13 & 15 \\ 2 & 4 & 6 & 8 & 10 & 12 & 14 & 16 \end{bmatrix} \tag{B.5}$$

$$\mathbf{X}_{(2)} = \begin{bmatrix} 1 & 2 & 9 & 10 \\ 3 & 4 & 11 & 12 \\ 5 & 6 & 13 & 14 \\ 7 & 8 & 15 & 16 \end{bmatrix} \tag{B.6}$$

$$\mathbf{X}_{(3)} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 \end{bmatrix} \tag{B.7}$$

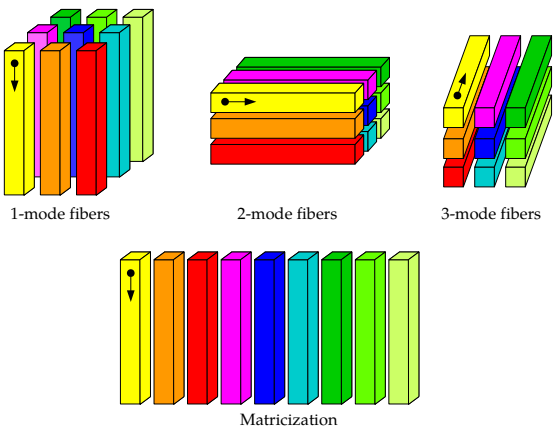


Figure B.1: On the upper level, a third order tensor of size $3 \times 3 \times 3$ is represented in a three-dimensional space. It appears three times with different colors although the entries in the tensor are identically positioned. The fibers along different modes consists of rectangle boxes that are colored to indicate how each set of n-mode fibers are matricized as depicted below. The arrow relates to the direction of the fibers used for matricizing.

The n-mode matricization is used for computing n-mode matrix products. We first state the definition before mentioning equivalences that will be used in the sequel.

Definition B.4. *Let* $I \in \mathbb{N}$. *The n-mode matrix product of a tensor* $\mathcal{X} \in \mathbb{R}^{J_1\times\ldots\times J_d}$ *with a matrix* $\mathbf{M} \in \mathbb{R}^{I\times J_n}$ *is denoted* $\mathcal{X} \times_n \mathbf{M}$. *The result is of size* $J_1 \times \ldots J_{n-1} \times I \times J_{n+1} \times \ldots \times J_d$ *and is defined elementwise with:*

$$(\mathcal{X} \times_n \mathbf{M})_{j_1,\ldots,j_{n-1},i,j_{n+1},\ldots,j_d} = \sum_{j_n=1}^{J_n} x_{j_1,\ldots,j_d} m_{i,j_n} \tag{B.8}$$

It is often more insightful to introduce the n-mode matrix product by relating to the n-mode matricization.

Proposition B.1. *Let* $(\mathcal{X}, \mathcal{Y}) \in \mathbb{R}^{J_1 \times \ldots \times J_d} \times \mathbb{R}^{I_1 \times \ldots \times I_d}$. *If* $\mathbf{M} \in \mathbb{R}^{I_n \times J_n}$, *then* $\mathcal{Y} = \mathcal{X} \times_n \mathbf{M}$ *is equivalently written with* $\mathbf{Y}_{(n)} = \mathbf{M}\mathbf{X}_{(n)}$.
Moreover, for $j \in \{1, \ldots, d\}$ *and a sequence of matrices* $\mathbf{M}_j \in \mathbb{R}^{I_j \times J_j}$, *we define:* $\bar{\mathbf{M}} := \mathbf{M}_d \otimes \ldots \otimes \mathbf{M}_{n+1} \otimes \mathbf{M}_{n-1} \otimes \ldots \otimes \mathbf{M}_1$. *Then, the three following equalities are equivalent:*

$$\mathcal{Y} = \mathcal{X} \times_1 \mathbf{M}_1 \times_2 \ldots \times_d \mathbf{M}_d \tag{B.9}$$

$$\mathbf{Y}_{(n)} = \mathbf{M}_n \mathbf{X}_{(n)} \bar{\mathbf{M}}^T \tag{B.10}$$

$$vec(\mathbf{Y}_{(n)}) = \left(\bar{\mathbf{M}} \otimes \mathbf{M}_n\right) vec(\mathbf{X}_{(n)}) \tag{B.11}$$

These relationships highlight that the n-mode matrix product can be recast into a matrix-matrix multiplication that relates two matricized tensors. For $d = 2, n = 1$, it boils down to the well-known equivalence between $\mathbf{Y}_{(1)} = \mathbf{M}_1 \mathbf{X}_{(1)} \mathbf{M}_2^T$ and $\text{vec}(\mathbf{Y}_{(1)}) = (\mathbf{M}_2 \otimes \mathbf{M}_1)\text{vec}(\mathbf{X}_{(1)})$. It highlights the vectorized form featuring Kronecker products and relates to the model structured studied in the chapters 2 and 3 for $d = 2$. The matrix $\mathbf{M}_2 \otimes \mathbf{M}_1$ may be large, and it is thus more efficient both memory-wise and in terms of computations to use the n-mode matrix product.

