KU LEUVEN

ARENBERG DOCTORAL SCHOOL
Faculty of Engineering Science

A Polynomial Approach to Globally Optimal Multidimensional System Identification

Lukas Vanpoucke

Dissertation presented in partial fulfillment
of the requirements for the degree of
Doctor of Engineering Science (PhD):
Electrical Engineering

Supervisor:
Prof. dr. ir. B. De Moor

June 2026

A POLYNOMIAL APPROACH TO GLOBALLY OPTIMAL MULTIDIMENSIONAL SYSTEM IDENTIFICATION

Lukas VANPOUCKE

Supervisor:
Prof. dr. ir. B. De Moor

Members of the
Examination Committee:
Prof. dr. ir. P. Sas, chair
Prof. dr. ir. K. Meerbergen
Prof. dr. F. Mohammadi
Prof. dr. ir. L. De Lathauwer
Dr. ir. P. Dreesen
(Maastricht University)

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor of Engineering Science (PhD): Electrical Engineering

June 2026

© 2026 Lukas Vanpoucke
Uitgegeven in eigen beheer, Lukas Vanpoucke, Leuven (Belgium)

Alle rechten voorbehouden. Niets uit deze uitgave mag worden vermenigvuldigd en/of openbaar gemaakt worden door middel van druk, fotokopie, microfilm, elektronisch of op welke andere wijze ook zonder voorafgaande schriftelijke toestemming van de uitgever.

All rights reserved. No part of the publication may be reproduced in any form by print, photoprint, microfilm, electronic or any other means without written permission from the publisher.

Preface

First of all, I would like to express my gratitude to the many people who have made this journey possible.

For starters, I would like to thank my promotor, Bart De Moor, for the opportunity to work on the 'back to the roots' project, which lies at the fascinating intersection of systems theory and polynomials. Though daunting at times, I am glad to have come into contact with these abstract topics, which have provided me with a newfound appreciation for the beauty of mathematics, in all its rigor. These past years, Bart has shaped me as a researcher, teaching me how to approach the challenges within, and to always keep an open mind through his many ideas and associations. He encouraged us to take initiative and ownership of our projects, maintaining a stern eye for quality, while he could always be relied upon for sound advice.

I would also like to acknowledge the support of the ERC under the ERC Advanced Grant 885682. My sincere gratitude goes to the members of the supervisory committee for their guidance and feedback throughout this project, and to the examination committee for the time and effort devoted to evaluating this manuscript, their insightful comments and their suggestions.

I will always have fond memories of my time at ESAT, where I enjoyed many pleasant coffee breaks and lunches, discussing the latest hypes in AI, world news, and, fortunately, many more lighthearted topics as well. For this, I would like to acknowledge my colleagues, Vincent, Axel, Alexandre, Lola, Bram, Jaldert, Thomas, Naomi, Melanie, Willem, Kostas, Zahra, Giulia, Fernando, Yoeri, and Yamini, for making it such an enjoyable place to work. I am also grateful to Ida, Elsy, and Aldona for all their efforts in keeping ESAT running smoothly, and for the downstairs coffee breaks that brought everyone together. Additional appreciation is owed to Katrien, Philippe, Mauricio, Benoît, and Christof for their mentorship, encouragement, readiness to help, and swift and thoughtful feedback over the years. Finally, I also want to express my special thanks to the

rest of the “rooters”, as Bart likes to call us—Hans, Sarthak, Sem, Marie, and especially Sibren—for the interesting ideas, discussions, and collaborations from which I learned so much, and, of course, the many exciting games of kicker.

Of course, this journey would not have been the same without my friends and family, who have given me so many wonderful moments and helped me navigate the challenges of everyday life. In most recent memory, many of them have, quite literally, helped us build a home, for which I am eternally grateful. I would like to thank my teammates at the Handball clubs of Rhino and Atomix for keeping my competitive spirit alive, as well as for the enjoyable drinks off the court. Benjamin, Alicia, and Thomas I thank for the lovely brunches and seaside trips over the past years. I have also had the pleasure of sharing many musical intermezzos with Nele and Joppe over the past years, often accompanied by enjoyable dinners and receptions. We even owe the name of our tiny band, Maatrix, to the topic of this PhD, or, more so, my inadequate explanation of the topic as “something with matrices”. To Ruben, Arno, Matteo, Cyriel, Victor, Felix, Klara, Kirsten, Marica, Elina, and Camille: nothing quite beats a drink, meal, or any kind of gathering with you after a hard week’s work. I am very grateful for all the incredibly fun times we have had, and undoubtedly will continue to have, and it is reassuring to know I can always count on you and your hospitality.

I have been very fortunate to have been warmly welcomed by my partner’s family Koen, Els, Lorien, Lise, and Seppe. I greatly appreciate your help throughout the years, especially during these past months. To my grandparents, thank you for all your love and for being my biggest fans. To my brother and sister: you will always be my oldest friends, and I especially cherish the nice times we had in Leuven together. I hope to proudly read your prefaces soon, and for many more family bike trips in the meantime. My deepest appreciation goes to my parents, whom I thank for all their love, support, and encouragement throughout my life, and for providing me with a warm home. Their toils in our renovation project also deserve a special mention. Last, though certainly not least, Katrijn: thank you sincerely for your unrelenting support, love, and care over the past years, and your exceptional hard work in these past months. You always manage to lift my spirits and soothe my mind, and I could not have done this without you. We have already enjoyed many adventures together, and I look forward to continuing to join you in the many travels and projects you so dream of.

Popularized Abstract

Engineering and sciences are full of linear models used to predict, analyze, and control (physical) processes. More and more, such models are being estimated from data, that is, by attempting to find the model parameters that best fit a given set of measurement data. In practice, it is often only possible to tackle such parameter fitting problems by finding a sufficiently good set of model parameters, rather than the best possible one. In an exact sense, the underlying parameter fitting problem is thus not solved. Instead, one resorts to an intuitive guesstimate, which could be considered 'heuristic' since one does not know whether it can be improved upon.

To address this discrepancy, this dissertation takes a fundamentally different approach that allows us to solve such problems in a globally optimal way, guaranteeing the best possible fit. In particular, a methodology previously applied to the estimation of one-dimensional linear models is extended to multidimensional linear models. To this end, we show that the latter model class is closely linked with a particular type of mathematical object: multivariate polynomials. Using this relation, we highlight known mathematical properties of the models and derive new ones as well. These properties allow us to formulate and study a particular multidimensional parameter fitting problem, to which the aforementioned globally optimal methodology is extended. In the process, new theoretical properties for this problem, more specifically of its minima, are obtained as well.

The methodology under study relies on the fact that the related mathematical objects—polynomial equations—can be handled well using computer-based techniques. More precisely, there exist algorithms that compute all their solutions, and we show that the best-fitting model can be found among such solutions. As such, to further extend the capabilities of the methodology, alternative algorithms to handle this underlying mathematical task are also explored. This results in two novel algorithms to perform computational steps within a particular type of solver.

Gepopulariseerde Samenvatting

In de techniek en de wetenschappen worden vaak lineaire modellen gebruikt om (fysische) processen te voorspellen, te analyseren en te regelen. Steeds vaker worden dergelijke modellen geschat op basis van gegevens, wat gebeurt door te zoeken naar de modelparameters die het beste aansluiten bij een reeks metingen. In de praktijk worden dergelijke parameterschattingsproblemen veelal opgelost door een set modelparameters te vinden die voldoende goed is, in plaats van de best mogelijke parameters te selecteren. In strikte zin wordt het onderliggende schattingsprobleem dus niet opgelost. In plaats daarvan gebruikt men een goede benadering, die echter als 'heuristisch' kan worden beschouwd, aangezien men niet weet of deze kan worden verbeterd.

Om deze discrepantie aan te pakken, hanteert dit proefschrift een fundamenteel andere benadering die ons in staat stelt dergelijke problemen op een globaal optimale manier op te lossen, zodat men met zekerheid de best mogelijke modelparameters vindt. In het bijzonder wordt een methodologie die eerder werd toegepast voor het schatten van eendimensionale lineaire modellen, uitgebreid naar multidimensionale lineaire modellen. Daartoe tonen we aan dat deze laatste modelklasse nauw verbonden is met een bepaald type wiskundig object: multivariate polynomen. We bespreken bekende wiskundige eigenschappen die voortvloeien uit deze verbinding en we leiden er ook nieuwe af. Deze eigenschappen stellen ons in staat een specifiek multidimensionaal parameterschattingsprobleem te formuleren en te bestuderen, waarop de bovengenoemde globaal optimale methodologie wordt toegepast. Tegelijkertijd worden nieuwe theoretische eigenschappen van dit soort problemen, meer bepaald, van hun minima, verkregen.

De bestudeerde methodologie berust op het feit dat de betreffende wiskundige objecten—polynomiale vergelijkingen—goed kunnen worden aangepakt met behulp van computergebaseerde technieken. Meer bepaald bestaan er algoritmen die al hun oplossingen berekenen, en we tonen aan dat het best passende model tussen dergelijke oplossingen gevonden kan worden. Om de mogelijkheden van de methodologie verder uit te breiden, worden er ook alternatieve algoritmes onderzocht om deze onderliggende wiskundige taak aan te pakken. Dit resulteert in twee nieuwe algoritmen om stappen uit te voeren binnen een bepaald type oplossingsalgoritme.

Abstract

Linear dynamical models are ubiquitous in sciences and engineering, where they describe all sorts of phenomena, which allows one to analyze, predict, or control the behavior of the underlying system through these models. For various reasons, it is not always possible to obtain such models from first principles. In such cases, one may resort to fitting a model from measurement data, a practice called system identification. This practice is more of an art than a science, relying on statistical assumptions, often unverifiable in practice, as well as locally optimal algorithms to solve the underlying optimization problems.

This dissertation focuses on a framework for the deterministic globally optimal system identification of linear models, which relies on characterizing the stationary points of such optimization problems as common roots of polynomial equations, which can be tackled with numerical root-finding techniques. The aim of this framework is to refrain from the use of such 'heuristic' methodologies, studying the properties of the stationary points—in particular, the local minimizers—in the process. More precisely, we generalize this framework to the multidimensional setting, tackling the least-squares misfit problem for finite-order, autonomous, scalar, linear, shift-invariant models in a globally optimal way. Moreover, we investigate novel methodologies for such numerical root-finding algorithms, as they form the backbone of this framework for globally optimal parameter estimation.

To this end, the intricate connection between the model class under consideration and multivariate polynomials is discussed, exploring the difference-equation, image, and state-space based representations, as well as the algorithmic transitions between them. A novel methodology to directly transition from a state-space model to an equivalent difference-equation representation is presented, based on the generalized Cayley-Hamilton theorem. This method also leads to a more general parameterization for multidimensional difference equations. Furthermore, polynomials with roots at infinity are linked to the less-studied class of descriptor models, providing a complete characterization of

their representations, including a rational state-space representation.

This connection with multivariate polynomials forms the foundation for extending the globally optimal parameter estimation approach to the multidimensional misfit problem. Indeed, we show that stationary points of this problem are common roots of polynomial equations. From the duality of the image and difference equation representations, a second such polynomial characterization is derived as well. Moreover, the theoretical properties of the local minimizers of this misfit problem are studied as well, which is subsequently employed to simplify the polynomial characterization and leads to a previously unknown interpolation property. The solution sets of the resulting polynomial characterizations are studied as well, showing that the minimizers correspond to isolated solutions, which allows them to be found by root-finding techniques. However, the presence of infinite components in the solution sets is also indicated, which hampers the convergence of algebraic root-finding techniques. While the computational complexity of the proposed methodology is high compared to practical methods, these properties provide useful theoretic insight into the local minimizers of these optimization problems, which can serve to benchmark and improve locally optimal approaches.

The studied framework relies on numerical root-finding algorithms for polynomial and multiparameter eigenvalue problems. The (block-)Macaulay framework, which relies on the highly structured (block-)Macaulay matrix, provides a linear algebra-based computational toolbox to handle multivariate polynomials. The null space basis is especially important to root-finding applications. In an attempt to ease the computational burden of the studied globally optimal approach, we explore ways to exploit the shift-invariant structure of Macaulay matrices to compute bases for its null and row space, without explicitly computing this large matrix itself. A novel methodology is presented which achieves this by 'shifting' previously computed bases and taking their intersections, based on a particular block-row decomposition of the Macaulay matrix. This re-use of intermediate results leads to a novel, recursive algorithm to compute these bases.

Additionally, Gröbner and border basis methods have long shown that the full null space of a Macaulay matrix, which can be rather large, is not required for algebraic root finding. Taking inspiration from this observation, a novel methodology to compute the multiplication maps is explored, one that fully relies on numerical linear algebra steps involving the coefficient matrices w.r.t. the underlying basis. This results in reduced symbolic bookkeeping and gives the approach more flexibility in choosing the underlying polynomials. Moreover, this approach hinges on a new rank-based check to decide upon the number of new basis elements to add in each iteration.

Beknopte samenvatting

Lineaire dynamische modellen zijn alomtegenwoordig in de wetenschap en de techniek, waar ze allerlei verschijnselen beschrijven. Zulke modellen laten toe het gedrag van het onderliggende systeem te analyseren. Om verschillende redenen is het niet altijd mogelijk om dergelijke modellen uit eerste principes af te leiden. In dergelijke gevallen kan men modellen trachten te schatten op basis van meetgegevens, een techniek die systeemidentificatie wordt genoemd. Deze praktijk is meer een kunst dan een wetenschap en berust meestal op statistische aannames, die in de praktijk vaak niet verifieerbaar zijn, of op lokaal optimale algoritmen om de onderliggende optimalisatieproblemen op te lossen.

Dit proefschrift richt zich op een raamwerk voor het deterministisch en globaal optimaal identificeren van lineaire modellen. Dit raamwerk berust op het karakteriseren van de stationaire punten van de onderliggende optimalisatieproblemen als gemeenschappelijke wortels van polynomen. Er zijn namelijk allerhande numerieke technieken beschikbaar om zulke wortels te bepalen. Het doel van dit raamwerk is om af te zien van het gebruik van 'heuristische' methoden en daarbij de eigenschappen van de stationaire punten—in het bijzonder de lokale minima—te bestuderen. Meer bepaald generaliseren we dit raamwerk naar multidimensionale modellen, waarbij we het kleinste-kwadraten-misfitprobleem voor autonome, scalaire, lineaire, verschuivingsinvariante modellen van eindige orde op een globaal optimale manier oplossen. Daarnaast onderzoeken we ook nieuwe oplossingsalgoritmen voor veeltermen, aangezien deze methodologie voor globaal optimale parameterschatting hier sterk op steunt.

Hiertoe wordt het nauwe verband tussen de beschouwde modelklasse en multivariate polynomen besproken, waarbij representaties gebaseerd op differentievergelijkingen, het beeld van een matrix en toestandsruimtemodellen worden besproken, evenals de algoritmes voor de overgangen tussen deze representaties. Er wordt een nieuwe methodologie gepresenteerd om direct vanuit een toestandsruimtemodel een equivalente representatie op basis van differentievergelijkingen te verkrijgen, gebaseerd op de veralgemeende stelling

van Cayley-Hamilton. Deze methode leidt ook tot een meer algemene parametrisatie voor multidimensionale differentievergelijkingen. Daarenboven worden polynomen met wortels op oneindig gekoppeld aan de minder bestudeerde klasse van descriptormodellen, wat tot een volledige karakterisatie van hun voorstellingen leidt, inclusief een rationaal toestandsruimtemodel.

Dit verband met multivariate polynomen vormt de basis om de globaal optimale schattingsmethode naar het multidimensionale misfitprobleem uit te breiden. Hiertoe tonen we aan dat de stationaire punten van dit probleem wortels van polynomiale vergelijkingen zijn. Uit de dualiteit van de beeld- en differentiegebaseerde voorstellingen wordt ook een tweede polynomiale karakterisatie afgeleid. Bovendien bestuderen we de theoretische eigenschappen van de lokale minima van dit misfitprobleem en gebruiken deze vervolgens om deze polynomiale karakterisatie te vereenvoudigen en een voorheen onbekende interpolatie-eigenschap af te leiden. De oplossingsverzamelingen van de resulterende polynomiale karakterisaties worden eveneens bestudeerd, waaruit blijkt dat de minima overeenkomen met geïsoleerde oplossingen. Dit maakt dat ze inderdaad kunnen worden gevonden met behulp van numerieke technieken voor het oplossen van veeltermstelsels. De aanwezigheid van oneindige componenten in de oplossingsverzamelingen wordt ook aangetoond, wat de convergentie van algebraïsche oplossingstechnieken belemmert. Hoewel de rekenkundige complexiteit van de voorgestelde methodologie hoog is in vergelijking met praktische methoden, bieden deze eigenschappen nuttige theoretische inzichten in de lokale minima van deze optimalisatieproblemen, wat kan helpen om lokaal optimale benaderingen te benchmarken en te verbeteren.

Het bestudeerde raamwerk steunt op numerieke algoritmes om polynomiale en multiparametereigenwaardeproblemen op te lossen. De (blok)Macaulaymethode, dat berust op de sterk gestructureerde Macaulaymatrix, biedt lineair-algebraische technieken om met multivariate polynomen om te gaan. De basis van de nulruimte van deze matrix is vooral belangrijk om dergelijke stelsels op te lossen. In een poging om de complexiteit van de bestudeerde globaal optimale benadering te verlichten, onderzoeken we manieren om de verschuivingsinvariante structuur van Macaulaymatrices te benutten om basissen voor de nul- en rijruimte te berekenen, zonder dat deze grote matrix zelf expliciet berekend wordt. Er wordt een nieuwe methodologie gepresenteerd die dit bereikt door eerder berekende basissen te ‘verschuiven’ en de intersectie van hun beelden te nemen. Deze aanpak berust op een specifieke blok-rijdecompositie van de Macaulaymatrix en hergebruikt tussenresultaten. Dit leidt tot een nieuw, recursief algoritme om zulke basissen te berekenen.

Gröbner- en borderbasismethoden hebben reeds aangetoond dat de volledige nulruimte van een Macaulaymatrix, die vrij groot kan zijn, niet nodig is voor het algebraïsch bepalen van wortels. Geïnspireerd door deze

observatie wordt een nieuwe methodologie voor het berekenen van de vermenigvuldigingsmatrices onderzocht. Het bekomen algoritme berust volledig op stappen gebaseerd op numerieke lineaire algebra die enkel betrekking hebben tot de coëfficiëntenmatrices. Dit resulteert in minder symbolische administratie en geeft het proces meer vrijheid in het kiezen van de onderliggende polynomen. Bovendien steunt dit algoritme op een nieuwe ranggebaseerde bepaling van het aantal nieuwe basiselementen dat in elke iteratie moet worden toegevoegd.