Example B.3. *We illustrate the 1-mode matrix product with an example in the case* $d = 3$ *and the tensor* $\mathcal{X}$ *that we have used in this section. Let the matrix* $\mathbf{M}$ *be:*

$$\mathbf{M}^T = \begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix} \tag{B.12}$$

Then:

$$\mathcal{Y}_{:,:,1} = \begin{bmatrix} 5 & 11 & 17 & 23 \\ 11 & 25 & 39 & 53 \\ 17 & 39 & 61 & 83 \end{bmatrix}, \mathcal{Y}_{:,:,2} = \begin{bmatrix} 29 & 35 & 41 & 47 \\ 67 & 81 & 95 & 109 \\ 105 & 127 & 149 & 171 \end{bmatrix} \tag{B.13}$$

The advantages are essentially computation- and memory- related. For $\mathbf{M} \in \mathbb{R}^{I \times J}$ parametrized with d Kronecker products as in Prop.B.1 and $\mathbf{x} \in \mathbb{R}^J$, multiplying $\mathbf{M}$ with $\mathbf{x}$ using n-mode matrix products instead of the vector form leads to a reduced computational cost which is useful for real-time control. The matrix-vector product $\mathbf{M}\mathbf{x}$ is equivalently written with $\mathcal{X} \times_1 \mathbf{M}_1 \times_2 \ldots \times_d \mathbf{M}_d$. The cost for computing a n-mode matrix product $\mathcal{X} \times_n \mathbf{M}_n$ is $I_n J$. It follows that the operation $\mathcal{X} \times_1 \mathbf{M}_1 \times_2 \ldots \times_d \mathbf{M}_d$ costs $J \sum_{j=1}^{d} I_j$. When all $I = J$, and I_j is independent of j, matrix-vector multiplication using a n-mode matrix product costs $dI^{(d+1)/d}$ whose exponent converges to one with increasing d.

Definition B.5. *Hitchcock (1927) The rank of a tensor* $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ *is defined as the minimum number of rank-1 tensors in a polyadic decomposition of* $\mathcal{X}$.

Definition B.6. *Carroll and Chang (1970) Harshman (1970) A canonical polyadic decomposition (CPD) of a third-order tensor* $\mathcal{X}$ *expresses* $\mathcal{X}$ *as a minimal sum of rank-1 terms.*

Figure B.2 illustrates the decomposition of a third-order tensor into a sum of rank-one terms.

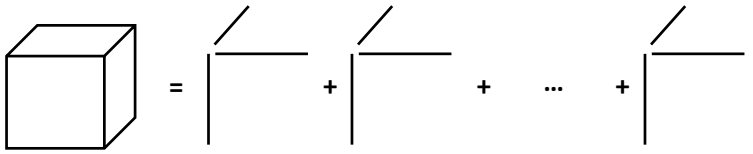


Figure B.2: Schematic of a CPD decomposition for a third-order tensor.

Lemma B.1. *Any real-valued tensor can be decomposed* exactly *with a CPD.*

Proof. Let $\mathcal{X} \in \mathbb{R}^{I \times \ldots \times I}$. Let $\mathbf{e}_i \in \mathbb{R}^I$ with only zero entries but a one at the position i. The set $\mathcal{S} = \{\mathbf{e}_{i_1} \circ \ldots \circ \mathbf{e}_{i_d} / i_j \in \{1, \ldots, I\}, j \in \{1, \ldots, d\}\}$ is the basis of the space containing all real-valued tensors of order d and dimension $I \times \ldots \times I$. An alternative way to see this is to work in $\mathbb{R}^{I^d}$ and replace the outer product with the Kronecker product. The rank is equal to the minimum dimension of the subspace containing $\mathcal{X}$ and formed from linear combination of vectors in $\mathcal{S}$. Let $\boldsymbol{\alpha} \in \mathbb{R}^{I \times \ldots \times I}$. $\mathcal{X}$ is then written as a linear combination of rank-one tensors:

$$\mathcal{X} = \sum_{i_1=1}^{I} \ldots \sum_{i_d=1}^{I} \alpha_{i_1,\ldots,i_d} \mathbf{e}_{i_1} \circ \ldots \circ \mathbf{e}_{i_d} \tag{B.14}$$

■

This lemma is totally different from the numerical procedure to find the optimal rank approximation: it only states that there exists a decomposition into a sum of rank one terms which yields exactly the tensor and is not concerned with the NP-hard problem of finding this rank.

The problem of best rank approximation is solved for matrices with the Eckart-Young theorem, Eckart and Young (1936), which states that, for a matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$, if its SVD is given as:

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V} = \sum_{i=1}^{rank(\mathbf{A})} \sigma_i \mathbf{u}_i \circ \mathbf{v_i} \tag{B.15}$$

then a best rank-r approximation is given by the first r terms in the above sum. Moreover, when r is equal to the minimum between the number and rows and the number of columns, the SVD decomposition is exact. The situation is different for tensors. In fact, computing the rank of a tensor is NP-hard, Håstad (1990). Moreover, the factors in the rank-2 CPD of a cubical tensor may not be factors of the rank-3 CPD as illustrated in Harshman (2004). Importantly, it may happen that a rank-r tensor can be approximated arbitrarily well with a tensor of rank strictly lower than r. In other words, the set of tensors whose rank is at most r is not closed, de Silva and Lim (2008). In such cases, the factors of the latter tensor diverge and

cancel each other while the cost function keeps on decreasing. This degeneracy was shown in de Silva and Lim (2008) and may appear when the CPD is not a proper model structure for the tensor, for example the data is too noisy.