List of Abbreviations

CPD Canonical polyadic decomposition

FONC First-order necessary conditions for optimality

GCH Generalized Cayley-Hamilton

GESD Generalized eigenspace decomposition

JCF Jordan canonical form

LTI Linear, time-invariant

mD Multidimensional/ m-dimensional

MEP Multiparameter eigenvalue problem

MHR Multidimensional harmonic retrieval

NLA Numerical linear algebra

ODE Ordinary differential equation

PBH Popov-Belevitch-Hautus

PDE Partial differential equation

PdE Partial difference equation

PEM Prediction error methods

SOS Sum-of-squares

SVD Singular value decomposition

List of Symbols

General notation

$\boldsymbol{A}_k$	kth state-transition matrix/multiplication map (out of m)
$\mathbb{C}[\boldsymbol{z}]/I$	Coordinate ring of the ideal I
m	Number of independent indices of signals/number of variables of polynomials
$\boldsymbol{\kappa}$	Multi-index in $\mathbb{N}_0^m$
$\mathbb{P}$	Projective space
$\boldsymbol{V}$	Matrix of arbitrary dimensions
$\mathbb{N}_0$	Natural numbers including 0
$\mathbb{R}[\boldsymbol{z}]$	Ring of polynomials in with real coefficients in the variables $\boldsymbol{z} \in \mathbb{C}^m$ ($\mathbb{C}[\boldsymbol{z}]$ defined similarly)
$\boldsymbol{v}$	Vector of arbitrary dimension
$\boldsymbol{z}^{\boldsymbol{\alpha}}$	Monomial $\prod_i^m z_k^{\alpha_k}$
I	Polynomial ideal generated by f_i as $I = \{\sum_{i=1}^s h_i f_i \mid h_i(\boldsymbol{z}) \in \mathbb{C}[\boldsymbol{z}]\}$

Part I: Systems theory and identification

$\boldsymbol{C}$	Model output matrix
$\mathcal{K}$	Set of grid points
$\boldsymbol{M}_{\boldsymbol{r}}$	Macaulay matrix built from difference equations $\boldsymbol{r}(\boldsymbol{\sigma})$
$\boldsymbol{\lambda}$	Eigentuple, ith eigentuples indicated as $\boldsymbol{\lambda}_i$
n	Model order

$\boldsymbol{\Gamma}$ Observability matrix

$\boldsymbol{r}(\boldsymbol{\sigma})$ Difference equations/GCH equations

$\boldsymbol{\sigma}$ Shift operators

$\boldsymbol{x}$ Model state

$\boldsymbol{y}$ Vector containing data sequence.

$\hat{\boldsymbol{y}}$ Vector containing model-compliant data sequence

$\tilde{\boldsymbol{y}}$ Vector containing misfit sequence

Part II: Algorithms

$\mathcal{C}$ Set of row column mult-indices of a Macaulay matrix

$\mathcal{G}_d$ Set of points corresponding to the cube $\{\boldsymbol{\rho} \in \mathbb{N}_0^m \mid \|\boldsymbol{\rho}\|_\infty \leq d\}$

$\boldsymbol{M}^{\mathcal{R},\mathcal{C}}$ Macaulay matrix with row multi-indices $\mathcal{R}$ and column multi-indices $\mathcal{C}$

$\boldsymbol{M}^{\mathcal{R}}$ Macaulay matrix with row multi-indices $\mathcal{R}$ and column multi-indices $\mathcal{R} \oplus \mathcal{S}_{\boldsymbol{L}}$

$\boldsymbol{L}_{\boldsymbol{\alpha}}$ Coefficient matrix of MEP associated with monomial $\boldsymbol{\lambda}^{\boldsymbol{\alpha}}$

$\boldsymbol{N}^{\mathcal{R},\mathcal{C}}$ Null space basis of Macaulay matrix with row multi-indices $\mathcal{R}$ and column multi-indices $\mathcal{C}$

$\boldsymbol{N}^{\mathcal{R}}$ Null space basis of Macaulay matrix with row multi-indices $\mathcal{R}$ and column multi-indices $\mathcal{R} \oplus \mathcal{S}_{\boldsymbol{L}}$

$\boldsymbol{b}(\boldsymbol{x})$ (Candidate set for) polynomial basis of $\mathbb{C}[\boldsymbol{x}]/I$

$\mathcal{R}$ Set of row multi-indices of a Macaulay matrix

$\mathcal{X}_d$ Set of points corresponding to the simplex $\{\boldsymbol{\rho} \in \mathbb{N}_0^m \mid \|\boldsymbol{\rho}\|_1 \leq d\}$

$\mathcal{B}_d^k$ Set of row multi-indices $\{\boldsymbol{\rho} \in \mathbb{N}_0^m \mid \|\boldsymbol{\rho}\|_\infty \leq d, \|\boldsymbol{\rho}\|_1 \leq k(d+1)-1\}$

$\mathcal{S}_{\boldsymbol{L}}$ Monomial support of $\boldsymbol{L}(\boldsymbol{x})$

$\boldsymbol{x}$ Variables $\boldsymbol{x} \in \mathbb{C}^m$

Contents

Popularized Abstract iii

Gepopulariseerde Samenvatting v

Abstract vii

Beknopte samenvatting ix

List of Abbreviations xiii

List of Symbols xvi

Contents xvii

List of Figures xxi

List of Tables xxiii

1 Introduction 1

1.1 Background and motivation 1

1.1.1 Multidimensional linear systems theory 2

1.1.2 System identification 3

1.1.3 Solving polynomial equations and multiparameter eigenvalue problems 10

1.2 Research objectives 13

1.3 Overview and Contributions 14

1.3.1 Part I: Systems theory and identification 14

1.3.2 Part II: Algorithms 17

1.3.3 Appendices 19

I Systems Theory and Identification 21

2 Multivariate Polynomials and Autonomous, Linear, Shift-invariant Models 23

2.1 Introduction . . . 24
2.2 Kernel representation . . . 26
2.2.1 Difference equations . . . 26
2.2.2 Homogenization, roots at infinity and the Macaulay matrix 27
2.3 Image representation . . . 31
2.3.1 Simple eigentuples . . . 31
2.3.2 Back to a kernel representation . . . 33
2.3.3 Multiplicities . . . 36
2.4 State-space representation . . . 38
2.4.1 Affine models . . . 38
2.4.2 Descriptor models . . . 43
2.4.3 Realization theory for descriptor systems . . . 48
2.5 Data-based realization . . . 50
2.5.1 Hankel matrix and its decomposition . . . 50
2.5.2 Order of the identified model . . . 52
2.6 Applications . . . 54
2.7 Conclusion . . . 56

3 Constructing Difference Equations using Generalized Cayley-Hamilton Theorem 56

3.1 Introduction . . . 58
3.2 The generalized Cayley-Hamilton Theorem . . . 62
3.3 Obtaining difference equations through the Generalized Cayley-Hamilton Theorem . . . 63
3.4 Closed-form expression . . . 68
3.5 Conclusion and future work . . . 70

4 Globally Optimal Misfit Identification of Multidimensional Systems 72

4.1 Introduction . . . 75
4.1.1 Problem formulation . . . 76
4.1.2 Contributions and outline . . . 78
4.2 Autonomous, shift-invariant, linear models . . . 80
4.2.1 Observability matrix . . . 80
4.2.2 Difference equations . . . 80
4.2.3 Generalized Cayley-Hamilton equations . . . 82
4.2.4 Macaulay matrix . . . 83
4.3 Polynomial characterization of the stationary points . . . 85
4.3.1 Image representation . . . 85
4.3.2 Misfit structure . . . 89

4.3.3 Kernel representation 90
4.4 Real models 93
4.4.1 Locally minimal real models 94
4.4.2 Model order of real minima 97
4.4.3 Real, local minima are isolated 99
4.4.4 Limitations 101
4.5 Lower-order models and infinite solutions 101
4.5.1 Polynomial formulations 102
4.5.2 Multiparameter eigenvalue problem 103
4.6 Numerical experiments 104
4.7 Conclusion and future work 107

II Algorithms 109

5 Exploiting the Block-Macaulay Matrix Structure 110
5.1 Introduction 112
5.2 The block-Macaulay framework 114
5.2.1 The block-Macaulay matrix 114
5.2.2 Null space and polynomial root finding 118
5.2.3 Shifting the row- and null space 121
5.3 Intersection-based null space computation 123
5.3.1 Avoiding construction of the Macaulay matrix 123
5.3.2 Decomposing the row indices for recursive computations 126
5.3.3 Computing bases for the intersections 130
5.4 Row space computation 132
5.5 Numerical experiments 133
5.5.1 General multiparameter eigenvalue problems 134
5.5.2 Application: globally optimal identification of autonomous LTI models 139
5.5.3 General discussion 142
5.6 Conclusion and future work 143

6 Direct Computation of Multiplication Maps 145
6.1 Introduction 146
6.2 Eigenvalue-based root finding 147
6.2.1 Multiplication maps 147
6.2.2 Variations on algebraic root finding 150
6.3 Basis-agnostic approach 152
6.4 Initialization and convergence criterion 155
6.4.1 Initialization 156
6.4.2 Reducing the equations 158
6.4.3 Convergence criterion 159

6.4.4 Building the multiplication maps . . . 161
6.5 Numerical experiments . . . 163
6.6 Conclusion . . . 166

7 Conclusion and future outlook 167
7.1 Overview of the contributions . . . 167
7.2 General considerations . . . 170
7.2.1 Engineering practice . . . 170
7.2.2 Other globally optimal approaches . . . 171
7.2.3 Computational strategies . . . 173
7.2.4 Low-rank generalized Hankel matrix approximation . . . 174
7.3 Future research . . . 174
7.3.1 Globally optimal parameter estimation . . . 175
7.3.2 Polynomial root finding . . . 176

A Jordan canonical form, commuting matrices and invariant subspaces 179
A.1 Jordan canonical form and invariant subspaces . . . 179
A.1.1 Construction of generalized eigenvectors . . . 180
A.1.2 Jordan chains and elementary invariant subspaces . . . 181
A.2 Joint Schur triangularization . . . 182
A.3 Jordan form and commutativity . . . 184
A.3.1 Decomposition into joint invariant subspaces associated with eigentuples . . . 184
A.3.2 Interaction with different Jordan chains . . . 185

Bibliography 189

List of publications 207

List of Figures

1.1 Misfit identification. 5
1.2 Misfit-versus-latency identification. 5
1.3 Local optimization methods and non-convexity. 7
1.4 Globally optimal system identification. 9
1.5 Illustration of homotopy continuation. 11
1.6 Overview of the dissertation. 15

2.1 Model representations and transitions. 25
2.2 Example of model output. 32
2.3 Linearly (in)dependent grid points. 36

4.1 Chapter overview. 79
4.2 Contour plot of 2D, first-order misfit problem. 89
4.3 Contour plot of misfit objective over real and complex model parameters. 96

5.1 Decomposition of row multi-indices of Macaulay matrix. 126
5.2 Runtime and residual comparison with MacaulayLab for varying degree block-Macaulay matrices. 135
5.3 Runtime and residual comparison with MacaulayLab for MEPs of varying degree. 137
5.4 Runtime and residual comparison with MacaulayLab for polynomials of varying dimension. 138
5.5 Runtime and residual comparison with MacaulayLab for misfit problems of varying data sizes. 141
5.6 Runtime and residual comparison with MacaulayLab for misfit problems of varying model order. 142

6.1 Visualization of border. 148
6.2 Runtime comparison with MacaulayLab. 164
6.3 Residual error comparison with MacaulayLab. 165

List of Tables

4.1 Real models that solve the image-based polynomial characterization for 2D, first-order misfit problem. 88
4.2 Complex solutions of image-based polynomial characterization for 2D, first-order misfit problem. 88
4.3 $\boldsymbol{g}$-values of solutions of kernel-based polynomial characterization for 2D, first-order misfit problem. 94
4.4 Real models that solve the image-based polynomial characterization for a 1D, second-order misfit problem. 95
4.5 Data for 2D, second-order misfit example problem. 105
4.6 Best fitting real models for 2D, second-order misfit problem. . . 105
4.7 Globally optimal model for 2D, second-order misfit problems with varying data. 106
4.8 Real models that solve image-based characterization for 3D, first-order misfit problem. 107

5.1 Number of rows of the computed null space for problems of varying degree. 138
5.2 Number of rows of the computed null space for problems of varying dimension. 139

Chapter 1

Introduction

1.1 Background and motivation

Sciences and engineering rely on a multitude of mathematical models to describe the behavior of real-world phenomena. These models help in analyzing, predicting, and sometimes even controlling the behavior of the underlying system. The ability to predict the behavior of a system under study, even to varying degrees of accuracy, generally leads to better-informed decisions, which simplifies design or allows one to more precisely control said system. This has the potential to reduce the need for time-consuming trial-and-error approaches, as well as expensive experiments or prototypes, allowing a more efficient allocation of resources in the process. As a consequence, it is both of interest to improve the accuracy of such model-based predictions, and to reduce the cost of obtaining the model itself through advancements in the modeling techniques used to acquire them.

The central theme of this dissertation is a particular such modeling approach: Distilling a dynamical model from a given data set of measurements, a practice known as *system identification*. In particular, we focus on the optimization problems underpinning this process, and we restrict our attention to a specific class of dynamic models: that of multidimensional (mD), linear, shift-invariant models. What sets the approach presented in this dissertation apart is the guarantee of a globally optimal parameter estimation step, whereas common approaches in the literature can only promise local optimality at best. Additionally, we touch upon computational tools that are useful for dealing with the class of mathematical objects central to the proposed modeling approach,

namely, multivariate polynomials.

In this chapter, we first provide a brief overview of the aforementioned model class and the practice of system identification, along with its connection to polynomials and polynomial root finding. Next, we continue by outlining the main research objectives of the thesis, followed by an overview of the text and the contributions therein.

1.1.1 Multidimensional linear systems theory

Many physical phenomena are governed by linear differential equations: well-known examples include Maxwell's equations, Newton's laws of motion, and Schrödinger's equation, which governs electromagnetism, mechanics, and quantum mechanics, respectively. Additionally, such models arise in economics [70, 93] and signal processing [22, 27, 91], among other fields. As such, linear models are fundamental to many applications. Furthermore, linear approximations are often used to analyze the behavior of nonlinear systems [121, 171] or to (adaptively) approximate them, often by linearizing around a stationary point, a strategy sometimes used in control.

To handle continuous models in practice, they are often discretized. One such discretization strategy is to approximate the differential operators by a finite differences approximation, leading to difference equations [83, 131]. This text exclusively deals with such discrete, linear systems that can be described by difference equations.

In the one-dimensional case, i.e., for so-called linear time-invariant systems (LTI), it is well known that such models can be represented in other ways than using differential or difference equations: They also allow a state-space representation, as well as a representation using transfer functions [90]. Linear systems have been generalized to the multidimensional setting from these different points of view: [6, 8, 87, 88, 98, 100, 162] study various state-space models, and [22, 56, 57, 130] study transfer functions. This dissertation focuses on models that stem from difference equations, which are most prevalent in the behavioral school of systems theory [22, 139, 140, 159, 160, 197, 205]. The connections between these different representations have also been studied: e.g. [58, 141] touch upon the state-space representations for difference equations-based models, and [55, 56, 87] do the same for models defined by state-space representations and transfer functions.

Using so-called *shift operators*, it becomes apparent that the sets of difference equations describing these linear models are essentially characterized by multivariate polynomial objects. These concepts are predominantly studied

in the fields of commutative algebra, module theory, and, above all, algebraic geometry [31, 32]. Consequently, the behavioral school of systems theory relies on many of the concepts from these at times highly abstract fields. This close link with multivariate polynomials is crucial to this work and will be employed extensively. More precisely, this text deals with finite-order, autonomous, scalar, linear, shift-invariant systems, which are fully characterized by sets of multivariate polynomials with a finite number of common roots.

1.1.2 System identification

Unfortunately, real-world problems are often not as straightforward. As such, deriving models through first principles is not always feasible. To circumvent this issue, models can be estimated from measurements of the system's in- and outputs, a practice known as *system identification*. Note that the obtained data is often 'noisy' due to various inaccuracies such as disturbances, unobserved inputs, and measurement errors. Because there are many unknowns, for starters, choosing a suitable model class, determining a dynamic model that adequately explains the behavior of the underlying (physical) system from such measurement data is somewhat of an art that involves multiple steps, often applied iteratively [120, 145].

Indeed, practical system identification first involves experiment design, where it is decided upon which inputs to apply during the data collection phase. This choice can affect the quality of the parameter estimates obtained in later phases [199]. Moreover, experiments and data collection may be expensive, so limiting oneself to as few well-designed experiments as possible is of interest. Next, the data is preprocessed, after which suitable model structures can be proposed. Clearly, choosing an appropriate model class is essential to obtaining a good result and often involves a delicate trade-off between the model complexity, as one wants to approximate the data sufficiently closely, without overfitting the noise within it. In the next phase, the model parameters for this particular structure are estimated. Furthermore, the performance of the obtained model should, of course, be validated on independent data, giving a sense of the predictive capabilities of the model. The insights obtained in this last phase can then be used to revisit the model structure selection phase, leading to an iterative process.

While it should be emphasized that all these steps are crucial for the overall success of this process, this dissertation focuses solely on the parameter estimation phase, which is often—but not uniquely—phrased as an optimization problem. We now provide a brief overview of the various parameter estimation techniques, concentrating on the multidimensional case.

State of the art and engineering practice

There are many different approaches to the parameter estimation phase. Firstly, there is subspace identification [183] and its mD extensions [59, 84, 153–156], which do not rely on any underlying measure of 'fit' to optimize. Instead, these methods use oblique projections and other numerical linear algebra manipulations, derived under certain statistical assumptions. Remark that nonparametric approaches to system identification exist as well, such as data-based approaches [125, 199] or transfer function estimates [145].

Optimization-based methods include prediction error methods (PEM). These aim to minimize the one-step-ahead prediction error, often deduced from the expected value of the next outputs, knowing previous inputs and outputs [3, 120, 182]. For LTI systems, these PEM methods also allow interpretation in the frequency domain [145]. Errors-in-variables methods [173], on the other hand, aim to minimize some weighted distance between the measured input-output sequences and an input-output sequence generated by the model, i.e., the *model-compliant sequence*. This can also be thought of as trying to minimally alter the observed data by so-called *misfits* until it becomes model-compliant, as briefly sketched in Figure 1.1. In this sense, this modeling approach attempts to find the model with the set of model-compliant sequences closest to the observed data. A subclass of this modeling approach is the output-error approach, which only minimizes the distance between the given output sequence and the model-compliant outputs [51, 52]. Misfit-versus-latency modeling is an extension to this approach that additionally considers the norm of a latent, unobserved input in its objective function [116]. This modeling setup is explained in some more detail in Figure 1.2. Misfit-based approaches, which are closely associated with behavioral modeling strategies [198], are considered in this dissertation.

Other approaches describe the mD system as a set of interconnected 1D models or a single, high-order 1D model [59, 71]. Similar strategies that on fitting finite-element approximations to the data [18] also exist. After this step, (nonlinear) 1D system identification techniques may be applied to obtain a finite-order model for the coefficients [28].

Applications

Multidimensional system identification arises in a multitude of applications, due to the broad applicability of linear models. A closely related field of study is control, where the predictive capabilities of such models are employed to guide systems to desired states. For more complex systems, it is often interesting to estimate the model dynamics from data, sometimes adaptively [40, 163, 200, 208].

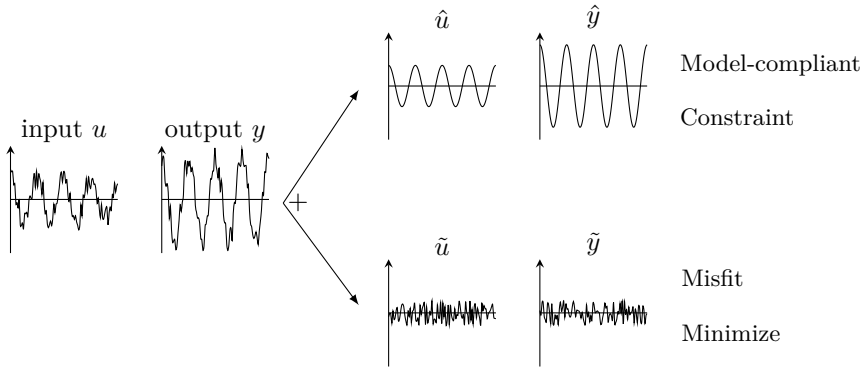


Figure 1.1: Illustration of how misfit identification alters the given data sequences in a 1D setting. The data is split up into a model-compliant component $\hat{u}, \hat{y}$, which is constrained to be generated by the selected model class, and a misfit component $\tilde{u}, \tilde{y}$, which one aims to minimize though selecting suitable model parameters.

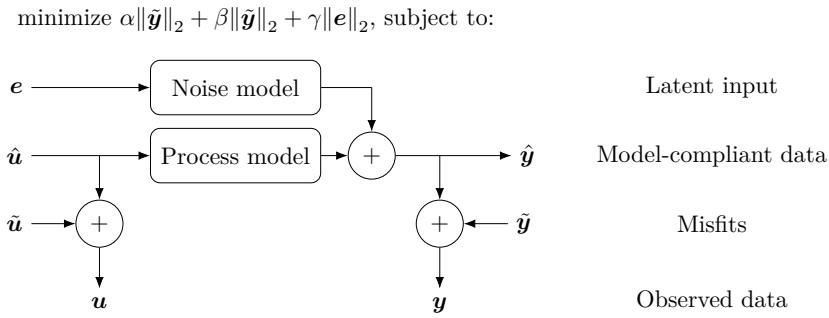


Figure 1.2: Diagram of misfit-versus-latency modeling. In contrast to misfit modeling, there is also a latent input, which models 'unobserved' effects that disturb the process. The goal is then to minimize some weighted sum of the norms of the misfits $\tilde{\boldsymbol{y}}, \tilde{\boldsymbol{u}}$ and the latent input $\boldsymbol{e}$, while constraining the model-compliant sequences $\hat{\boldsymbol{y}}, \hat{\boldsymbol{u}}$ and the latent sequence $\boldsymbol{e}$ to satisfy the chosen model.

More general applications for the parameter estimation of PDEs are also of interest [51, 144], as well as a variety of image and signal processing applications [22, 41, 91, 115, 155].

For the autonomous models considered in this thesis, the problem of multidimensional harmonic retrieval (MHR) is especially relevant. Here, one aims to identify exponential components in given data sequences [45, 86, 164, 168, 184, 185]. As will be detailed in Chapter 2, such exponential signal components are the fundamental outputs generated by the considered model class. MHR has been applied more specifically to various signal processing applications, for example, nuclear magnetic resonance spectroscopy [67, 119, 201], the analysis of wireless communications channels [69, 128], blind source separation [174], and direction of arrival problems [68], among others.

Local vs. global optimality

A common theme among all the outlined parameter estimation methodologies present in the literature is that they are all 'heuristic' to some degree: Some of these algorithms resort to statistical assumptions, which are generally hard to validate. Furthermore, in practice, optimization-based methodologies resort to local optimization algorithms to tackle the underlying problems. Such algorithms iteratively update a guess for the optimum, relying on various local approximations of the problem to obtain the next guess. This process is repeated until the first-order necessary conditions for optimality (FONC) are satisfied up to the desired precision. Because only local information is employed, these methods can at best guarantee the convergence to some local optimum. However, the parameter estimation problems under consideration are non-convex, implying that there can be, and there generally are, many local optima. Therefore, the outcomes of local optimization approaches are highly sensitive to the chosen initial estimate, as illustrated in Figure 1.3. Consequently, these procedures can in no way guarantee that the global optimum is found, which, in a mathematical sense, is the only point that solves the optimization problem exactly [138].

As only local optimality can be inferred from these commonly used approaches, this leaves some fundamental questions unanswered regarding these estimation problems: What did we converge to? Is it globally optimal? What does it mean to 'solve' such a problem? Is the obtained model unique? Moreover, the presence of randomness in the algorithm, e.g., a random initial guess, may affect the reproducibility of the results.

Contrary to these 'heuristic' approaches, this dissertation concerns a globally optimal strategy to tackle misfit identification problems for mD models. Previous

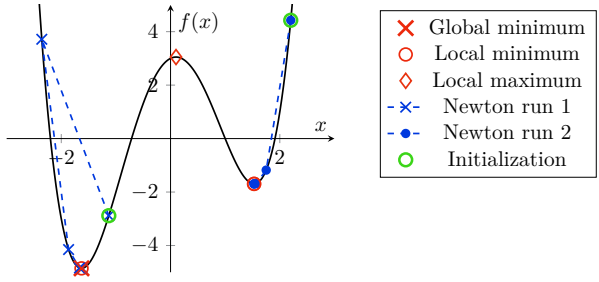


Figure 1.3: Brief illustration of a local optimization approach: we applied Newton's method to solve the FONC $\frac{df(x)}{dx} = 0$ of the non-convex objective $f(x) = x^4 - 5x^2 + x + 3$ for two different initial values: $x = -1.13$ and $x = 2.2$. While the algorithm converges towards the local minima, the final value strongly depends on the starting value. As such, one can, in general, at best guarantee that a local minimum is found, not necessarily the global minimum.

work has shown that for LTI models, globally optimal PEM identification [9], globally optimal ARMA modeling [189], and globally optimal misfit modeling [36] are all possible. The common approach used in these works hinges on the observation that LTI models are inherently polynomial in nature. This makes sure that the optimization problems arising in the parameter estimation phase are polynomial in nature as well, so that their first-order necessary conditions for optimality are multivariate polynomial equations. For these polynomial problems, off-the-shelf computational tools exist that can find all the common roots[1], which thus includes all local optima. This globally optimal strategy provides a deterministic methodology for optimal parameter estimation, of which Figure 1.4 sketches a brief overview.

Moreover, the resulting polynomial problems often have a particular algebraic structure, where one can discern 'auxiliary' variables that only appear linearly in the equations [36, 104, 189]. This structure allows them to be rephrased as a different type of polynomial problem: a so-called *multiparameter eigenvalue problem* (MEP). The goal of an MEP is to find the parameter values (eigenvalues) so that some polynomial matrix in said parameters has a nullity of at least one [188]. In this reformulation, these auxiliary variables are treated as eigenvectors, which lie in the null space of the polynomial matrix evaluated in the associated eigenvalue. Computationally, MEP-specific algorithms leverage this partially linear structure by first only computing the eigenvalues, after which the eigenvectors can be readily recovered from the related null spaces.

[1]To be more precise: all irreducible components of the solution set can be computed.

In some sense, this lowers the dimensionality of the problem, reducing its complexity compared to the original polynomial equations. MEPs also extend the polynomial eigenvalue problem by allowing multiple spectral parameters [35].

As this globally optimal methodology characterizes the stationary points, it also provides a natural framework to study their properties, where the properties of the minima are the most interesting for system identification. Indeed, it was noted in [36] that the optimal misfit sequences satisfy a condition akin to the one described by Walsh's Theorem in the context of $\mathcal{H}_2$-norm model-order reduction [157]. More precisely, it was observed that the optimal misfit sequence has the structure of a signal that was passed twice through the same FIR filter. This can also be viewed as an interpolation condition when considering the polynomial with the misfit sequence as its coefficients, which then interpolates the optimal poles with double their multiplicity. This property was formalized and leveraged to formulate a novel, simplified characterization of the minima in [103, 104].

The outlined globally optimal parameter estimation approach forms the central theme of this thesis: We investigate the extension of this methodology to the least-squares misfit identification problem for finite-order, autonomous, scalar, linear, shift-invariant models in general dimensions. We only consider the parameter estimation problem to be 'solved' when the global optimum can be obtained by means of a deterministic procedure. Additionally, to formulate and study the problem, we investigate the related theoretical properties of the considered model class and some of the computational tools involved in this approach.

Note that, while related to the aforementioned 'heuristic' approaches, these globally optimal methods solve an entirely different problem. Indeed, the former methods only aim to find a model that explains the data sufficiently well, instead of exactly solving the underlying optimization problem by finding the model that most closely fits the data. Clearly, the globally optimal approach is inherently more complex, since it computes all local minimizers, as well as additional, often non-interesting complex-valued roots. Since the number of solutions of polynomial equations grows rapidly as their degree and dimension increase, this approach imposes a significantly larger computational strain. As such, one cannot expect these approaches to be competitive with the prevailing locally optimal methodologies for practical problems. Nonetheless, these contributions demonstrate that global optimality can be achieved through deterministic means. Furthermore, these approaches may serve to benchmark the 'heuristic' approaches and provide theoretical insight into the optimization problems at hand. This may help improve such locally optimal methods, as was done in the

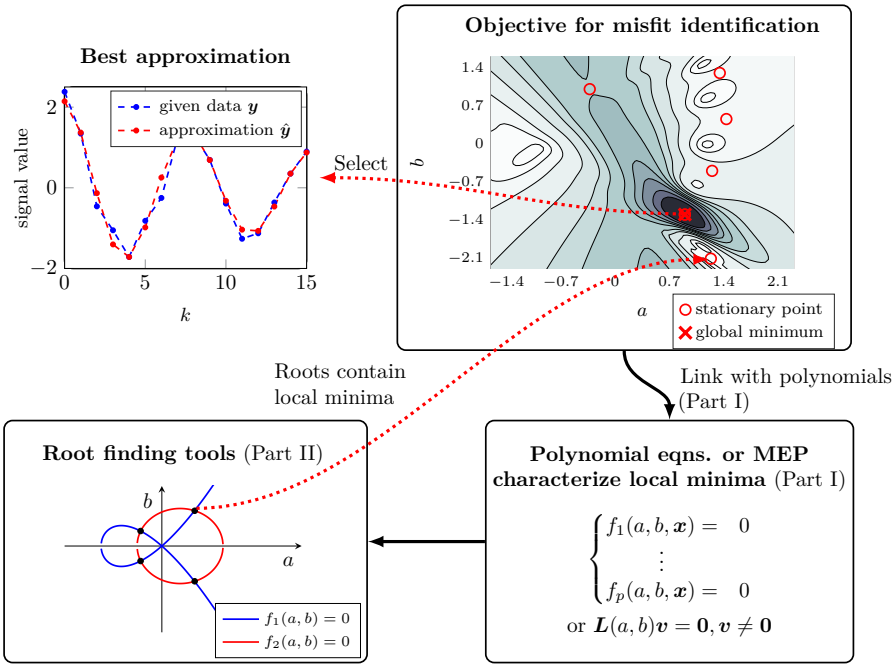


Figure 1.4: The strategy employed in globally optimal system identification. Given data $\boldsymbol{y}$, we search for the model whose output best approximates this data. For a polynomial model class and a least-squares objective, the first-order necessary conditions, which characterize all stationary points, are a set of polynomial equations or a multiparameter eigenvalue problem, where $\boldsymbol{x}$ and $\boldsymbol{v}$ are auxiliary variables. These equations may be further manipulated to eliminate non-interesting stationary points, as long as the resulting solution set still contains all local minima. For such polynomial problems, there exist algorithms that can find all the common roots, so that this set of points containing all the local minima, and thus the global minimum, can be obtained computationally. Finally, the global minimum optimum is selected by evaluating the objective function in each of the minima and picking the best-performing model.

LTI case in [206].

1.1.3 Solving polynomial equations and multiparameter eigenvalue problems

The aforementioned globally optimal approach to system identification hinges on the availability of qualitative numerical tools to find the common roots of the multivariate polynomials involved in the process. Furthermore, due to their polynomial nature, general computational tools to handle multivariate polynomials may also be applicable to the efficient manipulation of linear, shift-invariant systems. As previously mentioned, the polynomial root-finding problem characterizing the minima may be translated into an MEP in some cases, improving the computational efficiency by exploiting the linearity of some of the variables. Therefore, numerical solvers for both these types of problems are touched upon as well in the second part of this dissertation. We now give a brief overview of the state of the art in these fields.

Polynomial root-finding techniques can roughly be subdivided into two classes: algebraic techniques and homotopy-continuation methods. The latter type of method relies on numerical continuation techniques to continuously track the solutions of some known *start system* as it is deformed into the *target system* [24, 179, 192]. This procedure is illustrated using a simple univariate example in Figure 1.5. One of the key advantages of this approach is that each solution path can be tracked in parallel. A drawback is that it can only tackle 'square' sets of equations off the shelf, though a tailored *start system* might be provided to deal with non-square systems, if they are consistent, at least.

There are many flavors of algebraic techniques: The most popular symbolic methods are the variations on the Gröbner basis algorithms [26, 31, 53, 54, 110]. In some sense, these mimic Gaussian elimination by iteratively annihilating leading terms by taking polynomial combinations, rather than only linear combinations. Border basis approaches generalize this approach by removing the dependence on monomial orderings, which benefits numerical stability at the cost of tracking more polynomials [92, 133, 136, 137, 169, 181]. To be more precise, these techniques only compute a basis for the quotient ring defined by the polynomials, from which, at least for numerical root finding, a set of commuting multiplication matrices is derived. The common roots subsequently follow from a coupled eigenvalue problem involving these multiplication matrices [44, 169, 184, 185, 196]. Note that eigenvalue-based methods generally cannot deal with an infinite number of solutions, as the required eigenvalue problems would become infinitely large.

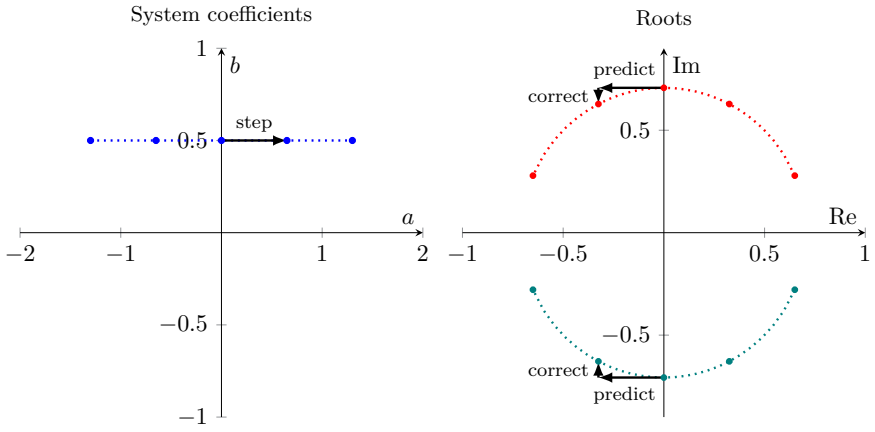


Figure 1.5: A brief illustration of the homotopy continuation method applied to a second-degree univariate polynomial equation $x^2 + ax + b = 0$. The roots of some known 'start equations' p_{start} are being tracked to the roots of the target equations p_{target} using continuation techniques. Differentiating $H(x(t), t) = tp_{\text{start}}(x(t)) + (1-t)p_{\text{target}}(x(t)) = 0$ then leads to an ordinary differential (ODE) equation for the roots trajectories $x(t)$. To find the new roots $x(t + \Delta t)$, a prediction for the new root is first made using approximation techniques for ODEs, e.g., Euler's method, resulting in a good initial approximation. Then, the corrector step applies a local root-finding algorithm, starting from said initial guess, and resulting in a better approximation of the actual root.

Resultant-based methods construct another polynomial, often the determinant of some structured matrix, in the coefficients of the polynomial equations to be solved. The properties of this new polynomial can then be leveraged to obtain the common roots of a given set of polynomials. Most famously, there is Macaulay's resultant and the associated u-resultant [31, 123]. Note that modern numerical techniques again extract eigenvalue problems from these constructions, so that these are subject to some of the same limitations. For a more complete overview of resultant-based methods, we refer the reader to [39, 63, 176]. From these structured matrices, other frameworks have risen, such as truncated normal forms [176–178] and the Macaulay framework [12, 43, 44]. In hopes of improving the numerical behavior, both these frameworks are mainly based on numerical linear algebra (NLA) steps. Remark that these approaches can also be viewed through the lens of Gröbner and border basis techniques, where they can be thought of as using different elimination strategies. Previous work has developed algorithms that rely on the specific problem structure

[11, 66, 191, 195]. These approaches can also be adapted to account for sparsity in the monomial support of the given equations using so-called toric rings and toric resultants [15, 16, 47, 175].

For MEPs, a distinction is made between square, a slightly different, but related problem, and rectangular MEPs, briefly introduced in the previous section as 'MEP'[2] [188]. This dissertation only considers the latter class of problem as it arises in system identification applications [36, 104, 189]. The Macaulay framework has also been extended to the block-Macaulay framework, tailored to solve these rectangular MEPs [188, 191]. Other solvers for this type of problem rely on first transforming the MEP to a square MEP through *randomized sketching* [81], which are subsequently solved through a system of coupled, structured generalized eigenvalue problems [80, 147]. Another approach translates the problem directly into such a coupled generalized eigenvalue problem through Kronecker products [4]. Both approaches apply to linear MEPs, though a linearization can be applied that defines auxiliary variables for nonlinear terms in the eigenvector, at the cost of larger coefficient matrices and spurious solutions [81, Section 5.2].

Currently, to the best of the author's knowledge, there are no homotopy-continuation algorithms available that exploit the linear structure of rectangular MEPs. For square MEPs, it must be noted that such methods do exist [42, 146, 161], so that the aforementioned transformations to square MEPs can be applied to tackle rectangular problems in this way. For a more complete overview of the methods concerning square MEPs, including optimization- and subspace-based methods, which only obtain a limited number of solutions, we refer the reader to [188].

The contributions within this dissertation only concern algebraic root-finding techniques, more specifically, the approach centered around multiplication maps, pioneered by Stetter and Auzinger [7, 169]. Indeed, as will become evident later on, there is a close connection between these approaches and multidimensional systems theory. Most notably, we focus on the Macaulay framework and its natural relation to multidimensional linear models. Homotopy continuation is used for purely computational purposes, however.

Of course, polynomial root finding and MEP solving have many more applications besides globally optimal system identification, though an in-depth treatment of these subjects is beyond the scope of this thesis. Firstly, similar strategies can be applied to any polynomial optimization problem [43, 73]. Other examples of fields of application include molecular biology [46], biochemical reaction networks [38], computing Nash equilibria [33], robotics [167, 194], computer vision [97, 158], and tensor decompositions [23, 99]. For MEPs in

[2]The term MEP without further distinction is used to refer to rectangular MEPs.

particular, the spectral analysis for PDEs is a common field of application [5, 148, 150, 180]. For a more elaborate overview, we refer the reader to [43, 176, 188].

1.2 Research objectives

The main goal of this dissertation is to expand and improve upon the deterministic, globally optimal system identification framework presented in [9, 36, 189]. Above all, we aim to extend the globally optimal approach for the misfit problem [36]. To this end, we explore the connection between multivariate polynomials and autonomous, scalar, linear, shift-invariant models; the properties of the mD misfit problem; and algebraic root-finding algorithms for multivariate polynomials and MEPs.

1. The main objective of this dissertation is to extend the globally optimal approach presented in [36] to the least-squares misfit identification of finite-order, autonomous, scalar, linear, shift-invariant models in general dimensions. This objective consists of several subgoals.
   (a) We aim to characterize the global optimum of the abovementioned least-squares misfit problem as one of the common roots of a polynomial problem.
   (b) To extend this problem to the kernel (difference equation) representation, a sufficiently general parameterization for the mD difference equations of an nth order model is required. This would extend the well-known characteristic equation in the 1D case.
   (c) We strive to extend this methodology to the case of multidimensional descriptor systems. To this end, the system-theoretic properties of such systems should be understood first.
2. Moreover, we aspire to leverage system-theoretic insights, similar to the ones that already arose in the 1D case, to characterize the global optimum more restrictively, thereby simplifying the polynomial characterization of the minima.
   (a) We seek to extend the aforementioned Walsh-like interpolation properties observed for the 1D case in [36] to this multidimensional problem.
   (b) We investigate whether such theoretic insights can be employed to arrive at more specific polynomial characterizations.

3. Finally, to further push the capabilities of this globally optimal modeling approach, it is of interest to extend the capabilities of root-finding algorithms. To this end, we explore whether certain substeps in the NLA-based root-finding approach of the Macaulay framework can be adjusted, improved or bypassed entirely. Note that this framework relies on computing the null space of a large, highly structured Macaulay matrix.

   (a) We intend to exploit this matrix structure to develop problem-specific algorithms to compute its null space bases, aiming to speed up the computation in the process.

   (b) We investigate whether, through this matrix structure, the explicit construction of the large Macaulay matrix, or blocks thereof, may be avoided entirely.

   (c) Border basis methods, through the lens of the Macaulay framework, only require part of the null space to obtain the multiplication maps. We aim to combine the insights from both frameworks to arrive at an algorithm that bypasses this expensive null space computation, while still relying solely on NLA-based computations and rank decisions.