The uniqueness of the CPD is studied in Kruskal (1977) and Domanov and De Lathauwer (2013). A more detailed presentation of tensors is found in Kolda and Kolda and Bader (2009) and examples of tensor decompositions in the TensorLab toolbox, Vervliet et al. (2016).

Bibliography

Akaike, H. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.

Ali, M., Ali, A., Chughtai, S. S., and Werner, H. Consistent identification of spatially interconnected systems. In *Proceedings of the 2011 American Control Conference*, pages 3583–3588, 2011.

Baldi, P. and Hornik, K. Neural networks and principal component analysis: Learning from examples without local minima. *Neural Netw.*, 2(1):53–58, 1989.

Bamieh, B. The structure of optimal controllers of spatially-invariant distributed parameter systems. In *Proceedings of the 36th IEEE Conference on Decision and Control*, volume 2, pages 1056–1061 vol.2, Dec 1997.

Bamieh, B., Paganini, F., and Dahleh, M. A. Distributed control of spatially invariant systems. *IEEE Transactions on Automatic Control*, 47(7):1091–1107, July 2002.

Bamieh, B. and Voulgaris, P. G. A convex characterization of distributed control problems in spatially invariant systems with communication constraints. *Systems & Control Letters*, 54(6):575 – 583, 2005.

Bartels, R. H. and Stewart, G. W. Solution of the matrix equation $ax + xb = c$. *Commun. ACM*, 15(9):820–826, 1972.

Beghi, A., Cenedese, A., and Masiero, A. A multiscale stochastic approach for phase screens synthesis. In *Proceedings of the 2011 American Control Conference*, pages 3084–3089, 2011.

Beghi, A., Cenedese, A., and Masiero, A. Stochastic realization approach to the efficient simulation of phase screens. *J. Opt. Soc. Am. A*, 25(2):515–525, 2008.

Benner, P. and Faßbender, H. On the numerical solution of large-scale sparse discrete-time Riccati equations. *Advances in Computational Mathematics*, 35(2): 119, 2011.

Benner, P. and Kürschner, P. Computing real low-rank solutions of Sylvester equations by the factored ADI method. *Computers & Mathematics with Applications*, 67(9):1656 – 1672, 2014.

Benner, P., Li, J.-R., and Penzl, T. Numerical solution of large-scale Lyapunov equations, Riccati equations, and linear-quadratic optimal control problems. *Numerical Linear Algebra with Applications*, 15(9):755–777, 2008.

Benner, P., Li, R.-C., and Truhar, N. On the ADI method for Sylvester equations. *Journal of Computational and Applied Mathematics*, 233(4):1035 – 1045, 2009.

Benzi, M., Bini, D., Kressner, D., Munthe-Kaas, H., Loan, C. V., Benzi, M., and Simoncini, V. *Exploiting Hidden Structure in Matrix Computations: Algorithms and Applications Cetraro, Italy 2015*. Springer Publishing Company, Incorporated, 1st edition, 2017.

Bergstra, J. and Bengio, Y. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.*, 13:281–305, 2012.

Beylkin, G. and Mohlenkamp, M. Algorithms for numerical analysis in high dimensions. *SIAM Journal on Scientific Computing*, 26(6):2133–2159, 2005.

Bijma, F., de Munck, J. C., and Heethaar, R. M. The spatio-temporal MEG covariance matrix modeled as a sum of Kronecker products. *NeuroImage*, 27(2): 402 – 415, 2005.

Boersma, S., Doekemeijer, B., Vali, M., Meyers, J., and van Wingerden, J.-W. A control-oriented dynamic wind farm model: Wfsim. *Wind Energy Science*, 3(1): 75–95, 2018.

Boussé, M., Debals, O., and De Lathauwer, L. A tensor-based method for large-scale blind source separation using segmentation. *IEEE Transactions on Signal Processing*, 65(2):346–358, 2017a.

Boussé, M., Debals, O., and De Lathauwer, L. Tensor-based large-scale blind system identification using segmentation. *IEEE Transactions on Signal Processing*, 65 (21):5770–5784, 2017b.

Bruls, J., Chou, C., Haverkamp, B., and Verhaegen, M. Linear and non-linear system identification using separable least-squares. *European Journal of Control*, 5(1):116 – 128, 1999.

Canuto, C., Simoncini, V., and Verani, M. On the decay of the inverse of matrices that are sum of Kronecker products. *Linear Algebra and its Applications*, 452:21 – 39, 2014.

Carroll, J. D. and Chang, J.-J. Analysis of individual differences in multidimensional scaling via an n-way generalization of "Eckart-Young" decomposition. *Psychometrika*, 35(3):283–319, 1970.

Chen, T., Ohlsson, H., and Ljung, L. On the estimation of transfer functions, regularizations and gaussian processes—revisited. *Automatica*, 48(8):1525 – 1535, 2012.

Chiuso, A., Muradore, R., and Marchetti, E. Dynamic calibration of adaptive optics systems: A system identification approach. In *2008 47th IEEE Conference on Decision and Control*, pages 750–755, 2008.