1.3 Overview and Contributions

This dissertation is subdivided into two parts. The first part focuses on the intricate link between autonomous, scalar, linear, shift-invariant systems and multivariate polynomials, and addresses the system-theoretic and system-identification challenges in the research Objectives 1 and 2. The second part focuses on the computational aspects of research Objective 3. Figure 1.6 provides a schematic overview of both parts, the chapters within, and their interconnection. Remark that many of the individual chapters are written as independent contributions. This was done to ease the road to publication for the contents within, and make each chapter more self-sufficient with more clearly outlined contributions. This does, however, come at the cost of a greater repetitiveness concerning background material. Let us now provide a more in-depth overview of the contents within these chapters and their significance.

1.3.1 Part I: Systems theory and identification

Part I first addresses the system-theoretic challenges concerning multidimensional, autonomous, scalar, linear, shift-invariant models. More specifically, the system-theoretic interpretation of multivariate polynomials and the Macaulay

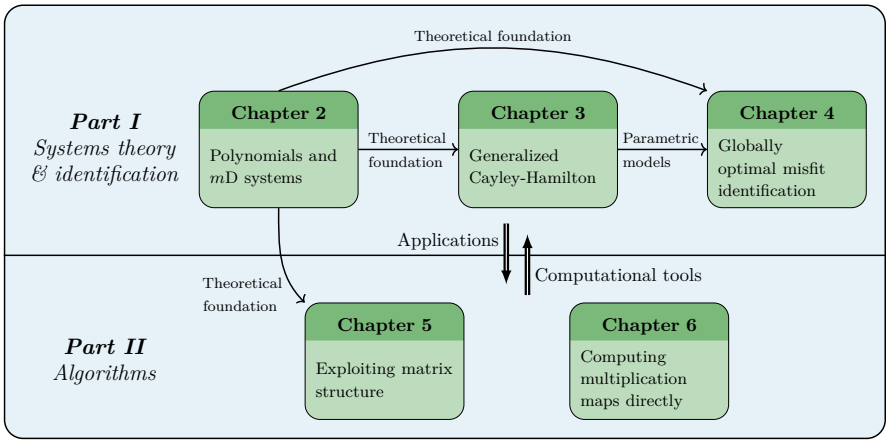


Figure 1.6: The dissertation is subdivided in two parts. Part one focuses on the system-theoretic aspects of multivariate polynomials and, most importantly, builds towards the main application of globally optimal misfit identification for mD systems. The second part then touches on the computational tools that are required for this globally optimal identification approach.

framework are outlined in Chapter 2, addressing the theoretical insights mentioned in Objective 1.c. Chapter 3 tackles the parameterization issue for the kernel representation stated in Objective 1.b. As suggested by Figure 1.6, this builds towards Chapter 4, where the misfit problem is tackled, building on the preceding insights. The latter chapter addresses Objectives 1.a, 2.a and 2.b.

Chapter 2

This chapter comprises a didactical literature study on the interpretation of multivariate polynomials with a finite solution set as autonomous, scalar, linear, shift-invariant models of finite-order. More precisely, the three main model representations are touched upon: the kernel representation, the image representation, and the state-space representations, as well as the transitions between them. Moreover, the less-established case of descriptor systems is tackled by relating descriptor models and their properties to affine models through a simple coordinate transform. While a treatment of the related projective *solutions at infinity* is present in the algebraic geometry literature, this is seldom viewed through a system-theoretic lens. As such, we prove that a rational state-space representation, similar to the shift equations found in the

former body of literature, precisely defines the model-compliant trajectories. Finally, a linear-algebra-based account of a generalization of the Popov-Belevitch-Hautus (PBH) test for these models is presented. This is more accessible to an engineering audience than the related, highly abstract results in the literature.

The contents of this chapter thus provide a didactical overview of the properties of finite-order, autonomous, scalar, linear, shift-invariant models in their most general form, i.e., descriptor models. As illustrated in Figure 1.6, this chapter serves as the theoretical foundation of this dissertation. It establishes the link between multivariate polynomials and shift-invariant models. This, in turn, is used extensively in Chapters 3 and 4 to extend the globally optimal identification approach, which heavily relies on this polynomial nature of the models. Moreover, this chapter provides grounds to extend these results to the case of descriptor models, as stated in Objective 1.c, through the theoretical treatment thereof. Finally, the insights of this chapter are used in Chapter 6 to compute projective multiplication maps, which are necessary for the correct termination of the algorithm.

Chapter 3

Here, a generalization of the Cayley-Hamilton theorem is applied to the state-space representation of the considered model class. This results in a novel way to obtain the difference equations directly from the state-space representation. Existing approaches in the literature first compute an image representation, from which difference equations are subsequently derived. Compared to these other approaches, the presented construction is more generally applicable to parametric models, that is, where the model parameters are variables. This makes it better suited for parameter estimation problems. Additionally, an explicit expression for these difference equations is provided.

As shown in Figure 1.6, the results of this chapter re-emerge in Chapter 4, where they provide a parameterization of the difference equations and the related kernel representation in terms of the eigentuples of the model. While the results have not been explicitly extended to the case of descriptor systems, the discussion in Chapter 2 allows one to readily relate any descriptor model to an affine model, apply the construction outlined in Chapter 3, and use this to derive difference equations for the descriptor model through a coordinate transform.

Chapter 4

The globally optimal methodology for the least-squares misfit problem for finite-order, autonomous, scalar, linear, shift-invariant models is tackled in this chapter. Novel polynomial characterizations akin to [36] are presented for both the image and kernel (difference equation) representation. Furthermore, the subtle duality between both characterizations is further explored, which remained absent in earlier work. The results within this chapter thus provide a complete, deterministic methodology to solve this problem exactly, free from heuristics and statistical assumptions. To this end, we show that the complex nature of the eigentuples, i.e., the model parameters, does not interfere with the presented globally optimal methodology.

Additionally, the properties of the local minima are investigated, leading to a Walsh-like interpolation condition, which forms a simpler polynomial characterization compared to the FONC. From this interpolation condition, the dual, kernel-based representation is eventually derived. This indicates that searching for problem-specific structure may be worthwhile for other identification problems as well. While this methodology is currently too computationally costly for practical problems, the obtained results can serve to benchmark locally optimal approaches, and the theoretical insights concerning the local minima may serve to further tailor such algorithms to this specific problem.

Moreover, we investigate the properties of the solution set of the resulting set of polynomial equations. It is illustrated that this set comprises positive-dimensional components, which limits the practical choice of solution algorithms for this particular problem. However, the isolatedness of the solutions corresponding to local minima is illustrated, ensuring that these points do indeed straightforwardly arise when solved numerically.

While the descriptor case is not explicitly tackled in this chapter, the insights of Chapter 2 may be used to compute the stationary points at infinity for this modeling problem. Moreover, the theory developed in said chapter provides a straightforward stepping stone to generalize these results to these more general models.

1.3.2 Part II: Algorithms

Part II of the dissertation focuses on some of the algorithmic aspects of numerical polynomial and MEP root finding. As these general tools may be employed for other purposes than globally optimal system identification as well, the topics

are presented somewhat disjointly from those in Part I for the sake of generality. Keep in mind that the presented tools not only apply for root finding purposes of the globally optimal identification procedure, as suggested in Figure 1.6, but also as NLA-based computational tools to handle linear, shift-invariant models. Chapter 5 proposes a way to leverage the highly structured nature of the (block)-Macaulay matrix to address Objectives 3.a and 3.b, leading to a novel algorithm to compute a null space basis for this matrix. Chapter 6 then investigates Objective 3.c, leading to a novel algorithm to compute the multiplication maps.

Chapter 5

In this chapter, it is investigated how the shift-invariant structure of the block-Macaulay matrix can be exploited to compute bases for its null and row space. The presented algorithm relies on reusing intermediate results by shifting them. These intermediate results are themselves null or row space bases associated with smaller block-rows of the Macaulay matrix. This 'shift' step is essentially a relabeling of the row or column indices, after which these bases associated with different block rows can be joined into a single basis for the larger, comprising matrix. For example, for the null space computation, this joining step amounts to performing a structured intersection operation.

While the preceding results provide a general framework to construct a family of algorithms relying on this shift-invariance, a specific choice of block-row decomposition is still required. A particular, disjoint row decomposition is proposed, one that can be applied recursively. Moreover, since we only rely on intermediate null or row space bases, the resulting algorithm operates without constructing the larger Macaulay matrix itself.

These algorithms can be used to tackle the MEPs associated with the misfit problem discussed in Chapter 4, though only for the case of LTI models, as these problems do not suffer from the infinite solution sets. It can also be applied to other globally optimal system identification problems [9, 189]. As it builds on the linear-algebra framework explained in Chapter 2, it can also be used to manipulate models represented by difference equations. Furthermore, it can be applied to more general polynomial problems and MEPs.

Chapter 6

This chapter combines insights of the border basis framework, which performs intermediate reductions on the generating set of polynomials, with those of

the Macaulay framework. This results in a novel algorithm to compute the multiplication maps associated with multivariate polynomials. The algorithm relies on shift equations to compress the information concerning multiplication, reducing the number of equations tracked compared to tracking the full Macaulay matrix null space. This equation, along with the reduction step to discover 'new' equations, is tackled purely through linear algebra, allowing the choice of underlying basis elements to be delegated to well-established NLA algorithms. Most importantly, the choice of new basis elements added in each iteration is based upon a rank decision and an accompanying matrix decomposition, a novel approach compared to the literature.

The presented algorithm provides a novel, simple, fully NLA-based approach to find the multiplication maps, avoiding the use of a Macaulay matrix null space. Compared to border bases, this allows for more flexibility in terms of choosing the underlying basis elements, due to the lower amount of symbolic bookkeeping. Again, these results may be applied to polynomial problems arising in system identification, or elsewhere, given that the solution set is not infinite. Moreover, it provides a more direct way to transition from a difference-equation-based model representation to a state-space representation.

1.3.3 Appendices

The state-space models and multiplication maps in this thesis all rely on commuting matrices. Appendix A contains a brief overview of some interesting properties of their eigenvalue structure. A proof for the well-known simultaneous Schur triangularization is provided. This is a useful property to better understand the intricate structure of the eigenvalues of commuting matrices and how they can be grouped into tuples. Furthermore, we briefly elaborate on the Jordan canonical form and invariant subspaces, and most importantly, how the eigenstructure of a set of commuting matrices creates a decomposition into such invariant subspaces. The latter two topics are not essential to understand the main results of this dissertation, though they provide useful and didactical insight into the (generalized) eigenstructure of state-space models and multiplication matrices.

Part I

Systems Theory and Identification

Chapter 2

Multivariate Polynomials and Autonomous, Scalar, Linear, Shift-Invariant Models of Finite Order

Chapter abstract

We explore the intricate link between multivariate polynomials and autonomous, scalar, linear, shift-invariant models. More precisely, the different model representations are studied for finite data sets, where the case of descriptor systems also has to be accounted for. This case is closely linked to homogeneous polynomials and roots at infinity. Most importantly, it is highlighted that any descriptor model can be associated with affine models—which have been well-studied—through a simple linear coordinate transform. This allows one to extend the kernel, image and difference equation model representations to the descriptor case, as well as the transitions between them. A linear-algebra-based generalization of the Popov-Belevitch-Hautus test to this model class is also presented.

Big picture

The current chapter serves as a prelude to the contributions in the remainder of this dissertation. It provides the theoretic foundation for much of the following chapters by highlighting how multivariate polynomials define autonomous, dynamic models and how the properties of polynomials extend to the properties of such models. These insights are crucial for Chapters 3 and 4. The former chapter tackles the transition from a state-space model to a difference-equation-based (kernel) representation, while the latter considers the extension of the globally optimal misfit identification framework to this multidimensional model class, relying on the polynomial nature to formulate a polynomial optimization problem. This chapter also tackles the case of descriptor models, which offers a first step towards generalizing the results within the next chapters to this more general case. Moreover, the different model representations and the transitions between them also provide the basis for the algorithms connected to algebraic root finding in Part II of the dissertation. The steps within these root-finding approaches can be interpreted as manipulations of the model class under consideration.

2.1 Introduction

This work explores the close connection between multivariate polynomials and autonomous, scalar, linear, shift-invariant models This class of models is a straightforward generalization of linear, time-invariant (LTI) models that can be characterized in three different ways: through the so-called kernel, image, and state-space representations [44, 58, 139], as visualized in Figure 2.1. While the results contained in this chapter mostly provide a system-theoretic interpretation to well-known results from algebraic geometry, these often rely on abstract notions from said field or module theory, making it less accessible to an engineering audience.

For affine models, such an interpretation has mostly been presented in the literature [44, 72]. However, the less-studied case of descriptor models is tackled as well. These models are connected to polynomials with so-called *solutions at infinity*, which are uncovered by homogenizing the polynomials. As such, the aim of this chapter is to introduce the model class under consideration in the most general sense, highlight its close link with multivariate polynomials, and provide a brief overview of the different model representations and the linear-algebra-based transitions between them. A rational state-space representation

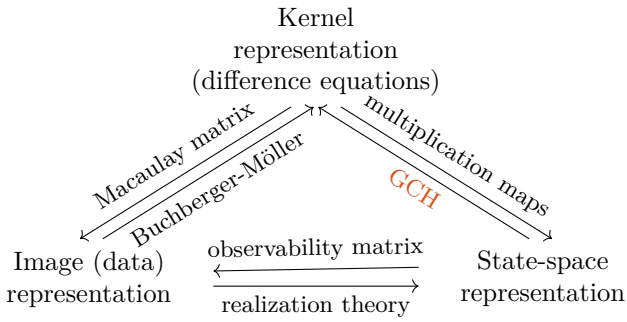


Figure 2.1: Different representations of autonomous, scalar, linear, shift-invariant models and the linear-algebra-based techniques to transition between them. The red color of the 'GCH' (Generalized Cayley-Hamilton) and 'multiplication maps' techniques indicates that this will not be discussed in the current chapter. Instead, it is a contribution that can be found in Chapters 3 and 6 of this dissertation, respectively.

for these descriptor models is provided. Finally, we briefly touch upon on the realization of such state-space models from model-compliant data, and provide an accessible linear-algebra-based proof for the generalization of the Popov-Belevitch-Hautus test for this class of models.

The remainder of this chapter is organized as follows: Section 2.2 considers the difference-equation-based kernel representation, the Macaulay matrix, and introduces the process of homogenization. Section 2.3 subsequently discusses the image representation and the Buchberger-Möller algorithm to go back to a kernel representation. These two sections mainly aim to provide a less formal overview of the literature, using examples to illustrate the concepts. Next, both state-space models and multidimensional realization theory are discussed in Section 2.4, starting with an overview for affine models. This theory is subsequently generalized to the case of descriptor models, wherein most of the contribution lies. Section 2.5 briefly touches upon the problem of data-based realization, and presents a novel, linear-algebra-based proof of a generalization of the Popov-Belevitch-Hautus test for the considered model class. Finally, Section 2.6 explores the connection with PDEs and mentions other applications, after which we finish with some concluding remarks.

2.2 Kernel representation

2.2.1 Difference equations

Discrete, multidimensional (mD) autonomous, single-output, multishift-invariant, linear models generate discrete mD signals, which exists on a discrete grid of multi-indices $\boldsymbol{\kappa} \in \mathbb{N}_0^m$. In this text, we focus on scalar signals. Any function that associates a grid point $\boldsymbol{\kappa}$ with some scalar numerical value can be viewed as a multidimensional (mD) signal[1] $y_{\boldsymbol{\kappa}} : \mathbb{N}_0^m \to \mathbb{R}^p; \boldsymbol{\kappa} \mapsto y_{\boldsymbol{\kappa}}$. Using so-called shift operators $\sigma_k : \sigma_k y_{\boldsymbol{\kappa}} = y_{\boldsymbol{\kappa}+\boldsymbol{e}_k}, k = 1, \ldots, m$, where $\boldsymbol{e}_k$ is the kth column of the identity matrix, the models under consideration can be characterized by a set of difference equations $\boldsymbol{r}(\boldsymbol{\sigma}) \in \mathbb{R}^q[\boldsymbol{\sigma}]$, where $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_m)$, which impose linear relations between the model-compliant signals at different grid points. Since the m shift operators commute by their definition, this set of q difference equations amounts to a set of q multivariate polynomials in $\boldsymbol{\sigma}$. Any model-compliant signal $\hat{y}_{\boldsymbol{\kappa}}$ then satisfies this set of equations at each grid point:

$$\forall \boldsymbol{\kappa} \in \mathbb{N}_0^m : \ \boldsymbol{r}(\boldsymbol{\sigma})\hat{y}_{\boldsymbol{\kappa}} = \boldsymbol{0}. \tag{2.1}$$

By definition, the collection of signals satisfying (2.1) then forms the kernel of the difference operator $\boldsymbol{r}(\boldsymbol{\sigma})$, so that this model characterization is referred to as the *kernel representation*. In the behavioral framework for systems theory, this space of model-compliant signals, i.e., the kernel of $\boldsymbol{r}(\boldsymbol{\sigma})$, is called *the behavior* $\mathfrak{B} = \ker \boldsymbol{r}(\boldsymbol{\sigma})$ [58, 139, 197, 205]. By the linearity of the shift operator $\boldsymbol{r}(\boldsymbol{\sigma})$, this set of model-compliant signals forms a vector space. This observation forms the basis for the image representation, which will be discussed in the next section. In this dissertation, we will restrict ourselves to such models with a finite-dimensional behavior. To clarify this abstract characterization, consider the example below.

Example 2.2.1. Consider the following set of second-degree difference equations:

$$\boldsymbol{0} = \begin{bmatrix} \hat{y}_{\boldsymbol{\kappa}+\boldsymbol{e}_1} - \hat{y}_{\boldsymbol{\kappa}+\boldsymbol{e}_1+\boldsymbol{e}_2} \\ \hat{y}_{\boldsymbol{\kappa}+\boldsymbol{e}_2} - \hat{y}_{\boldsymbol{\kappa}+\boldsymbol{e}_1+\boldsymbol{e}_2} \\ \hat{y}_{\boldsymbol{\kappa}+\boldsymbol{e}_1+\boldsymbol{e}_2} \end{bmatrix} = \begin{bmatrix} \sigma_1 - \sigma_1\sigma_2 \\ \sigma_2 - \sigma_1\sigma_2 \\ \sigma_1\sigma_2 \end{bmatrix} \hat{y}_{\boldsymbol{\kappa}} = \boldsymbol{r}(\boldsymbol{\sigma})\hat{y}_{\boldsymbol{\kappa}}.$$

Notice that $\boldsymbol{r}(\boldsymbol{\sigma})$ is a vector of polynomials in the 'variables' $\boldsymbol{\sigma} = (\sigma_1, \sigma_2)$. To obtain a model-compliant output signal, we can propose a solution signal of

[1] *Notation:* $\mathbb{N}_0$ designates the set of natural numbers, including zero.

the form $\hat{y}_{\boldsymbol{\kappa}} = \boldsymbol{\lambda}^{\boldsymbol{\kappa}} = \prod_{i=1}^{m} \lambda_i^{\kappa_i}$, where $\boldsymbol{\lambda} \in \mathbb{C}^m$, which, by (2.1), results in the equations:

$$\forall \boldsymbol{\kappa} \in \mathbb{N}_0^2 : \mathbf{0} = \boldsymbol{r}(\boldsymbol{\sigma})\hat{y}_{\boldsymbol{\kappa}} = \boldsymbol{r}(\boldsymbol{\sigma})\boldsymbol{\lambda}^{\boldsymbol{\kappa}}$$

$$= \begin{bmatrix} \boldsymbol{\lambda}^{\boldsymbol{\kappa}+(1,0)} - \boldsymbol{\lambda}^{\boldsymbol{\kappa}+(1,1)} \\ \boldsymbol{\lambda}^{\boldsymbol{\kappa}+(0,1)} - \boldsymbol{\lambda}^{\boldsymbol{\kappa}+(1,1)} \\ \boldsymbol{\lambda}^{\boldsymbol{\kappa}+(1,1)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\lambda}^{(1,0)} - \boldsymbol{\lambda}^{(1,1)} \\ \boldsymbol{\lambda}^{(0,1)} - \boldsymbol{\lambda}^{(1,1)} \\ \boldsymbol{\lambda}^{(1,1)} \end{bmatrix} \boldsymbol{\lambda}^{\boldsymbol{\kappa}} = \boldsymbol{\lambda}^{\boldsymbol{\kappa}} \boldsymbol{r}(\boldsymbol{\lambda}).$$

For complex-valued $\boldsymbol{\lambda}$, the above equations are satisfied if and only if $\boldsymbol{\lambda}$ satisfies the system of multivariate polynomial equations $\boldsymbol{r}(\boldsymbol{\lambda}) = \mathbf{0}$. These solutions $\boldsymbol{\lambda}$ will also be interchangeably referred to as *eigentuples* or (common) roots throughout this text. In this simple case, the $\boldsymbol{\lambda}^{(1,1)}$ term can be eliminated in the first two equations leading to the single solution $(\lambda_1, \lambda_2) = (0, 0)$ and the associated signal $\hat{y}_{\boldsymbol{\kappa}} = 0^{\kappa_1} 0^{\kappa_2}$, i.e., a 2D discrete impulse: $\hat{y}_{(0,0)} = 0$ and $\hat{y}_{\boldsymbol{\kappa}}$ is zero everywhere else. ◇

Example 2.2.1 demonstrates that difference equations give rise to multivariate exponential signals associated with their common roots. However, this construction only covers the case where the eigentuples are simple, i.e., all common roots are of multiplicity one. In case of a common root of higher multiplicity, said root will lead to the same number of linearly independent basis signals, each of a different format. Moreover, it can be shown that the kernel of $\boldsymbol{r}(\boldsymbol{\sigma})$ is precisely the span of these signals [19, 140, 142, 172]. Unfortunately, a formal proof of this statement would lead us too far.

Multiplicities are discussed in some more depth in Section 2.3.3. First, we look into the more practical case where only a finite number of grid points are considered.

2.2.2 Homogenization, roots at infinity and the Macaulay matrix

Example 2.2.1 illustrates how every common root of the multivariate polynomials $\boldsymbol{r}(\boldsymbol{\lambda})$ generates a model-compliant signal component $\hat{y}_{\boldsymbol{\kappa}} = \boldsymbol{\lambda}^{\boldsymbol{\kappa}}$. However, we often consider signals over a finite set of grid points. In this work, sets of the form $\mathcal{K} = \{\boldsymbol{\kappa} \in \mathbb{N}_0^m \mid \|\boldsymbol{\kappa}\|_1 \leq d\}$ are considered almost exclusively, that is, all grid points up to a certain 'degree' d. In this case, not only the affine roots, but the projective roots determine the number of solution 'modes' of the form $\boldsymbol{\lambda}^{\boldsymbol{\kappa}}$. To obtain the projective roots of a set of polynomials, which may contain so-called *roots at infinity*, we must consider their homogenization.

Polynomials can be homogenized by adding a new variable λ_0 as follows: $\lambda_0^d \boldsymbol{r}(\frac{\boldsymbol{\lambda}}{\lambda_0})$, where d is the degree of $\boldsymbol{r}(\boldsymbol{\lambda})$. This causes all terms of the resulting $m+1$-variate polynomial to be precisely of degree d. Conversely, a homogeneous polynomial is a polynomial where all nonzero terms are of the same total degree. To illustrate why these solutions at infinity must be accounted for, we reconsider the running example.

Example 2.2.2. Reconsider Example 2.2.1. The difference equations $\boldsymbol{r}(\boldsymbol{\sigma})\hat{y}_{\boldsymbol{\kappa}} = \mathbf{0}$ then describe linear relations among the grid points. As such, when we consider the finite set of points $\mathcal{K} = \{\boldsymbol{\kappa} \in \mathbb{N}_0^m \mid \|\boldsymbol{\kappa}\|_1 \leq 2\}$, these equations can be written out in a matrix format:

$$
\begin{array}{r}
\\
\boldsymbol{r}(\boldsymbol{\sigma})\hat{y}_{(0,0)} \\
\\
\\
\boldsymbol{r}(\boldsymbol{\sigma})\hat{y}_{(1,0)} \\
\\
\\
\boldsymbol{r}(\boldsymbol{\sigma})\hat{y}_{(0,1)} \\
\\
\end{array}
\begin{array}{c}
\begin{array}{cccccccccc}
(0,0) & (1,0) & (0,1) & (2,0) & (1,1) & (0,2) & (3,0) & (2,1) & (1,2) & (0,3)
\end{array} \\
\begin{bmatrix}
 & 1 & & & -1 & & & & & \\
 & & 1 & & -1 & & & & & \\
 & & & & 1 & & & & & \\
 & & & 1 & & & & -1 & & \\
 & & & & 1 & & & -1 & & \\
 & & & & & & & 1 & & \\
 & & & & 1 & & & & -1 & \\
 & & & & & 1 & & & -1 & \\
 & & & & & & & & 1 &
\end{bmatrix}
\end{array}
\begin{bmatrix}
\hat{y}_{(0,0)} \\ \hat{y}_{(1,0)} \\ \hat{y}_{(0,1)} \\ \hat{y}_{(2,0)} \\ \hat{y}_{(1,1)} \\ \hat{y}_{(0,2)} \\ \hat{y}_{(3,0)} \\ \hat{y}_{(2,1)} \\ \hat{y}_{(1,2)} \\ \hat{y}_{(0,3)}
\end{bmatrix}
= \boldsymbol{M}_{\boldsymbol{r}}\hat{\boldsymbol{y}} = \mathbf{0}.
$$

Solving this set of linear equations then results in the model-compliant sequences $\hat{\boldsymbol{y}}$ over $\mathcal{K}$. It may be verified that for any such solution it holds that $0 = \hat{y}_{(1,0)} = \hat{y}_{(0,1)} = \hat{y}_{(2,0)} = \hat{y}_{(1,1)} = \hat{y}_{(0,2)}$, while $\hat{y}_{(0,0)}, \hat{y}_{(3,0)}$ and $\hat{y}_{(0,3)}$ are clearly not restricted. On the other hand, when one considers these equations over all grid points instead, it holds that $0 = r_1(\boldsymbol{\sigma}) - r_3(\boldsymbol{\sigma})\hat{y}_{(2,0)} = \sigma_1\hat{y}_{(2,0)} = \hat{y}_{(3,0)}$, and, through a similar analysis, $0 = \hat{y}_{(0,3)}$. Clearly $\boldsymbol{M}_{\boldsymbol{r}}$ does not take these second-degree shifts of $\boldsymbol{r}(\boldsymbol{\sigma})$ into account, as these consider points outside of $\mathcal{K}$, causing two additional degrees of freedom for the solutions $\hat{\boldsymbol{y}}$.

This discrepancy is closely linked to the concept *solutions at infinity* of the underlying polynomials. To study these points, we homogenize the difference equation as follows by introducing a homogenization shift operator σ_0:

$$
\boldsymbol{r}(\sigma_0, \sigma_1, \sigma_2)\hat{y}_{\boldsymbol{\kappa}} = \begin{bmatrix} \lambda_0\boldsymbol{\sigma}^{(1,0)} - \boldsymbol{\sigma}^{(1,1)} \\ \lambda_0\boldsymbol{\sigma}^{(0,1)} - \boldsymbol{\sigma}^{(1,1)} \\ \boldsymbol{\sigma}^{(1,1)} \end{bmatrix} \hat{y}_{\boldsymbol{\kappa}} = \mathbf{0}. \tag{2.2}
$$

This technique forces all terms to be precisely of degree two in the shift operators $(\sigma_0, \sigma_1, \sigma_2)$, which nicely encodes that the second-degree equation $0 = \sigma_0\boldsymbol{\sigma}^{(1,0)}\hat{y}_{\boldsymbol{\kappa}}$ is formed through a linear combination involving difference equations of up to degree two. As such, it is not applicable to $\hat{y}_{(2,0)}$ for the considered data.

It then can be verified that the rows of $\boldsymbol{M_r}$ exactly describe the linear relations imposed by $\sigma_0\boldsymbol{r}(\sigma_0,\sigma_1,\sigma_2), \sigma_1\boldsymbol{r}(\sigma_0,\sigma_1,\sigma_2)$ and $\sigma_2\boldsymbol{r}(\sigma_0,\sigma_1,\sigma_2)$ over the homogenized grid of exactly degree 3 $\mathcal{K}_{\text{hom}} = \{\boldsymbol{\kappa} \in \mathbb{N}_0^{m+1} \mid \|\boldsymbol{\kappa}\|_1 = 3\}$. Indeed, the points in $\mathcal{K}$ and $\mathcal{K}_{\text{hom}}$ can be associated one to one with each other: $\forall \boldsymbol{\kappa} \in \mathcal{K} : (3 - \|\boldsymbol{\kappa}\|_1, \boldsymbol{\kappa}) \in \mathcal{K}_{\text{hom}}$ and $\forall \boldsymbol{\kappa} \in \mathcal{K}_{\text{hom}} : \boldsymbol{\kappa}_{1:m} \in \mathcal{K}$, i.e., we simply disregard κ_0. Taking this bijection between the grids into account, it can be seen that $\boldsymbol{M_r}$ implicitly considers these homogenized difference equations over $\mathcal{K}_{\text{hom}}$.

In similar fashion as before, signals of the form $(\lambda_0, \lambda_1, \lambda_2)^{(\kappa_0,\kappa_1,\kappa_2)}$, where $(\kappa_0, \kappa_1, \kappa_2) \in \mathcal{K}_{\text{hom}}$, can be proposed as candidate solutions to (2.2) over $\mathcal{K}_{\text{hom}}$. Working this out, keeping in mind that $\mathbf{0}$ does not represent a projective point[2], we conclude that $(\lambda_0, \lambda_1, \lambda_2)$ must satisfy the homogenized polynomial equations $\boldsymbol{r}(\lambda_0, \lambda_1, \lambda_2) = \mathbf{0}$, so that each projective solution leads to one specific exponential solution signal. Indeed, considering these equations as polynomials unveils that on top of the known root $(\lambda_0, \lambda_1, \lambda_2) = (1, 0, 0)$, there are two additional projective roots: $(0, 1, 0)$ and $(0, 0, 1)$. These lead to the two additional model-compliant signals previously observed. ◇

Example 2.2.2 illustrates how the projective roots come into play when compiling the difference equations in a specific matrix format. $\boldsymbol{M_r}$ belongs to the class of Macaulay matrices, which in the jargon of algebraic geometry can be viewed as a finite-dimensional representation of the polynomial ideal associated with the equations [32]. We now present a more formal definition of this class of matrices.

Definition 2.2.1 (Macaulay matrix)**.** Given a degree d polynomial, or equivalently, difference equation $r(\boldsymbol{\sigma}) = \sum_{\boldsymbol{\alpha}\in\mathcal{A}} c_{\boldsymbol{\alpha}}\boldsymbol{\sigma}^{\boldsymbol{\alpha}}$, where $\mathcal{A} = \{\boldsymbol{\alpha} \in \mathbb{N}_0^m \mid \|\boldsymbol{\alpha}\|_1 \leq d\}$ finite. The associated Macaulay matrix of degree d_t is then defined as follows, using multi-indices[3] for the block rows and columns:

$$(\boldsymbol{M_r})_{\boldsymbol{\rho},\boldsymbol{\kappa}} = c_{\boldsymbol{\rho}-\boldsymbol{\kappa}}, \text{ where } \boldsymbol{\kappa}, \boldsymbol{\rho} \in \mathbb{N}_0^m; \|\boldsymbol{\kappa}\|_1 \leq d_t, \|\boldsymbol{\rho}\|_1 \leq d_t - d,$$

where $\boldsymbol{c_\alpha}$ is considered zero when $\boldsymbol{\alpha} \notin \mathcal{A}$.

This definition can be naturally extended to a vector or set of difference equations by stacking the individual Macaulay matrices:

$$\boldsymbol{M_r} = \begin{bmatrix} \boldsymbol{M}_{r_1} \\ \vdots \\ \boldsymbol{M}_{r_q} \end{bmatrix}.$$

[2]In system theoretic terms, the eigentuple $\mathbf{0}$ defines the trivial zero trajectory, which does not contribute to the dimension of the model-compliant space. It should thus be ignored in the solution count.

[3]Multi-indices are used extensively to index matrix elements throughout this text. This is more convenient, as the specific ordering of the m-tuples is often of secondary importance.

Row permutations of such matrices may equivalently be considered Macaulay matrices. △

When $\boldsymbol{r}(\boldsymbol{\lambda})$ has a finite number of projective roots and given that d_t is large enough, the nullity of this matrix is equal to said number of roots, counting multiplicities. Moreover, in case these roots are simple, the right null space (kernel) of the matrix is spanned by the vectors that can be defined elementwise as $(\boldsymbol{v}_i)_{\boldsymbol{\kappa}} = \boldsymbol{\lambda}_i^{\boldsymbol{\kappa}}$, so-called *generalized Vandermonde vectors*, where $\boldsymbol{\lambda}_i, i \in \{1, \dots, n\}$ are the projective common roots of $\boldsymbol{r}(\boldsymbol{\lambda})$ [12, 44, 176, 188]. Since each root contributes one exponential solution mode, n is referred to as the *order* of the model. For roots with multiplicities, a single root contributes multiple such basis signals, as we will discuss later on. The Macaulay matrix offers a practical check to see if a signal complies with the model, as the common roots are often not known in practice. Moreover, this class of matrices can be employed to find said roots by applying multidimensional realization theory, to be discussed later on [12, 44, 188].

Let us briefly clarify the concept of projective roots. Projective space can be considered an extension to Euclidean space by adding 'points at infinity'. A projective 'point' is defined as the equivalence class

$$[\boldsymbol{p}] = \{a\boldsymbol{p} \mid a \in \mathbb{C}, \boldsymbol{p} \in \mathbb{C}^{m+1} \backslash \{\boldsymbol{0}\}\}$$

where any member of the set can be chosen to represent it: $[a\boldsymbol{p}] = [\boldsymbol{p}]$ for any a. Remark that $\boldsymbol{p} = \boldsymbol{0}$ is excluded and does not represent any projective point. For notational convenience, we will simply denote projective points by one of their representative coordinates in $\mathbb{C}^{m+1}$, dropping the brackets. For example, the coordinates $(1, 2, 3), (2, 4, 6)$ and $(3, 6, 9)$ all represent the same projective point. As such, the associated affine point in $\mathbb{C}^m$ can by retrieved by normalizing by the homogenization variable: $(\frac{p_1}{p_0}, \dots, \frac{p_m}{p_0})$. When $p_0 = 0$, the point is thus considered to lie at infinity.

While a set of homogeneous polynomials $\boldsymbol{r}(\boldsymbol{\lambda}), \boldsymbol{\lambda} \in \mathbb{C}^{m+1}$ of degree d does not define a function over the projective points—it takes different values when different representatives of the same point are considered: $\boldsymbol{r}(a\boldsymbol{\lambda}) = a^d \boldsymbol{r}(\boldsymbol{\lambda})$—its common roots are well-defined. As such, the common roots of a set of homogeneous polynomials correspond precisely to a set of projective points. For a more detailed exposition on the subject, we refer the reader to [12, 32, 176]. Regarding notation, $\boldsymbol{\lambda} \in \mathbb{P}^m$ will designate a projective point and $\boldsymbol{\kappa} \in \mathbb{N}_0^{m+1}$ will indicate a homogenized multi-index for the remainder of this chapter.

In 1D, LTI models with eigentuples at infinity are called *singular* or *descriptor systems*. Such systems do not arise without explicit homogenization, however. Moreover, using the Weierstrass canonical form, these models can be split into a causal ('forward') and anti-causal ('backward') part [60, 118].

Finally, we note that there exist techniques such as border and Gröbner that eliminate the solutions at infinity by computing a different set of equations with the same non-infinite roots [32, 133, 169]. Alternatively, components due to the solutions at infinity may be eliminated from the null space of a Macaulay matrix using a column compression [44], which is especially useful for polynomial root-finding purposes, where these roots are often not of interest. For the sake of completeness, in this chapter, these solutions will be considered, though.

2.3 Image representation

2.3.1 Simple eigentuples

In the previous section, it was already highlighted that the kernel of the difference equations, i.e., the behavior, forms a vector space that is finite-dimensional if the number of projective common roots is finite. Indeed, the shift operators are linear operators, so that any linear combination of model-compliant signals is model-compliant itself. As such, a basis for the behavior suffices to express all model-compliant signals. Moreover, it was demonstrated how such a basis can be constructed by considering exponential signals associated with the roots of the difference equations, i.e., the eigentuples. Focusing on the finite-grid case and difference equations with simple roots, when a large enough degree d is considered, any model-compliant signal can thus be written as an image of these basis signals, leading to the *image representation*:

$$\forall \boldsymbol{\kappa} \in \mathbb{N}_0^{m+1}, \|\boldsymbol{\kappa}\|_1 = d : \boldsymbol{\Gamma}_{\boldsymbol{\kappa},:} = \begin{bmatrix} \boldsymbol{\lambda}_1^{\boldsymbol{\kappa}} & \cdots & \boldsymbol{\lambda}_n^{\boldsymbol{\kappa}} \end{bmatrix}, \text{ so that } \hat{\boldsymbol{y}} = \boldsymbol{\Gamma}\boldsymbol{a},$$

where $\boldsymbol{a} \in \mathbb{C}^n$ and " : " indicates all indices. Note that the matrix $\boldsymbol{\Gamma}$ is a special case of an *observability matrix*, which is discussed later on.

Clearly, any signal of the above form always lies in the null space of $\boldsymbol{M}_{\boldsymbol{r}}$. However, to guarantee that range $\boldsymbol{\Gamma}$ equals this space generally requires one to consider degrees above the *degree of regularity*, where $\boldsymbol{\Gamma}$ will form a basis [12], e.g., see Example 5 in [186]. Recall that a κ_0-variable was added to homogenize the m-grid of degree lower than or equal to d, as explained in Example 2.2.2.

Similar to autonomous LTI models, this class of models also describes multivariate sinusoidal signals through complex-conjugate eigentuples. This is illustrated in Figure 2.2, where the output of a fourth-order 2D linear, shift-invariant model is visualized. Of course, when a finite grid is considered, these basis signals can be cast into a matrix format. As previously mentioned, the resulting generalized Vandermonde matrix then forms a basis for the right null space of an appropriately large Macaulay matrix associated with the difference

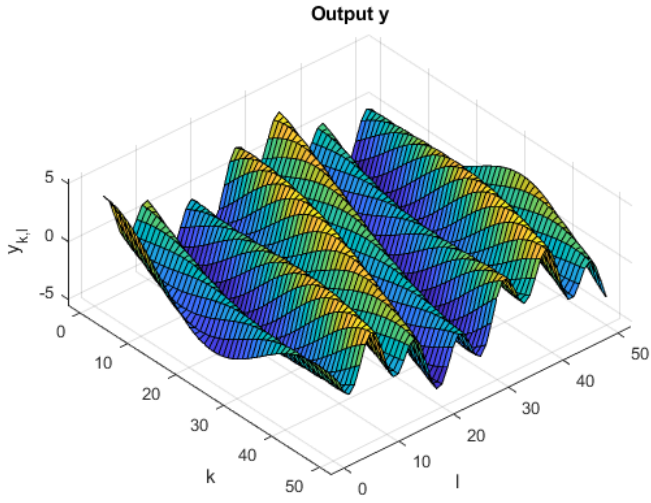


Figure 2.2: Output data of a model with eigentuples $(e^{i\frac{\pi}{4}}, e^{i\frac{\pi}{20}}), (e^{-i\frac{\pi}{4}}, e^{-i\frac{\pi}{20}}), (e^{i\frac{\pi}{4}}, e^{i\frac{\pi}{15}})$ $(e^{-i\frac{\pi}{4}}, e^{-i\frac{\pi}{15}})$. More precisely, $y_{(\kappa_1,\kappa_2)} = 2\cos\big(\kappa_1\frac{\pi}{4}\big)\cos\big(\kappa_2\frac{\pi}{20}\big) + 2\cos\big(\kappa_1\frac{\pi}{15}\big)\cos\big(\kappa_2\frac{\pi}{30}\big)$.

equations. Example 2.3.1 illustrates this dual representation for the running example.