Chiuso, A. The role of vector autoregressive modeling in predictor-based subspace identification. *Automatica*, 43(6):1034 – 1048, 2007.

Chiuso, A. and Pillonetto, G. A Bayesian approach to sparse dynamic network identification. *Automatica*, 48(8):1553 – 1565, 2012.

Cichocki, A., Lee, N., and Oseledets, I. *Tensor Networks for Dimensionality Reduction and Large-scale Optimization: Part 2 Applications and Future Perspectives*. Now Publishers Inc., Hanover, MA, USA, 2017.

Correia, C., Conan, J.-M., Kulcsár, C., Raynaud, H.-F., and C.Petit. Adapting optimal LQG methods to ELT-sized AO systems. *Proceedings of the 1st Conference on Adaptive Optics for Extremely Large Telescopes*, 2010.

Crespo, A., Hernández, J., and Frandsen, S. Survey of modelling methods for wind turbine wakes and wind farms. *Wind Energy*, 2(1):1–24, 1999.

Curtain, R. F. and Zwart, H. *An Introduction to Infinite-dimensional Linear Systems Theory*. Springer-Verlag, Berlin, Heidelberg, 1995.

de Silva, V. and Lim, L. Tensor rank and the ill-posedness of the best low-rank approximation problem. *SIAM Journal on Matrix Analysis and Applications*, (3): 1084–1127, 2008.

de Visser, C. C., Brunner, E., and Verhaegen, M. On distributed wavefront reconstruction for large-scale adaptive optics systems. *J. Opt. Soc. Am. A*, 33(5): 817–831, 2016.

Demko, S., Moss, W., and Smith, P. W. Decay rates for inverses of band matrices. 2010.

Ding, F. and Chen, T. Iterative least-squares solutions of coupled Sylvester matrix equations. *Systems & Control Letters*, 54(2):95 – 107, 2005.

Doekemeijer, B. M., Boersma, S., Pao, L. Y., Knudsen, T., and van Wingerden, J.-W. Online model calibration for a simplified les model in pursuit of real-time closed-loop wind farm control. *Wind Energy Science*, 3(2):749–765, 2018.

Doelman, R. and Verhaegen, M. Sequential convex relaxation for convex optimization with bilinear matrix equalities. In *2016 European Control Conference (ECC)*, pages 1946–1951, 2016.

Domanov, I. and De Lathauwer, L. On the uniqueness of the canonical polyadic decomposition of third-order tensors—part ii: Uniqueness of the overall decomposition. *SIAM Journal on Matrix Analysis and Applications*, 34(3):876–903, 2013.

Doostan, A. and Iaccarino, G. A least-squares approximation of partial differential equations with high-dimensional random inputs. *Journal of Computational Physics*, 228(12):4332 – 4345, 2009.

Eckart, C. and Young, G. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

Ellerbroek, B. The TMT Adaptive Optics Program. In *Second International Conference on Adaptive Optics for Extremely Large Telescopes*, page 6, 2011.

Ellerbroek, B. and Rhoadarmer, T. Adaptive wavefront control algorithms for closed loop adaptive optics. *Mathematical and Computer Modelling*, 33(1):145 – 158, 2001.

Ellerbroek, B. Efficient computation of minimum-variance wave-front reconstructors with sparse matrix techniques. *J. Opt. Soc. Am. A*, 19(9):1803–1816, 2002.

Farina, L. Positive systems in the state space approach: Main issues and recent results. *Proceedings of MTNS*, 2002.

Fraanje, R., Rice, J., Verhaegen, M., and Doelman, N. Fast reconstruction and prediction of frozen flow turbulence based on structured Kalman filtering. *J. Opt. Soc. Am. A*, 27(11):A235–A245, 2010.

Fried, D. L. Time-delay-induced mean-square error in adaptive optics. *J. Opt. Soc. Am. A*, 7(7):1224–1225, 1990.

Gardiner, J. D. and Laub, A. J. A generalization of the matrix sign function solution for algebraic Riccati equations. In *1985 24th IEEE Conference on Decision and Control*, pages 1233–1235, 1985.

Gavel, D. and Wiberg, D. Toward Strehl-optimizing adaptive optics controllers, 2003.

Gebraad, P. *Data-driven wind plant control.* PhD thesis, Delft Center for Systems and Control, Delft University of Technology, 2014.

Gilles, L., Massioni, P., Kulcsár, C., Raynaud, H.-F., and Ellerbroek, B. Distributed Kalman filtering compared to Fourier domain preconditioned conjugate gradient for laser guide star tomography on extremely large telescopes. *J. Opt. Soc. Am. A*, 30(5):898–909, 2013.

Giraldi, L., Nouy, A., and Legrain, G. Low-rank approximate inverse for preconditioning tensor-structured linear systems. *SIAM Journal on Scientific Computing*, 36(4):A1850–A1870, 2014.

Givone, D. D. and Roesser, R. P. Multidimensional linear iterative circuits; general properties. *IEEE Transactions on Computers*, C-21(10):1067–1073, 1972.

Golub, G. H., Nash, S., and Van Loan, C. *A Hessenberg-Schur Method for the Problem AX + XB = C.* Cornell University, Ithaca, NY, USA, 1978.

Gragg, W. B. and Lindquist, A. On the partial realization problem. *Linear Algebra and its Applications*, 50:277 – 319, 1983.

Granas, A. and Dugundji, J. *Fixed Point Theory*. Springer-Verlag, New York, New York City, 2001.

Grasedyck, L., Kressner, D., and Tobler, C. A literature survey of low-rank tensor approximation techniques. 2013.

Guyon, O. and Males, J. Adaptive optics predictive control with Empirical Orthogonal Functions (EOFs). *arXiv:1707.00570 [astro-ph.IM]*, 2017.

Guyon, O. Extreme adaptive optics. *Annual Review of Astronomy and Astrophysics*, 56(1):315–355, 2018.

Haber, A. *Estimation and control of large-scale systems with an application to adaptive optics for EUV lithography*. PhD thesis, Delft Center for Systems and Control, Delft University of Technology, 2014.

Haber, A. and Verhaegen, M. Sparsity preserving optimal control of discretized PDE systems. *arXiv:1801.05194 [math.OC]*, 2018.

Haber, A. and Verhaegen, M. Sparse solution of the Lyapunov equation for large-scale interconnected systems. *Automatica*, 73:256 – 268, 2016.

Halko, N., Martinsson, P. G., and Tropp, J. A. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Rev.*, 53(2):217–288, 2011.

Hansen, P. C., Nagy, J. G., and O'Leary, D. P. *Deblurring Images: Matrices, Spectra, and Filtering*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2006.

Hardy, J. W. *Adaptive Optics for Astronomical Telescopes*. 1998.

Harshman, R. A. Foundations of the Parafac procedure: Models and conditions for an "explanatory" multimodal factor analysis. 1970.

Harshman, R. A. The problem and nature of degenerate solutions or decompositions of 3-way arrays. *Paper presented at the American Institute of Mathematics Tensor Decomposition Workshop, Palo Alto, CA*, 2004.

Herrmann, J. Phase variance and Strehl ratio in adaptive optics. *J. Opt. Soc. Am. A*, 9(12):2257–2258, 1992.

Hinnen, K. *Data-driven optimal control for adaptive optics*. PhD thesis, Delft Center for Systems and Control, Delft University of Technology, 2007.

Hinnen, K., Verhaegen, M., and Doelman, N. Exploiting the spatiotemporal correlation in adaptive optics using data-driven $\mathcal{H}_2$-optimal control. *J. Opt. Soc. Am. A*, 24(6):1714–1725, Jun 2007.

Hitchcock, F. L. The expression of a tensor or a polyadic as a sum of products. *Journal of Mathematics and Physics*, 6(1-4):164–189, 1927.

Hoff, P. D. Multilinear tensor regression for longitudinal relational data. *Ann. Appl. Stat.*, 9(3):1169–1193, 2015.

Håstad, J. Tensor rank is NP-complete. *Journal of Algorithms*, 11(4):644 – 654, 1990.

Inigo, G. *Estimation and control of noise amplifier flows using data-based approaches*. PhD thesis, Fluids mechanics [physics.class-ph], Ecole Polytechnique, France, 2015.

Joshi, S. S., Speyer, J. L., and Kim, J. A systems theory approach to the feedback stabilization of infinitesimal and finite-amplitude disturbances in plane poiseuille flow. *Journal of Fluid Mechanics*, 332:157–184, 1997. doi: 10.1017/S0022112096003746.

Kenney, C. S. and Laub, A. J. The matrix sign function. *IEEE Transactions on Automatic Control*, 40(8):1330–1348, 1995.

Kim, J. and Bewley, T. R. A linear systems approach to flow control. *Annual Review of Fluid Mechanics*, 39(1):383–417, 2007.

Kim, S.-J., Koh, K., Lustig, M., Boyd, S., and Gorinevsky, D. An interior-point method for large-scale ℓ_1-regularized least squares. *Selected Topics in Signal Processing, IEEE Journal of*, 1:606 – 617, 2008.

Knudsen, T. Consistency analysis of subspace identification methods based on a linear regression approach. *Automatica*, 37(1):81 – 89, 2001.

Kolda, T. G. Multilinear operators for higher-order decompositions.

Kolda, T. G. and Bader, B. W. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009.

Kolmogorov, A. N. The local structure of turbulence in incompressible viscous fluid for very large Reynolds numbers. *Proceedings: Mathematical and Physical Sciences*, 434(1890):9–13, 1991.

Kruskal, J. B. Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. *Linear Algebra and its Applications*, 18(2):95 – 138, 1977.

Kulcsár, C., Raynaud, H.-F., Petit, C., Conan, J.-M., and de Lesegno, P. V. Optimal control, observers and integrators in adaptive optics. *Opt. Express*, 14(17):7464–7476, Aug 2006.

Kulcsár, C., Raynaud, H.-F., Petit, C., and Conan, J.-M. Minimum variance prediction and control for adaptive optics. *Automatica*, 48(9):1939 – 1954, 2012.

Lele, S. and Mendel, J. Modeling and recursive state estimation for two-dimensional noncausal filters with applications in image restoration. *IEEE Transactions on Circuits and Systems*, 34(12):1507–1517, 1987.

Lessard, L. and Lall, S. Convexity of decentralized controller synthesis. *IEEE Transactions on Automatic Control*, 61(10):3122–3127, Oct 2016.