Example 2.3.1. Reconsider Example 2.2.2 and the Macaulay matrix $\boldsymbol{M_r}$ within it. Then observe that the columns of the following matrix span the three exponential basis signals over the grid $\mathcal{K}_{\text{hom}}$, where $\boldsymbol{\lambda}_1 = (1,0,0), \boldsymbol{\lambda}_2 = (0,1,0)$ and $\boldsymbol{\lambda}_3 = (0,0,1)$:

$$\boldsymbol{\Gamma} = \begin{bmatrix} \boldsymbol{\lambda}_1^{(3,0,0)} & \boldsymbol{\lambda}_2^{(3,0,0)} & \boldsymbol{\lambda}_3^{(3,0,0)} \\ \boldsymbol{\lambda}_1^{(2,1,0)} & \boldsymbol{\lambda}_2^{(2,1,0)} & \boldsymbol{\lambda}_3^{(2,1,0)} \\ \boldsymbol{\lambda}_1^{(2,0,1)} & \boldsymbol{\lambda}_2^{(2,0,1)} & \boldsymbol{\lambda}_3^{(2,0,1)} \\ \boldsymbol{\lambda}_1^{(1,2,0)} & \boldsymbol{\lambda}_2^{(1,2,0)} & \boldsymbol{\lambda}_3^{(1,2,0)} \\ \boldsymbol{\lambda}_1^{(1,1,1)} & \boldsymbol{\lambda}_2^{(1,1,1)} & \boldsymbol{\lambda}_3^{(1,1,1)} \\ \boldsymbol{\lambda}_1^{(1,0,2)} & \boldsymbol{\lambda}_2^{(1,0,2)} & \boldsymbol{\lambda}_3^{(1,0,2)} \\ \boldsymbol{\lambda}_1^{(0,3,0)} & \boldsymbol{\lambda}_2^{(0,3,0)} & \boldsymbol{\lambda}_3^{(0,3,0)} \\ \boldsymbol{\lambda}_1^{(0,2,1)} & \boldsymbol{\lambda}_2^{(0,2,1)} & \boldsymbol{\lambda}_3^{(0,2,1)} \\ \boldsymbol{\lambda}_1^{(0,1,2)} & \boldsymbol{\lambda}_2^{(0,1,2)} & \boldsymbol{\lambda}_3^{(0,1,2)} \\ \boldsymbol{\lambda}_1^{(0,0,3)} & \boldsymbol{\lambda}_2^{(0,0,3)} & \boldsymbol{\lambda}_3^{(0,0,3)} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{matrix} (3,0,0) \\ (2,1,0) \\ (2,0,1) \\ (1,2,0) \\ (1,1,1) \\ (1,0,2) \\ (0,3,0) \\ (0,2,1) \\ (0,1,2) \\ (0,0,3) \end{matrix}.$$

Of course, any basis $\boldsymbol{\Gamma}$ for $\operatorname{null} \boldsymbol{M}_{\boldsymbol{r}}$ works just as well to represent the set of model-compliant signals: By linearity, any sequence $\hat{\boldsymbol{y}} = \boldsymbol{\Gamma}\boldsymbol{x}$ where $\boldsymbol{x} \in \mathbb{C}^n$ is model-compliant. This particular choice of basis only reveals the intricate connection between the projective eigentuples and the basis signals. In practice, these roots are not known, however, so one cannot directly construct this basis. On the contrary, the Macaulay framework uses a numerical basis to find the common roots of multivariate polynomial equations [12, 44, 188]. ◇

2.3.2 Back to a kernel representation

Due to the duality of the null and row space of a (Macaulay) matrix, returning to a kernel representation is then as simple as taking the left kernel of the basis matrix. However, this naive approach generally results in a set of difference equations of a large degree, which cannot be straightforwardly applied to smaller grids.

Alternatively, the (projective) Buchberger-Möller algorithm can be employed, leading to a reduced Gröbner basis for the ideal, which minimizes the degree of the difference equations for the considered graded ordering of the grid [1, 13, 14, 124, 129]. Generally, this algorithm eliminates the eigentuples at infinity. However, a simple change of coordinates suffices to move these eigentuples from 'infinity' to affine space, where they can be treated like normal common roots. In this new coordinate system, the Buchberger-Möller algorithm can then be applied to find a set of equations, the common roots of which correspond to the coordinate transforms of the original eigentuples. Reverting the coordinate transform then leads to equations in the original variables with the desired set of common roots, including those at infinity [124]. This process is illustrated in Example 2.3.2 below. Note that for any finite set of projective points, almost all linear coordinate transformations are such that none of the points lie at infinity in the new coordinate system[4].

Example 2.3.2. Reconsider the basis $\boldsymbol{\Gamma}$ from Example 2.3.1. Performing the invertible change of coordinates $(\lambda_0, \lambda_1, \lambda_2) = (\xi_0 + \xi_1, \xi_1 + \xi_2, \xi_0 + \xi_2) = \boldsymbol{P}\boldsymbol{\xi}$, with

$$\boldsymbol{P} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

[4]This can be seen as follows: Let $\boldsymbol{\xi}_0 = \boldsymbol{p} \begin{bmatrix} \boldsymbol{\lambda}_1 & \cdots & \boldsymbol{\lambda}_n \end{bmatrix}$, with $\boldsymbol{p} \in \mathbb{C}^m$. If this vector contains zero elements, there always exists an arbitrarily small perturbation on $\boldsymbol{p}$ such that these elements become nonzero, since $\boldsymbol{\lambda}_i \neq \boldsymbol{0}$.

such that $\boldsymbol{P}^{-1}\boldsymbol{\lambda}_i = \boldsymbol{\xi}_i$, leads to $\boldsymbol{\xi}_1 = (\frac{1}{2}, \frac{1}{2}, -\frac{1}{2}), \boldsymbol{\xi}_2 = (-\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$ and $\boldsymbol{\xi}_3 = (\frac{1}{2}, -\frac{1}{2}, \frac{1}{2})$. This allows us to rewrite $\boldsymbol{\Gamma}$ in as follows:

$$\boldsymbol{\Gamma} = \begin{bmatrix} (\boldsymbol{P}\boldsymbol{\xi}_1)^{(3,0,0)} & (\boldsymbol{P}\boldsymbol{\xi}_2)^{(3,0,0)} & (\boldsymbol{P}\boldsymbol{\xi}_3)^{(3,0,0)} \\ (\boldsymbol{P}\boldsymbol{\xi}_1)^{(2,1,0)} & (\boldsymbol{P}\boldsymbol{\xi}_2)^{(2,1,0)} & (\boldsymbol{P}\boldsymbol{\xi}_3)^{(2,1,0)} \\ (\boldsymbol{P}\boldsymbol{\xi}_1)^{(2,0,1)} & (\boldsymbol{P}\boldsymbol{\xi}_2)^{(2,0,1)} & (\boldsymbol{P}\boldsymbol{\xi}_3)^{(2,0,1)} \\ (\boldsymbol{P}\boldsymbol{\xi}_1)^{(1,2,0)} & (\boldsymbol{P}\boldsymbol{\xi}_2)^{(1,2,0)} & (\boldsymbol{P}\boldsymbol{\xi}_3)^{(1,2,0)} \\ (\boldsymbol{P}\boldsymbol{\xi}_1)^{(1,1,1)} & (\boldsymbol{P}\boldsymbol{\xi}_2)^{(1,1,1)} & (\boldsymbol{P}\boldsymbol{\xi}_3)^{(1,1,1)} \\ (\boldsymbol{P}\boldsymbol{\xi}_1)^{(1,0,2)} & (\boldsymbol{P}\boldsymbol{\xi}_2)^{(1,0,2)} & (\boldsymbol{P}\boldsymbol{\xi}_3)^{(1,0,2)} \\ (\boldsymbol{P}\boldsymbol{\xi}_1)^{(0,3,0)} & (\boldsymbol{P}\boldsymbol{\xi}_2)^{(0,3,0)} & (\boldsymbol{P}\boldsymbol{\xi}_3)^{(0,3,0)} \\ (\boldsymbol{P}\boldsymbol{\xi}_1)^{(0,2,1)} & (\boldsymbol{P}\boldsymbol{\xi}_2)^{(0,2,1)} & (\boldsymbol{P}\boldsymbol{\xi}_3)^{(0,2,1)} \\ (\boldsymbol{P}\boldsymbol{\xi}_1)^{(0,1,2)} & (\boldsymbol{P}\boldsymbol{\xi}_2)^{(0,1,2)} & (\boldsymbol{P}\boldsymbol{\xi}_3)^{(0,1,2)} \\ (\boldsymbol{P}\boldsymbol{\xi}_1)^{(0,0,3)} & (\boldsymbol{P}\boldsymbol{\xi}_2)^{(0,0,3)} & (\boldsymbol{P}\boldsymbol{\xi}_3)^{(0,0,3)} \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 3 & & 3 & & & 1 & & & \\ & 1 & 1 & 2 & 2 & & 1 & 1 & & \\ 1 & 2 & 1 & 1 & 2 & & & 1 & & \\ & & & 1 & 2 & 1 & 1 & 2 & 1 & \\ & 1 & 1 & 1 & 2 & 1 & & 1 & 1 & \\ 1 & 1 & 2 & & 2 & 1 & & & 1 & \\ & & & & & & 1 & 3 & 3 & 1 \\ & & & 1 & 2 & 1 & & 1 & 2 & 1 \\ & 1 & 1 & & 2 & 2 & & & 1 & 1 \\ 1 & & 3 & & & 3 & & & & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{\xi}_1^{(3,0,0)} & \boldsymbol{\xi}_2^{(3,0,0)} & \boldsymbol{\xi}_3^{(3,0,0)} \\ \boldsymbol{\xi}_1^{(2,1,0)} & \boldsymbol{\xi}_2^{(2,1,0)} & \boldsymbol{\xi}_3^{(2,1,0)} \\ \boldsymbol{\xi}_1^{(2,0,1)} & \boldsymbol{\xi}_2^{(2,0,1)} & \boldsymbol{\xi}_3^{(2,0,1)} \\ \boldsymbol{\xi}_1^{(1,2,0)} & \boldsymbol{\xi}_2^{(1,2,0)} & \boldsymbol{\xi}_3^{(1,2,0)} \\ \boldsymbol{\xi}_1^{(1,1,1)} & \boldsymbol{\xi}_2^{(1,1,1)} & \boldsymbol{\xi}_3^{(1,1,1)} \\ \boldsymbol{\xi}_1^{(1,0,2)} & \boldsymbol{\xi}_2^{(1,0,2)} & \boldsymbol{\xi}_3^{(1,0,2)} \\ \boldsymbol{\xi}_1^{(0,3,0)} & \boldsymbol{\xi}_2^{(0,3,0)} & \boldsymbol{\xi}_3^{(0,3,0)} \\ \boldsymbol{\xi}_1^{(0,2,1)} & \boldsymbol{\xi}_2^{(0,2,1)} & \boldsymbol{\xi}_3^{(0,2,1)} \\ \boldsymbol{\xi}_1^{(0,1,2)} & \boldsymbol{\xi}_2^{(0,1,2)} & \boldsymbol{\xi}_3^{(0,1,2)} \\ \boldsymbol{\xi}_1^{(0,0,3)} & \boldsymbol{\xi}_2^{(0,0,3)} & \boldsymbol{\xi}_3^{(0,0,3)} \end{bmatrix} = \boldsymbol{R}\boldsymbol{\Xi}$$

Keep in mind that any basis for range $\boldsymbol{\Gamma}$ suffices to perform the algorithm. Notice the similar structure of $\boldsymbol{\Gamma}$ and $\boldsymbol{\Xi}$. Indeed, both these matrices are special cases of *observability matrices*, to which we will get back in the next section. The latter of both observability matrices can then be used to construct a set of homogeneous polynomial equations in the new variables $\boldsymbol{\xi}$, after which the coordinate transform is reverted, and a set of polynomials in terms of $\boldsymbol{\lambda}$ is obtained. Also notice that, since the coordinate transform is invertible, so is $\boldsymbol{R}$, and, as a consequence, $\boldsymbol{\Gamma}$ and $\boldsymbol{\Xi}$ are of the same rank.

Let us now briefly illustrate the main ideas behind the Buchberger-Möller algorithm: First, an appropriate ordering for the multi-exponents, i.e., the grid points, is required. Here the graded inverse lexicographic ordering is used [32]. Then, through any (numerical) linear-algebra technique, the first $n = 3$ linearly independent rows of $\boldsymbol{\Xi}$ are determined. In this case, the rows associated with

the multi-indices $(3,0,0), (2,1,0)$ and $(2,0,1)$ are independent:

$$\mathbf{\Xi} = \frac{1}{8}\begin{bmatrix} 1 & -1 & 1 \\ 1 & 1 & -1 \\ -1 & 1 & 1 \\ 1 & -1 & 1 \\ -1 & -1 & -1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \\ -1 & 1 & 1 \\ 1 & 1 & -1 \\ -1 & 1 & 1 \end{bmatrix} \begin{matrix} (3,0,0) \\ (2,1,0) \\ (2,0,1) \\ (1,2,0) \\ (1,1,1) \\ (1,0,2) \\ (0,3,0) \\ (0,2,1) \\ (0,1,2) \\ (0,0,3) \end{matrix}.$$

Observe that any of the subsequent rows can then be written as linear combinations of these first three rows. These linear combinations lead to the following relations that hold in all $\boldsymbol{\xi}_i$: $\boldsymbol{\xi}^{(1,2,0)} - \boldsymbol{\xi}^{(3,0,0)} = 0$; $\boldsymbol{\xi}^{(1,1,1)} + \boldsymbol{\xi}^{(3,0,0)} + \boldsymbol{\xi}^{(2,1,0)} + \boldsymbol{\xi}^{(2,0,1)} = 0$ and $\boldsymbol{\xi}^{(1,0,2)} - \boldsymbol{\xi}^{(3,0,0)} = 0$. Now, since none of the eigentuples $\boldsymbol{\xi}_i$ lie at infinity, i.e., $\xi_0 \neq 0$, we can safely divide away the ξ_0 factor in each of these equations:

$$\begin{aligned} -\boldsymbol{\xi}^{(2,0,0)} + \boldsymbol{\xi}^{(0,2,0)} &= 0, \\ \boldsymbol{\xi}^{(2,0,0)} + \boldsymbol{\xi}^{(1,1,0)} + \boldsymbol{\xi}^{(1,0,1)} + \boldsymbol{\xi}^{(0,1,1)} &= 0, \\ -\boldsymbol{\xi}^{(2,0,0)} + \boldsymbol{\xi}^{(0,0,2)} &= 0. \end{aligned}$$

This essentially serves to dehomogenize the equations, as the classical form of the Buchberger-Möller algorithm constructs non-homogeneous bases.

Now, these linear expressions allow us to define a model-compliant signal at $(0,2,0), (0,1,1)$ and $(0,0,2)$ based on the preceding points $(2,0,0), (1,1,0)$, and $(1,0,1)$. This situation is illustrated in Figure 2.3. Note that by shifting these equations, all multiples of the dependent monomials can be expressed in terms of their predecessors as well. Indeed, the shifts of these polynomials with either ξ_0, ξ_1 or ξ_2, provide us with linear relations between the last four rows of $\mathbf{\Xi}$ and their predecessors, all of which could already be rewritten in terms of the first three rows. As such, a set of equations is obtained that recursively describes all rows in terms of the first three, providing a full description of $\mathbf{\Xi}$. This is what makes reduced Gröbner bases[5] so practical: for the considered grid ordering, they essentially provide us with 'initial conditions' and recursively applying the equations straightforwardly leads to the set of compatible signals.

[5]A similar observation applies to border bases.

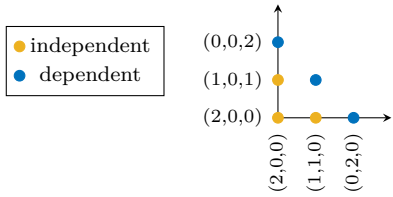


Figure 2.3: Visual representation of the linearly dependent and independent grid points in Example 2.3.2.

Now, the coordinate transform can be undone through substituting in $\boldsymbol{P}^{-1}\boldsymbol{\lambda} = \frac{1}{2}(\lambda_0 - \lambda_1 + \lambda_2, \lambda_0 + \lambda_1 - \lambda_2, -\lambda_0 + \lambda_1 + \lambda_2) = \boldsymbol{\xi}$ to obtain the sought-after equations in terms of $\boldsymbol{\lambda}_i$:

$$\boldsymbol{\lambda}^{(1,1,0)} - \boldsymbol{\lambda}^{(1,0,1)} = 0,$$

$$\boldsymbol{\lambda}^{(1,0,1)} = 0,$$

$$-\boldsymbol{\lambda}^{(1,0,1)} + \boldsymbol{\lambda}^{(0,1,1)} = 0.$$

Notice that linear combinations of these equations readily span the homogenization of the original equations (2.2) and vice versa, so that they are equivalent representations of the same behavior, that is, the projective common roots of both sets of equations coincide. These equations do not necessarily form a reduced Gröbner basis, however. ◇

2.3.3 Multiplicities

In the previous subsections, the assumption that $\boldsymbol{r}(\boldsymbol{\sigma})$ has simple common roots was made, that is, roots without multiplicities. Algebraically, the multiplicity of a common root corresponds to the number of linearly independent derivative evaluations that vanish in said root [169]. Moreover, this vector space of derivative and evaluation operators should be 'closed', that is, they should vanish on all polynomial multiples of the considered equations as well. Since a detailed treatment is of limited importance to the text, we refer the reader to [12, 34, 169] for more details regarding multiplicities.

In a system theoretic sense, the multiplicity of a common root is equal to the number of linearly independent basis signals associated with said root, which is closely related to the abovementioned operators [139, 140, 142, 172]. To illustrate this, let us substitute $\hat{y}_{\boldsymbol{\kappa}}$ by a signal of the form[6] $\frac{\partial \boldsymbol{\lambda}^{\boldsymbol{\kappa}}}{\partial \lambda_l} = \kappa_l \boldsymbol{\lambda}^{\boldsymbol{\kappa}-\boldsymbol{e}_l}$ into

[6]For $\lambda_l = 0$ this signal is undefined at $\kappa_l = \mathbf{0}$. In this case, let it be zero by convention.

the difference equations. To work this out, let us take a look at the action of σ_l^d on such a signal: $\sigma_l^d \kappa_l \boldsymbol{\lambda}^{\boldsymbol{\kappa}-\boldsymbol{e}_l} = (\kappa_l + d)\boldsymbol{\lambda}^{\boldsymbol{\kappa}+(d-1)\boldsymbol{e}_l} = \kappa_l \lambda_l^d \boldsymbol{\lambda}^{\boldsymbol{\kappa}-\boldsymbol{e}_l} + d\lambda_l^{d-1}\boldsymbol{\lambda}^{\boldsymbol{\kappa}}$, so that:

$$\forall \boldsymbol{\kappa} \in \mathbb{N}_0^m : \mathbf{0} = \boldsymbol{r}(\boldsymbol{\sigma})\kappa_l \boldsymbol{\lambda}^{\boldsymbol{\kappa}-\boldsymbol{e}_l} = \boldsymbol{r}(\boldsymbol{\lambda})\kappa_l \boldsymbol{\lambda}^{\boldsymbol{\kappa}-\boldsymbol{e}_l} + \frac{\partial \boldsymbol{r}(\boldsymbol{\lambda})}{\partial \lambda_l}\boldsymbol{\lambda}^{\boldsymbol{\kappa}},$$

if and only if both $\boldsymbol{r}(\boldsymbol{\lambda}) = 0$ and $\frac{\partial \boldsymbol{r}(\boldsymbol{\lambda})}{\partial \lambda_l} = 0$, that is, the common root $\boldsymbol{\lambda}$ has a multiplicity of at least two. Note that both equations can be thought of as applying a specific evaluation and derivative operators to $\boldsymbol{r}(\boldsymbol{\lambda})$. Multiple vanishing derivative operators, potentially of higher order, lead to even more such basis signals.