Li, G., Wen, C., and Zhang, A. Fixed point iteration in identifying bilinear models. *Systems & Control Letters*, 83:28 – 37, 2015.

Li, N., Kindermann, S., and Navasca, C. Some convergence results on the regularized alternating least-squares method for tensor decomposition. *Linear Algebra and its Applications*, 438(2):796 – 812, 2013.

Ljung, L. *System Identification - Theory for the User*. PTR Prentice Hall, Upper Saddle River, N.J., 2nd ed edition, 1999.

Massioni, P. and Verhaegen, M. Distributed control for identical dynamically coupled systems: A decomposition approach. *IEEE Transactions on Automatic Control*, 54(1):124–135, 2009a.

Massioni, P. and Verhaegen, M. Subspace identification of distributed, decomposable systems. In *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pages 3364–3369, 2009b.

Massioni, P. Distributed control for alpha-heterogeneous dynamically coupled systems. *Systems & Control Letters*, 72:30 – 35, 2014.

Massioni, P. and Verhaegen, M. Subspace identification of circulant systems. *Automatica*, 44:2825–2833, 2008.

Massioni, P., Kulcsár, C., Raynaud, H.-F., and Conan, J.-M. Fast computation of an optimal controller for large-scale adaptive optics. *J. Opt. Soc. Am. A*, 28(11): 2298–2309, 2011.

Massioni, P., Gilles, L., and Ellerbroek, B. Adaptive distributed Kalman filtering with wind estimation for astronomical adaptive optics. *J. Opt. Soc. Am. A*, 32 (12):2353–2364, 2015.

Mohlenkamp, M. J. Musings on multilinear fitting. *Linear Algebra and its Applications*, 438(2):834 – 852, 2013. Tensors and Multilinear Algebra.

Motee, N. and Jadbabaie, A. Optimal control of spatially distributed systems. *IEEE Transactions on Automatic Control*, 53(7):1616–1629, Aug 2008.

Neichel, B., Fusco, T., Sauvage, J.-F., Correia, C., Dohlen, K., El-Hadi, K., Blanco, L., Schwartz, N., Clarke, F., Thatte, N. A., Tecza, M., Paufique, J., Vernet, J., Le Louarn, M., Hammersley, P., Gach, J.-L., Pascal, S., Vola, P., Petit, C., Conan, J.-M., Carlotti, A., Vérinaud, C., Schnetler, H., Bryson, I., Morris, T., Myers, R., Hugot, E., Gallie, A. M., and Henry, D. M. The adaptive optics modes for HARMONI: from Classical to Laser Assisted Tomographic AO. In *Adaptive Optics Systems V*, 2016.

Nocedal, J. and Wright, S. J. *Numerical Optimization*. Springer, New York, NY, USA, second edition, 2006.

Noll, R. J. Zernike polynomials and atmospheric turbulence. *J. Opt. Soc. Am.*, 66 (3):207–211, 1976.

Olshevsky, V., Oseledets, I., and Tyrtyshnikov, E. Superfast inversion of two-level toeplitz matrices using newton iteration and tensor-displacement structure. In *Recent Advances in Matrix and Operator Theory*, pages 229–240, Basel, 2008. Birkhäuser Basel.

Oseledets, I. and Dolgov, S. Solution of linear systems and matrix inversion in the tt-format. *SIAM Journal on Scientific Computing*, 34(5):A2718–A2739, 2012.

Penzl, T. A cyclic low-rank Smith method for large sparse Lyapunov equations. *SIAM Journal on Scientific Computing*, 21(4):1401–1418, 1999.

Petit, C., Sauvage, J.-F., Fusco, T., Sevin, A., Suarez, M., Costille, A., Vigan, A., Soenke, C., Perret, D., Rochat, S., Barrufolo, A., Salasnich, B., Beuzit, J.-L., Dohlen, K., Mouillet, D., Puget, P., Wildi, F., Kasper, M., Conan, J.-M., Kulcsár, C., and Raynaud, H.-F. SPHERE eXtreme AO control scheme: final performance assessment and on sky validation of the first auto-tuned LQG based operational system, 2014.

Piatrou, P. and Roggemann, M. C. Performance study of Kalman filter controller for multiconjugate adaptive optics. *Appl. Opt.*, 46(9):1446–1455, 2007.

Pillonetto, G., Dinuzzo, F., Chen, T., Nicolao, G. D., and Ljung, L. Kernel methods in system identification, machine learning and function estimation: A survey. *Automatica*, 50(3):657 – 682, 2014.

Piscaer, P. *Sparse VARX Model Identification for Large-Scale Adaptive Optics*. Master thesis, Delft Center for Systems and Control, Delft University of Technology, 2016.

Poyneer, L. A., Macintosh, B. A., and Véran, J.-P. Fourier transform wavefront control with adaptive prediction of the atmosphere. *J. Opt. Soc. Am. A*, 24(9): 2645–2660, Sep 2007.

Poyneer, L. A., Palmer, D. W., Macintosh, B., Savransky, D., Sadakuni, N., Thomas, S., Véran, J.-P., Follette, K. B., Greenbaum, A. Z., Ammons, S. M., Bailey, V. P., Bauman, B., Cardwell, A., Dillon, D., Gavel, D., Hartung, M., Hibon, P., Perrin, M. D., Rantakyrö, F. T., Sivaramakrishnan, A., and Wang, J. J. Performance of the Gemini Planet Imager adaptive optics system. *Appl. Opt.*, 55(2):323–340, 2016.

Ramos, J. A. and Mercère, G. A stochastic subspace system identification algorithm for state-space systems in the general 2-D Roesser model form. *International Journal of Control*, 0(0):1–29, 2018.

Rantzer, A. Distributed control of positive systems. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 6608–6611, 2011.

Rice, J. *Efficient algorithms for distributed control: a structured matrix approach*. PhD thesis, Delft Center for Systems and Control, Delft University of Technology, 2010.

Rice, J. K. and Verhaegen, M. Distributed control: A sequentially semi-separable approach for spatially heterogeneous linear systems. *IEEE Transactions on Automatic Control*, 54(6):1270–1283, 2009.

Rice, J. K. and Verhaegen, M. Efficient system identification of heterogeneous distributed systems via a structure exploiting extended Kalman filter. *IEEE Transactions on Automatic Control*, 56(7):1713–1718, 2011.

Roberts, J. D. Linear model reduction and solution of the algebraic Riccati equation by use of the sign function. *International Journal of Control*, 32(4):677–687, 1980.

Roddier, F. *Adaptive Optics in Astronomy*. Cambridge University Press, New York, NY, USA, 1st edition, 2004.

Roesser, R. A discrete state-space model for linear image processing. *IEEE Transactions on Automatic Control*, 20(1):1–10, 1975.

Rogers, M., Li, L., and Russell, S. J. Multilinear dynamical systems for tensor time series. In *Advances in Neural Information Processing Systems 26*, pages 2634–2642, 2013.

Rotkowitz, M. and Lall, S. A characterization of convex problems in decentralized control. *IEEE Transactions on Automatic Control*, 51(2):274–286, Feb 2006.

Roux, B. L., Conan, J.-M., Kulcsár, C., Raynaud, H.-F., Mugnier, L. M., and Fusco, T. Optimal control law for classical and multiconjugate adaptive optics. *J. Opt. Soc. Am. A*, 21(7):1261–1276, Jul 2004.

Sabino, J. *Solution of large-scale Lyapunov equations via the block modified Smith method*. PhD thesis, Rice University, 2006.

Sayed, A. H. and Kailath, T. Recursive least-squares adaptive filters. *The Digital Signal Processing Handbook*, 21(1), 1998.

Sivo, G., Kulcsár, C., Conan, J.-M., Raynaud, H.-F., Éric Gendron, Basden, A., Vidal, F., Morris, T., Meimon, S., Petit, C., Gratadour, D., Martin, O., Hubert, Z., Sevin, A., Perret, D., Chemla, F., Rousset, G., Dipper, N., Talbot, G., Younger, E., Myers, R., Henry, D., Todd, S., Atkinson, D., Dickson, C., and Longmore, A. First on-sky SCAO validation of full LQG control with vibration mitigation on the CANARY pathfinder. *Opt. Express*, 22(19):23565–23591, 2014.

Smith, R. Matrix equation $xa + bx = c$. *SIAM Journal on Applied Mathematics*, 16 (1):198–201, 1968.

Torres, P., van Wingerden, J. W., and Verhaegen, M. Output-error identification of large scale 1D-spatially varying interconnected systems. *IEEE Transactions on Automatic Control*, 60(1):130–142, 2015.

Tsiligkaridis, T. and Hero, A. O. Covariance estimation in high dimensions via Kronecker product expansions. *IEEE Transactions on Signal Processing*, 61(21): 5347–5360, 2013.

Udell, M., Horn, C., Zadeh, R., and Boyd, S. Generalized low-rank models. *Foundations and Trends in Machine Learning*, 9:1 – 118, 2016.

Van den Hof, P. *System identification. Data-driven modelling of dynamic systems.* Lecture notes, Dutch Institute for Systems and Control, 2018.

van Loan, C. The ubiquitous Kronecker product. *Journal of Computational and Applied Mathematics*, 123(1):85 – 100, 2000. Numerical Analysis 2000. Vol. III: Linear Algebra.

van Loan, C. and Pitsianis, N. Approximation with Kronecker products. In *Linear Algebra for Large Scale and Real Time Applications*, pages 293–314. Kluwer Publications, 1993.

Varnai, P. *Exploiting Kronecker structures, with applications to optimization problems arising in the field of adaptive optics.* Master of Science thesis, Delft Center for Systems and Control, Delft University of Technology, 2017.

Verhaegen, M. and Hansson, A. N2SID. *Automatica*, 72(C):57–63, 2016.

Verhaegen, M. and Verdult, V. *Filtering and System Identification: A Least Squares Approach.* Cambridge University Press, New York, NY, USA, 1st edition, 2007.

Vervliet, N. and De Lathauwer, L. *Numerical optimization based algorithms for data fusion, Data Fusion Methodology and Applications.* Elsevier, vol. 33; pp. 1 - 41 edition, 2018.

Vervliet, N., Debals, O., Sorber, L., Van Barel, M., and De Lathauwer, L. Tensorlab 3.0, 2016. Available online. `https://www.tensorlab.net`.