In general, the multiplicity structure of an eigentuple is closely tied to the number of linearly independent differential operators that vanish in said eigentuple while defining a 'closed' vector space w.r.t. polynomial multiplication [34, 132, 169, 204]. More complex multiplicity structures give rise to different signals in the kernel [12]. These are increasingly cumbersome to work out, however. Let us illustrate such a more complicated multiplicity structure through an example instead.

Example 2.3.3. Consider the difference equations below:

$$\boldsymbol{r}(\boldsymbol{\sigma})\hat{y}_{\boldsymbol{\kappa}} = \begin{bmatrix} \sigma_1^2 - 4\sigma_0\sigma_1 + 4\sigma_0^2 \\ \sigma_2^2 - 6\sigma_0\sigma_2 + 9\sigma_0^2 \end{bmatrix} \hat{y}_{\boldsymbol{\kappa}} = \mathbf{0}$$

It can be verified that $\boldsymbol{r}(1,2,3) = \frac{\partial \boldsymbol{r}(\boldsymbol{\lambda})}{\partial \lambda_1}|_{\boldsymbol{\lambda}=(1,2,3)} = \frac{\partial \boldsymbol{r}(\boldsymbol{\lambda})}{\partial \lambda_2}|_{\boldsymbol{\lambda}=(1,2,3)} = \frac{\partial^2 \boldsymbol{r}(\boldsymbol{\lambda})}{\partial \lambda_1 \partial \lambda_2}|_{\boldsymbol{\lambda}=(1,2,3)} = 0$. Since there are four linearly independent evaluation and differential operators associated with $(1,2,3)$, it is a common root with multiplicity four. For any finite degree, there is a finite number of monomials such that these operators can be easily cast into a vector format by applying them to said monomial basis. The resulting sequences can then also be interpreted as a basis signal, e.g., $\forall \boldsymbol{\kappa} : \hat{y}_{\boldsymbol{\kappa}} = \frac{\partial^2 \boldsymbol{\lambda}^{\boldsymbol{\kappa}}}{\partial \lambda_1 \partial \lambda_2}|_{\boldsymbol{\lambda}=(1,2,3)}$ is model-compliant. Over a third-degree grid, this gives us the following basis signals:

$$\begin{array}{cccc|c} \boldsymbol{\lambda}^{\boldsymbol{\kappa}}|_{\boldsymbol{\lambda}=(1,2,3)} & \frac{\partial \boldsymbol{\lambda}^{\boldsymbol{\kappa}}}{\partial \lambda_1}|_{\boldsymbol{\lambda}=(1,2,3)} & \frac{\partial \boldsymbol{\lambda}^{\boldsymbol{\kappa}}}{\partial \lambda_2}|_{\boldsymbol{\lambda}=(1,2,3)} & \frac{\partial^2 \boldsymbol{\lambda}^{\boldsymbol{\kappa}}}{\partial \lambda_1 \partial \lambda_2}|_{\boldsymbol{\lambda}=(1,2,3)} & \boldsymbol{\kappa} \\ \hline 1 & 0 & 0 & 0 & (3,0,0) \\ 2 & 1 & 0 & 0 & (2,1,0) \\ 3 & 0 & 1 & 0 & (2,0,1) \\ 4 & 4 & 0 & 0 & (1,2,0) \\ 6 & 3 & 2 & 1 & (1,1,1) \\ 9 & 0 & 6 & 0 & (1,0,2) \\ 8 & 12 & 0 & 0 & (0,3,0) \\ 12 & 12 & 4 & 4 & (0,2,1) \\ 18 & 9 & 12 & 6 & (0,1,2) \\ 27 & 0 & 27 & 0 & (0,0,3) \end{array}.$$

For this example, these vector representations then span the kernel of any associated Macaulay matrix[7]. Note that this can be readily generalized to more complex multiplicity structures [12, 44]. ◇

Note that the multiplicity of common roots is preserved by linear coordinate transforms $\boldsymbol{P}$: By the chain rule, $\mathbf{0} = \frac{\partial \boldsymbol{r}(\boldsymbol{\lambda})}{\partial \lambda_i} = \frac{\partial \boldsymbol{w}(\boldsymbol{\xi}(\boldsymbol{\lambda}))}{\partial \lambda_i} = \sum_{j=0}^{m} \frac{\partial \boldsymbol{w}(\boldsymbol{\xi})}{\partial \xi_j} \frac{\partial \xi_j}{\partial \lambda_i}$, where $\frac{\partial \xi_j}{\partial \lambda_i} = \frac{\partial (\boldsymbol{P}^{-1}\boldsymbol{\lambda})_j}{\partial \lambda_i} = (\boldsymbol{P}^{-1})_{j,i}$ are constants, implying that $\boldsymbol{w}$ also has a multiplicity in the solution $\boldsymbol{P}^{-1}\boldsymbol{\xi}$ whenever $\boldsymbol{\lambda}$ is a multiple root of $\boldsymbol{r}(\boldsymbol{\lambda})$. This argument can be extended to higher-order derivatives.

2.4 State-space representation

2.4.1 Affine models

Another possible representation for autonomous, scalar, linear, shift-invariant models is a *state-space representation*. Indeed, this representation is well established for affine mD systems [58], that is, those without eigentuples at infinity. Note that the affine case is of the greatest importance to the remainder of this dissertation and the most accessible sources concern this case [12, 44].

By the exposition in Section 2.3.2, any model with eigentuples at infinity, a so-called *descriptor model*, can always be transformed into an affine model by considering a simple change of coordinates $\boldsymbol{P}\boldsymbol{\xi} = \boldsymbol{\lambda}$, where $\boldsymbol{P} \in \mathbb{C}^{(m+1)\times(m+1)}$ is invertible, such that $\xi_0 \neq 0$ [1, 124]. This allows us to define alternative characterizing polynomials as $\boldsymbol{r}(\boldsymbol{\lambda}) = \boldsymbol{r}(\boldsymbol{P}\boldsymbol{\xi}) = \boldsymbol{w}(\boldsymbol{\xi})$, the common roots of which can be associated one to one with those of $\boldsymbol{r}(\boldsymbol{\lambda})$, since $\boldsymbol{P}$ is invertible. As such, the affine case is hardly restrictive, and a suitable state-space representation for the descriptor case is later derived from these well-known affine techniques. This construction stems from homogeneous multiplication matrices [15, 175]. For treatment of more general mD descriptor (also called singular or implicit models), linear state-space models, we refer the reader to [88, 117, 127].

Let us have a closer look at these affine techniques first. Models without eigentuples at infinity allow a state-space representation of the following form [12, 44, 58, 72], adapted to the homogenized grid where all grid points are of

[7] In this simple case, the nullity immediately stabilizes to the number of common roots counting multiplicities, i.e., four. In general, one may need a larger degree than that of the equations themselves.

the same 1-norm[8]:

$$\begin{aligned} \boldsymbol{x}_{\boldsymbol{\kappa}+\boldsymbol{e}_1} &= \boldsymbol{A}_1 \boldsymbol{x}_{\boldsymbol{\kappa}+\boldsymbol{e}_0} \\ &\vdots \\ \boldsymbol{x}_{\boldsymbol{\kappa}+\boldsymbol{e}_m} &= \boldsymbol{A}_m \boldsymbol{x}_{\boldsymbol{\kappa}+\boldsymbol{e}_0} \\ \hat{y}_{\boldsymbol{\kappa}} &= \boldsymbol{C} \boldsymbol{x}_{\boldsymbol{\kappa}} \end{aligned} \tag{2.3}$$

where $\boldsymbol{A}_k \boldsymbol{A}_l = \boldsymbol{A}_l \boldsymbol{A}_k \ \forall k, l \in \{1, \ldots, m\}$; $\boldsymbol{A}_k \in \mathbb{C}^{n \times n}, \boldsymbol{C} \in \mathbb{C}^{1 \times n}, \boldsymbol{x}_{\boldsymbol{\kappa}} \in \mathbb{C}^n$.

Given an initial state $\boldsymbol{x}_{d\boldsymbol{e}_0} = \boldsymbol{x}$, the output of such a model can then be written as $\hat{y}_{\boldsymbol{\kappa}} = \boldsymbol{C}\boldsymbol{A}^{\boldsymbol{\kappa}}\boldsymbol{x}$ for $\|\boldsymbol{\kappa}\|_1 = d$, with $\boldsymbol{A}^{\boldsymbol{\kappa}} = \prod_{k=0}^{m} \boldsymbol{A}_k^{\kappa_k}$ and $\boldsymbol{A}_0 = \boldsymbol{I}$. Note that the set of model-compliant outputs is invariant under a joint similarity transform on the *state-transition matrices*, that is, the model $\boldsymbol{A}_k^{\text{new}} = \boldsymbol{V}^{-1}\boldsymbol{A}_k\boldsymbol{V}, \boldsymbol{C}^{\text{new}} = \boldsymbol{C}\boldsymbol{V}, \boldsymbol{x}_{\boldsymbol{\kappa}}^{\text{new}} = \boldsymbol{V}^{-1}\boldsymbol{x}_{\boldsymbol{\kappa}}$ results in the same set of outputs, where $\boldsymbol{V} \in \mathbb{C}^{n \times n}$ is invertible.

The commutativity of the state-transition matrices ensures that they can all be put into an upper triangular form using the same Schur transformation[9], i.e., there exists a unitary $\boldsymbol{U}$ so that $\forall k : \boldsymbol{A}_k = \boldsymbol{U}\boldsymbol{T}_k\boldsymbol{U}^{\mathsf{T}}$ and $\boldsymbol{T}_k$ are upper triangular [29, 82, 165]. This joint triangular form unveils the eigenvalues, which makes it possible to group them into eigentuples as follows: $\boldsymbol{\xi}_i = ((\boldsymbol{T}_1)_{i,i}, \ldots, (\boldsymbol{T}_m)_{i,i})$. These eigentuples are exactly those described by an equivalent difference equation representation. Let us briefly delve into this statement.

For illustrative purposes, assume that the eigentuples are distinct. By the discussion in Appendix A.3, there then exists a similarity transform that jointly diagonalizes the state-transitions matrices [143], i.e., $\forall k : \boldsymbol{A}_k = \boldsymbol{V}\boldsymbol{D}_k\boldsymbol{V}^{-1}$, where $\boldsymbol{D}_k$ is diagonal. If we now plug a signal generated by a model of the form (2.3) into a set of difference equations, we obtain $\boldsymbol{0} = \boldsymbol{w}(\boldsymbol{\sigma})\boldsymbol{C}\boldsymbol{A}^{\boldsymbol{\kappa}}\boldsymbol{x} = \boldsymbol{C}\boldsymbol{w}(\boldsymbol{A}_0, \ldots, \boldsymbol{A}_m)\boldsymbol{A}^{\boldsymbol{\kappa}}\boldsymbol{x}$, which only vanishes for all $\boldsymbol{\kappa}$ when $\boldsymbol{w}(\boldsymbol{A}_0, \ldots, \boldsymbol{A}_m) = \boldsymbol{V}\boldsymbol{w}(\boldsymbol{D}_0, \ldots, \boldsymbol{D}_m)\boldsymbol{V}^{-1} = \boldsymbol{0}$. From this analysis it is clear that an equivalent difference equation description is characterized by a set of polynomials $\boldsymbol{w}(\boldsymbol{\sigma})$ with precisely the eigentuples of $(\boldsymbol{A}_0, \ldots, \boldsymbol{A}_m)$ as their common roots[10]. In the general case, i.e., even in the case of multiple common roots, this holds true as well. Indeed, given difference equations $\boldsymbol{w}(\boldsymbol{\sigma})$ and an equivalent state

[8] Replacing each element of $\mathcal{K}$ with its homogenized counterpart in $\mathcal{K}_{\text{hom}}$ naturally leads to the additional $+\boldsymbol{e}_0$ in the right-hand-side multi-index. Indeed, this ensures that the $1-$norm does not decrease upon a state-transition.

[9] We refer the interested reader to Appendix A.2 for a proof of this property, as well as an analysis of the structure of the eigenspaces of commuting matrices.

[10] If there were more roots, or multiplicities, the dimension of the space of model-compliant signals would be larger than n.

space model (2.3), it holds that $\boldsymbol{w}(\boldsymbol{A}_0, \ldots, \boldsymbol{A}_m) = \boldsymbol{0}$. This observation is the foundation of eigenvalue-based polynomial root finding, where these matrices are referred to as *multiplication maps* [169, 176, 188].

We now illustrate the above representation and show how it can be obtained in practice by applying *multidimensional realization theory* from an image representation [44].

Example 2.4.1. Reconsider the linear coordinate transform $\boldsymbol{P}$ from Example 2.3.2. As previously stated, this defines another set of polynomials $\boldsymbol{r}(\boldsymbol{P}\boldsymbol{\xi}) = \boldsymbol{w}(\boldsymbol{\xi})$, and thus an associated Macaulay matrix $\boldsymbol{M}_{\boldsymbol{w}} = \boldsymbol{M}_{\boldsymbol{r}}\boldsymbol{R}$. Indeed, it was already derived that $\boldsymbol{\Xi} = \boldsymbol{R}^{-1}\boldsymbol{\Gamma}$ forms a basis for all model-compliant trajectories with $\|\boldsymbol{\kappa}\|_1 = 3$, such that it naturally forms a basis for null $\boldsymbol{M}_{\boldsymbol{w}}$.

Due to the close connection with the affine eigentuples, a representation of the form (2.3) is most easily derived from normalized eigentuples, i.e., with $\xi_0 = 1$. Notice that scaling the eigentuples by a factor c scales the associated signal by a constant factor as well: $(c\boldsymbol{\xi})^{\boldsymbol{\kappa}} = c^{\|\boldsymbol{\kappa}\|_1}\boldsymbol{\xi}^{\boldsymbol{\kappa}}$, where $\|\boldsymbol{\kappa}\|_1$ is constant due to the homogeneous setting. As such, the signals $(c\boldsymbol{\xi})^{\boldsymbol{\kappa}}$ span the same set of model-compliant trajectories, i.e.,range $\boldsymbol{\Xi}$. As such, consider the normalization of the eigentuples $\boldsymbol{\xi}_1 = (1, 1, -1), \boldsymbol{\xi}_2 = (1, -1, -1)$ and $\boldsymbol{\xi}_3 = (1, -1, 1)$. The model-compliant signals can then be written as $\hat{y}_{\boldsymbol{\kappa}} = (1, 1, -1)^{\boldsymbol{\kappa}}x_1 + (1, -1, -1)^{\boldsymbol{\kappa}}x_2 + (1, -1, 1)^{\boldsymbol{\kappa}}x_3$. Since all ξ_0-values, i.e., the first coordinates of $\boldsymbol{\xi}_i$, are 1, their contributions in the exponentials can then essentially be ignored. By assigning the multivariate exponential terms to the state vector, a state-space model in the form of (2.3) can be defined by constructing the diagonal state-transition

matrices $\boldsymbol{A}_{\xi_k} = \mathrm{diag}((\boldsymbol{\xi}_1)_k, (\boldsymbol{\xi}_2)_k, (\boldsymbol{\xi}_3)_k)$:

$$\begin{aligned}
\boldsymbol{x}_{\boldsymbol{\kappa}+\boldsymbol{e}_1} &= \begin{bmatrix} (1,1,-1)^{\boldsymbol{\kappa}+\boldsymbol{e}_1} x_1 \\ (1,-1,-1)^{\boldsymbol{\kappa}+\boldsymbol{e}_1} x_2 \\ (1,-1,1)^{\boldsymbol{\kappa}+\boldsymbol{e}_1} x_3 \end{bmatrix} \\
&= \begin{bmatrix} 1 & & \\ & -1 & \\ & & -1 \end{bmatrix} \begin{bmatrix} (1,1,-1)^{\boldsymbol{\kappa}+\boldsymbol{e}_0} x_1 \\ (1,-1,-1)^{\boldsymbol{\kappa}+\boldsymbol{e}_0} x_2 \\ (1,-1,1)^{\boldsymbol{\kappa}+\boldsymbol{e}_0} x_3 \end{bmatrix} = \boldsymbol{A}_{\xi_1} \boldsymbol{x}_{\boldsymbol{\kappa}+\boldsymbol{e}_0}, \\
\boldsymbol{x}_{\boldsymbol{\kappa}+\boldsymbol{e}_2} &= \begin{bmatrix} (1,1,-1)^{\boldsymbol{\kappa}+\boldsymbol{e}_1} x_1 \\ (1,-1,-1)^{\boldsymbol{\kappa}+\boldsymbol{e}_1} x_2 \\ (1,-1,1)^{\boldsymbol{\kappa}+\boldsymbol{e}_1} x_3 \end{bmatrix} \\
&= \begin{bmatrix} -1 & & \\ & -1 & \\ & & 1 \end{bmatrix} \begin{bmatrix} (1,1,-1)^{\boldsymbol{\kappa}+\boldsymbol{e}_0} x_1 \\ (1,-1,-1)^{\boldsymbol{\kappa}+\boldsymbol{e}_0} x_2 \\ (1,-1,1)^{\boldsymbol{\kappa}+\boldsymbol{e}_0} x_3 \end{bmatrix} = \boldsymbol{A}_{\xi_2} \boldsymbol{x}_{\boldsymbol{\kappa}+\boldsymbol{e}_0}, \\
\boldsymbol{C} &= \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}.
\end{aligned}$$

To clearly distinguish between the state-transition matrices of this affine model and the ones of the descriptor model characterized by $\boldsymbol{r}(\boldsymbol{\sigma})$, we added the subscripts ξ_k, instead of only k.

Remark that not all choices of $\boldsymbol{C}$ are sufficient to fully describe all model-compliant trajectories: indeed, choosing $\begin{bmatrix} 0 & 1 & 1 \end{bmatrix}$ would not output the first exponential component, for example. However, there always exists a $\boldsymbol{C}$ that does lead to an n-dimensional space of trajectories.

When the eigentuples are unknown, the state-transition matrices can be obtained from any basis for the trajectories of degree d, given that d is large enough. In practice, 'large enough' means when the nullity of the associated Macaulay matrix stabilizes[11] with increasing degree d [12, 188]. In our example, it can be verified that $d = 3$ satisfies these conditions.