Voorsluys, M. *Subspace identification of Roesser models for large-scale adaptive optics.* MSc thesis at Delft Center for Systems and Control, Delft University of Technology, 2015.

Wang, X., Ding, F., Alsaadi, F. E., and Hayat, T. Convergence analysis of the hierarchical least squares algorithm for bilinear-in-parameter systems. *Circuits, Systems, and Signal Processing*, 35(12):4307–4330, Dec 2016.

Wang, Y., Matni, N., and Doyle, J. C. Separable and localized system-level synthesis for large-scale systems. *IEEE Transactions on Automatic Control*, 63(12):4234–4249, Dec 2018.

Wu, Z. Multidimensional state-space model Kalman filtering with application to image restoration. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 33(6):1576–1592, 1985.

Xu, Y. and Yin, W. A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion. *SIAM Journal on Imaging Sciences*, 6(3):1758–1789, 2013.

Yu, C. and Verhaegen, M. Subspace identification of distributed clusters of homogeneous systems. *IEEE Transactions on Automatic Control*, 62(1):463–468, 2017.

Yu, C. and Verhaegen, M. Subspace identification of individual systems operating in a network (SI^{2}ON). *IEEE Transactions on Automatic Control*, 63(4):1120–1125, 2018a.

Yu, C. and Verhaegen, M. Structured modeling and control of adaptive optics systems. *IEEE Transactions on Control Systems Technology*, 26(2):664–674, 2018b.

Yu, C., Verhaegen, M., and Hansson, A. Subspace identification of local systems in one-dimensional homogeneous networks. *IEEE Transactions on Automatic Control*, 63(4):1126–1131, 2018a.

Yu, C., Ljung, L., and Verhaegen, M. Identification of structured state-space models. *Automatica*, 90:54 – 61, 2018b.

Zorzi, M. and Chiuso, A. A Bayesian approach to sparse plus low rank network identification. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 7386–7391, 2015.

List of Publications

Below is a numbered list of publications in reverse chronological order for their submission.

Journal papers

4. B. Sinquin, M. Verhaegen, "QUARKS: Identification of Large-Scale Kronecker Vector-AutoRegressive Models", *IEEE Transactions on Automatic Control*, vol. 64, no. 2, pp.448-463, 2019.

3. B. Sinquin, M. Verhaegen, "K4SID: Large-Scale Subspace Identification with Kronecker modeling", *IEEE Transactions on Automatic Control*, vol. 64, no. 3, pp. 960-975, 2019.

2. G. Monchen, B. Sinquin, M. Verhaegen, "Recursive Kronecker-Based Vector Autoregressive Identification for Large-Scale Adaptive Optics", *IEEE Transactions on Control Systems Technology*, 2018.

1. B. Sinquin, M. Verhaegen, "Tensor-based predictive control for extremely large-scale single conjugate adaptive optics", *J. Opt. Soc. Am. A* 35, 1612-1626 (2018).

Contributions to workshops and conferences

6. B. Sinquin, M. Verhaegen, "Subspace identification of 1D spatially-varying systems using Sequentially Semi-Separable matrices", *Proceedings of the American Control Conference, Boston, MA*, pp. 54-59, (2016).

5. B. Sinquin, M. Verhaegen, "Towards Scalable Subspace Identification with Kronecker modeling", *Proceedings of the 36th Benelux Meeting on Systems and Control*, Spa, Belgium, (2017).

4. B. Sinquin, M. Verhaegen, "Identification of Kronecker-structured autoregressive models", Talk at the *Symposium on Information Theory and Signal Processing in the Benelux*, Delft, the Netherlands, (2017).

3. B. Sinquin, M. Verhaegen, "Kronecker-ARX Models in Identifying (2D) Spatial-Temporal Systems", *Proceedings of 2017 IFAC World Congress*, Toulouse, France, vol. 50, issue 1, pp. 14131-14136, (2017).

2. B. Sinquin, M. Verhaegen, "Solving Kronecker-structured discrete Lyapunov and Riccati equations", Poster presentation at *EURASIP Summer School on Tensor-Based Signal Processing*, Leuven, Belgium, (2018).

1. B. Sinquin, M. Verhaegen, "Tensor-based predictive control for large-scale single conjugate adaptive optics", Talk at the *Conference on Adaptive Optics wavefront sensing and control in the VLT/ELT era*, Paris, France, (2018).

Curriculum Vitæ

Baptiste Sinquin was born in 1991 in Quimperlé, France.

From 2009 to 2011, he studied mathematics and physics in Lycée Dupuy de Lôme, Lorient, to prepare for entrance exams to the French Grandes Ecoles. He started his engineering studies at Ecole Centrale de Lyon where he received during three years a multi-disciplinary education including mathematics, electrical engineering, mechanics, materials science. From September 2013, he pursued a double degree with Université Lyon I to study signal and image processing for biomedical applications.

He discovered Delft and the Netherlands in May 2014 during a five months internship under the supervision of Prof.dr.ir. M.Verhaegen. Early November 2014, he started his doctoral studies in the same group focusing on control for high resolution imaging. He investigated a particular matrix structure for identifying the spatio-temporal dynamics of large systems in a scalable manner, and for predicting a stochastic disturbance for control. The main targeted application was the adaptive optics for ground-based large telescopes.

He can be reached via e-mail: baptiste.sinquin@gmail.com.