Now notice that the state-transition matrices, and, by extension, any joint similarity transform thereof, satisfy the following shift equations, obtained by

[11] When there are roots at infinity, a so-called 'gap' appears, which can be used to eliminate these roots using a column compression [44].

selecting the appropriate rows of $\boldsymbol{\Xi}$:

$$\begin{matrix}(3,0,0)\\(2,1,0)\\(2,0,1)\\(1,2,0)\\(1,1,1)\\(1,0,2)\end{matrix}\!\!\begin{bmatrix} 1 & -1 & 1\\ 1 & 1 & -1\\ -1 & 1 & 1\\ 1 & -1 & 1\\ -1 & -1 & -1\\ 1 & -1 & 1\end{bmatrix}\boldsymbol{T}\boldsymbol{T}^{-1}\boldsymbol{A}_{\xi_1}\boldsymbol{T} = \frac{1}{8}\begin{bmatrix} 1 & 1 & -1\\ 1 & -1 & 1\\ -1 & -1 & -1\\ 1 & 1 & -1\\ -1 & 1 & 1\\ 1 & 1 & -1\end{bmatrix}\!\!\begin{matrix}(2,1,0)\\(1,2,0)\\(1,1,1)\\(0,3,0)\\(0,2,1)\\(0,1,2)\end{matrix}\boldsymbol{T}, \quad (2.4)$$

$$\begin{matrix}(3,0,0)\\(2,1,0)\\(2,0,1)\\(1,2,0)\\(1,1,1)\\(1,0,2)\end{matrix}\!\!\begin{bmatrix} 1 & -1 & 1\\ 1 & 1 & -1\\ -1 & 1 & 1\\ 1 & -1 & 1\\ -1 & -1 & -1\\ 1 & -1 & 1\end{bmatrix}\boldsymbol{T}\boldsymbol{T}^{-1}\boldsymbol{A}_{\xi_2}\boldsymbol{T} = \frac{1}{8}\begin{bmatrix} -1 & 1 & 1\\ -1 & -1 & -1\\ 1 & -1 & 1\\ -1 & 1 & 1\\ 1 & 1 & -1\\ -1 & 1 & 1\end{bmatrix}\!\!\begin{matrix}(2,0,1)\\(1,1,1)\\(1,0,2)\\(0,2,1)\\(0,1,2)\\(0,0,3)\end{matrix}\boldsymbol{T}.$$

These equations also highlight the effect of a change of basis. Indeed, $\boldsymbol{T}$ may be any (unknown) invertible matrix, resulting in another basis for range $\boldsymbol{\Xi}$. Applying the shift equations above to this basis then suffices to determine $\boldsymbol{A}_{\xi_1}^{\text{new}} = \boldsymbol{T}^{-1}\boldsymbol{A}_{\xi_1}\boldsymbol{T}$ and $\boldsymbol{A}_{\xi_2}^{\text{new}} = \boldsymbol{T}^{-1}\boldsymbol{A}_{\xi_2}\boldsymbol{T}$, which can be used to build an equivalent state-space model, as previously indicated. Since all eligible state transition matrices are joint similarity transforms of $\boldsymbol{A}_{\xi_1}$ and $\boldsymbol{A}_{\xi_2}$, respectively, their eigentuples always correspond to the common roots of $\boldsymbol{w}(\boldsymbol{\xi})$. As such, it holds that $\boldsymbol{w}(\boldsymbol{A}_{\xi_1}^{\text{new}}, \boldsymbol{A}_{\xi_2}^{\text{new}}) = \boldsymbol{T}^{-1}\boldsymbol{w}(\boldsymbol{A}_{\xi_1}, \boldsymbol{A}_{\xi_2})\boldsymbol{T} = 0$. Remark that equations of this format also apply to systems with a more complex multiplicity structure. $\diamond$

Now that we have built some intuition on the inner workings of the state-space format, we briefly discuss how to return to an image representation starting from a state-space model. Recall that an image representation simply requires a basis for all the linearly independent signals over the considered grid. As such, to obtain all possible model outputs, one could consider the set of signals $\boldsymbol{C}\boldsymbol{A}^{\boldsymbol{\kappa}}\boldsymbol{x}$ generated by n linearly independent initial state vectors $\boldsymbol{x}$, e.g., the columns of the identity matrix. It can be worked out that the model-compliant output associated with any other initial state can then be written as a linear combination of these basis signals. Capturing these trajectories in a matrix format then results in an *observability matrix*.

Definition 2.4.1 (observability matrix)**.** Given a state-space model of the form (2.3), the corresponding observability matrix of degree d is then defined row-wise as:

$$\forall \boldsymbol{\kappa} \in \mathbb{N}^{m+1}, \|\boldsymbol{\kappa}\|_1 = d : \boldsymbol{\Gamma}_{\boldsymbol{\kappa},:} = \boldsymbol{C}\boldsymbol{A}^{\boldsymbol{\kappa}}. \qquad \triangle$$

The columns of such a matrix do not necessarily form a basis, however. In general, this observability matrix only guarantees to span the exact set of

trajectories found in the null space of a Macaulay matrix, once the degree of regularity is considered, an upper bound for which can be found in [12]. Computing such a matrix for the model considered in Example 2.4.1 results in a multiple of the basis $\boldsymbol{\Xi}$.

Note that the *shift-invariant structure* of $\boldsymbol{\Gamma}$ helps to shed some light on the shift equations (2.4) that were set up in Example 2.4.1. This allows us to easily obtain a general form for these equations, which can be solved for $\boldsymbol{A}_k$ whenever the left-hand-side coefficient matrix is of full column rank:

$$\forall k \in \{1, \ldots, m\}, \|\boldsymbol{\kappa}\|_1 = d - 1 : \boldsymbol{\Gamma}_{\boldsymbol{\kappa}+\boldsymbol{e}_0,:}\boldsymbol{A}_k = \boldsymbol{\Gamma}_{\boldsymbol{\kappa}+\boldsymbol{e}_k,:}. \tag{2.5}$$

Since $\boldsymbol{A}_0 = \boldsymbol{I}$, one can choose $\boldsymbol{C} = \boldsymbol{\Gamma}_{d\boldsymbol{e}_0,:}$, the first row of $\boldsymbol{\Gamma}$, to obtain a full state-space description.

2.4.2 Descriptor models

We now derive a state-space representation for descriptor models, i.e., models with eigentuples at infinity. First, the model-compliant trajectories of such a descriptor model are described in terms of an associated affine state-space model (2.3), which, as previously explained, can always be obtained by applying an appropriate coordinate transform to the difference equations. The state-transition matrices of this related affine model are $\boldsymbol{A}_{\boldsymbol{\xi}}$. Applying the change of coordinates $\boldsymbol{P}$ to $\boldsymbol{A}_{\boldsymbol{\xi}_j}$ then results in another set of state-transition matrices: $\forall k \in \{0, \ldots, m\} : \boldsymbol{A}_{\lambda_k} = (\sum_{l=0}^{m} \boldsymbol{P}_{k,l}\boldsymbol{A}_{\xi_l})$. Now observe that trajectories of the form $\hat{y}_{\boldsymbol{\kappa}} = \boldsymbol{C}\boldsymbol{A}_{\boldsymbol{\lambda}}^{\boldsymbol{\kappa}}\boldsymbol{x}$, where $\boldsymbol{A}_{\boldsymbol{\lambda}} = (\boldsymbol{A}_{\lambda_0}, \ldots, \boldsymbol{A}_{\lambda_m})$, comply with $\boldsymbol{r}(\boldsymbol{\sigma})$:

$$\forall \|\boldsymbol{\kappa}\|_1 = d - \deg(\boldsymbol{r}(\boldsymbol{\sigma})) : \boldsymbol{r}(\boldsymbol{\sigma})\hat{y}_{\boldsymbol{\kappa}} = \boldsymbol{C}\boldsymbol{r}(\boldsymbol{A}_{\boldsymbol{\lambda}})\boldsymbol{A}_{\boldsymbol{\lambda}}^{\boldsymbol{\kappa}}\boldsymbol{x} = \boldsymbol{C}\boldsymbol{w}(\boldsymbol{A}_{\boldsymbol{\xi}})\boldsymbol{A}_{\boldsymbol{\lambda}}^{\boldsymbol{\kappa}}\boldsymbol{x} = \boldsymbol{0},$$

since $\boldsymbol{w}(\boldsymbol{A}_{\boldsymbol{\xi}}) = \boldsymbol{0}$. Much like in Example 2.3.2, this definition of $\boldsymbol{A}_{\boldsymbol{\lambda}}$ also allows us to associate any signal of the form $\boldsymbol{C}\boldsymbol{A}_{\boldsymbol{\lambda}}^{\boldsymbol{\kappa}}\boldsymbol{x}$ with a signal $\boldsymbol{C}\boldsymbol{A}_{\boldsymbol{\xi}}^{\boldsymbol{\gamma}}\boldsymbol{x}$ by an invertible linear combination over the multi-indices $\boldsymbol{\gamma}$, where $\|\boldsymbol{\gamma}\|_1 = \|\boldsymbol{\kappa}\|_1$. As such, the former signals also span an n-dimensional set of trajectories, which form a basis for the $\boldsymbol{r}(\boldsymbol{\sigma})$-compliant signals of degree d. Let us illustrate this using the running example.

Example 2.4.2. Applying the linear transformation defined by $\boldsymbol{P}$ from Example 2.3.2 to the state-transition matrices $\boldsymbol{A}_{\xi_i}$ obtained in Example 2.4.1 we obtain:

$$\boldsymbol{A}_{\lambda_0} = \boldsymbol{A}_{\xi_0} + \boldsymbol{A}_{\xi_1} = \begin{bmatrix} 2 & & \\ & 0 & \\ & & 0 \end{bmatrix}$$

$$\boldsymbol{A}_{\lambda_1} = \boldsymbol{A}_{\xi_1} + \boldsymbol{A}_{\xi_2} = \begin{bmatrix} 0 & & \\ & -2 & \\ & & 0 \end{bmatrix}$$

$$\boldsymbol{A}_{\lambda_2} = \boldsymbol{A}_{\xi_0} + \boldsymbol{A}_{\xi_2} = \begin{bmatrix} 0 & & \\ & 0 & \\ & & 2 \end{bmatrix}.$$

As one would expect, the eigentuples $\boldsymbol{\lambda}_i = ((\boldsymbol{A}_{\lambda_0})_{i,i}, (\boldsymbol{A}_{\lambda_1})_{i,i}, (\boldsymbol{A}_{\lambda_2})_{i,i})$ of $(\boldsymbol{A}_{\lambda_0}, \boldsymbol{A}_{\lambda_1}, \boldsymbol{A}_{\lambda_2})$ are precisely the projective common roots[12] $\boldsymbol{\lambda}_1 = (2,0,0), \boldsymbol{\lambda}_2 = (0,-2,0)$ and $\boldsymbol{\lambda}_3 = (0,0,2)$ of $\boldsymbol{r}(\boldsymbol{\lambda})$. Choosing $\boldsymbol{C} = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$ as before, an observability matrix can be set up for this model, which forms a basis for the model-compliant signals, and thus for the null space of $\boldsymbol{M}_{\boldsymbol{r}}$:

$$\boldsymbol{\Gamma} = \begin{bmatrix} \boldsymbol{C}\boldsymbol{A}_{\boldsymbol{\lambda}}^{(3,0,0)} \\ \boldsymbol{C}\boldsymbol{A}_{\boldsymbol{\lambda}}^{(2,1,0)} \\ \boldsymbol{C}\boldsymbol{A}_{\boldsymbol{\lambda}}^{(2,0,1)} \\ \boldsymbol{C}\boldsymbol{A}_{\boldsymbol{\lambda}}^{(1,2,0)} \\ \boldsymbol{C}\boldsymbol{A}_{\boldsymbol{\lambda}}^{(1,1,1)} \\ \boldsymbol{C}\boldsymbol{A}_{\boldsymbol{\lambda}}^{(1,0,2)} \\ \boldsymbol{C}\boldsymbol{A}_{\boldsymbol{\lambda}}^{(0,3,0)} \\ \boldsymbol{C}\boldsymbol{A}_{\boldsymbol{\lambda}}^{(0,2,1)} \\ \boldsymbol{C}\boldsymbol{A}_{\boldsymbol{\lambda}}^{(0,1,2)} \\ \boldsymbol{C}\boldsymbol{A}_{\boldsymbol{\lambda}}^{(0,0,3)} \end{bmatrix} = \begin{array}{c} \begin{matrix} (2,0,0)^{\boldsymbol{\kappa}} & (0,-2,0)^{\boldsymbol{\kappa}} & (0,0,2)^{\boldsymbol{\kappa}} \end{matrix} \\ \begin{bmatrix} 8 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & -8 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 8 \end{bmatrix} \end{array}.$$

Notice that the range of this matrix corresponds exactly to the range of $\boldsymbol{\Gamma}$ in Example 2.3.1 and thus describes the same model. In Example 2.3.2 it, was already shown that each trajectory spanned by this basis can be associated with a trajectory spanned by $\boldsymbol{\Xi}$ and vice versa. ◇

Now that the model-compliant trajectories are known, a suitable state-space description remains to be found. Since there is no need anymore to distinguish between the descriptor system and the coordinate-transformed system, we drop

[12] Remember that representations of projective points can be freely scaled.

the $\lambda-$subscripts and denote $\boldsymbol{A}_{\lambda_k}$ by $\boldsymbol{A}_k$ from here on. To find the projective roots of a set of polynomials, a set of 'rational' shift equations has been suggested in the literature [12, 16]. However, it is not proven that these rational shift relations precisely define the model-compliant trajectories. We now address this caveat.

First, let us build some intuition with regard to where such a rational form comes from. In the affine case, $\boldsymbol{A}_0$ could be assumed to be $\boldsymbol{I}$, which, intuitively, implies that a shift in the 0th index can be inverted. For descriptor systems, $\boldsymbol{A}_0$ is not invertible, though. From Equation 2.4.2 it is evident that model-compliant state sequences are of the form $\boldsymbol{x}_{\boldsymbol{\kappa}} = \boldsymbol{A}^{\boldsymbol{\kappa}}\boldsymbol{x}$, such that the following equality can be derived $\boldsymbol{A}_k\boldsymbol{A}_l\boldsymbol{x}_{\boldsymbol{\kappa}} = \boldsymbol{A}_k\boldsymbol{x}_{\boldsymbol{\kappa}+\boldsymbol{e}_l} = \boldsymbol{A}_l\boldsymbol{x}_{\boldsymbol{\kappa}+\boldsymbol{e}_k}$. This leads to the state-space equations:

$$\forall k, l \in \{0, \ldots, m\}, \boldsymbol{\kappa} \in \mathbb{N}_0^{m+1}, \|\boldsymbol{\kappa}\|_1 = d-1 : \boldsymbol{A}_l\boldsymbol{x}_{\boldsymbol{\kappa}+\boldsymbol{e}_k} = \boldsymbol{A}_k\boldsymbol{x}_{\boldsymbol{\kappa}+\boldsymbol{e}_l}, \quad (2.6)$$

$$\text{where } \boldsymbol{A}_k\boldsymbol{A}_l = \boldsymbol{A}_l\boldsymbol{A}_k,$$

$$\hat{y}_{\boldsymbol{\kappa}} = \boldsymbol{C}\boldsymbol{x}_{\boldsymbol{\kappa}},$$

where the eigentuples of $(\boldsymbol{A}_0, \ldots, \boldsymbol{A}_m)$ are nonzero $(m+1)$-tuples. This last condition reflects that the eigentuples correspond to the projective roots of $\boldsymbol{r}(\boldsymbol{\sigma})$, which cannot include $\mathbf{0}$ by the definition of projective points.

By the previous derivation, it should be clear that all state trajectories of the form $\boldsymbol{A}_{\boldsymbol{\lambda}}^{\boldsymbol{\kappa}}\boldsymbol{x}$ satisfy (2.6). The converse—while not obvious—is also true, leading to the following theorem.

Theorem 2.4.1. *Any trajectory $\boldsymbol{x}_{\boldsymbol{\kappa}}$ that satisfies* (2.6) *is of the form $\boldsymbol{C}\boldsymbol{A}_{\boldsymbol{\lambda}}^{\boldsymbol{\kappa}}\boldsymbol{x}$.*

Proof. As previously stated, it only remains to be shown that any state sequence $\boldsymbol{x}_{\boldsymbol{\kappa}}$, where $\|\boldsymbol{\kappa}\|_1 = d$, that complies with (2.6) can be written in the form $\boldsymbol{A}_{\boldsymbol{\lambda}}^{\boldsymbol{\kappa}}\boldsymbol{x}$.

First, since the matrices $\boldsymbol{A}_k$ commute, by the discussion in Appendix A.3 [82, 143], there exists a basis $\boldsymbol{V}$ such that $\forall k : \boldsymbol{A}_k = \boldsymbol{V}\boldsymbol{J}_k\boldsymbol{V}^{-1}$, where the $\boldsymbol{J}_k$ are pairwise commutative, and each $\boldsymbol{J}_k$ is block-diagonal

$$\boldsymbol{J}_k = \begin{bmatrix} \boldsymbol{J}_k^{(1)} & & \\ & \boldsymbol{J}_k^{(2)} & \\ & & \ddots \end{bmatrix}.$$

Moreover, each set of blocks $(\boldsymbol{J}_1^{(s)}, \ldots, \boldsymbol{J}_m^{(s)})$ is associated with precisely one eigentuple of $(\boldsymbol{A}_1, \ldots, \boldsymbol{A}_m)$, and its dimension reflects the multiplicity of this eigentuple. Since these eigentuples correspond to projective points, there is

always at least one nonzero element in each eigentuple. This implies that for each tuple $(\boldsymbol{J}_1^{(s)}, \ldots, \boldsymbol{J}_m^{(s)})$ there exists an index i such that $\boldsymbol{J}_i^{(s)}$ has nonzero eigenvalues and is thus invertible.

Since $\boldsymbol{V}$ is invertible, we can associate each $\boldsymbol{x}_{\boldsymbol{\kappa}}$ one to one with another state sequence as follows $\boldsymbol{x}_{\boldsymbol{\kappa}} = \boldsymbol{V}\boldsymbol{v}_{\boldsymbol{\kappa}}$. Substituting $\boldsymbol{A}_k = \boldsymbol{V}\boldsymbol{J}_k\boldsymbol{V}^{-1}$ and left-multiplying both sides by $\boldsymbol{V}^{-1}$, (2.6) results in an equivalent model that describes $\boldsymbol{v}_{\boldsymbol{\kappa}}$:

$$\forall k, l \in \{0, \ldots, n\}, \|\boldsymbol{\kappa}\|_1 = d - 1 : \tag{2.7}$$

$$\begin{bmatrix} \boldsymbol{J}_k^{(1)} & & \\ & \boldsymbol{J}_k^{(2)} & \\ & & \ddots \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_{\boldsymbol{\kappa}+\boldsymbol{e}_l}^{(1)} \\ \boldsymbol{v}_{\boldsymbol{\kappa}+\boldsymbol{e}_l}^{(2)} \\ \vdots \end{bmatrix} = \begin{bmatrix} \boldsymbol{J}_l^{(1)} & & \\ & \boldsymbol{J}_l^{(2)} & \\ & & \ddots \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_{\boldsymbol{\kappa}+\boldsymbol{e}_k}^{(1)} \\ \boldsymbol{v}_{\boldsymbol{\kappa}+\boldsymbol{e}_k}^{(2)} \\ \vdots \end{bmatrix}.$$

This essentially decouples (2.6) into a set of similar, lower-dimensional equations. Choosing $\boldsymbol{C}^{\text{new}} = \boldsymbol{C}\boldsymbol{V}$ then outputs $\hat{y}_{\boldsymbol{\kappa}}$ compliant with (2.6).

As indicated, for each of these smaller models, there then exists an invertible state-transition matrix $\boldsymbol{J}_k^{(s)}$. Defining $\boldsymbol{J}^{(s)} = \left(\boldsymbol{J}_1^{(s)}, \ldots, \boldsymbol{J}_m^{(s)}\right)$ and assuming $\left(\boldsymbol{J}^{(s)}\right)^{d\boldsymbol{e}_k}$ is invertible, we can define $\boldsymbol{v}^{(s)}$ as the unique solution of $\boldsymbol{v}_{d\boldsymbol{e}_k}^{(s)} = \left(\boldsymbol{J}^{(s)}\right)^{d\boldsymbol{e}_k} \boldsymbol{v}^{(s)}$. By repeated application of (2.7), it holds that[13] $\left(\boldsymbol{J}^{(s)}\right)^{\boldsymbol{\gamma}} \boldsymbol{v}_{\boldsymbol{\kappa}}^{(s)} = \left(\boldsymbol{J}^{(s)}\right)^{\boldsymbol{\gamma}-\boldsymbol{e}_k} \left(\boldsymbol{J}_l^{(s)}\right) \boldsymbol{v}_{\boldsymbol{\kappa}-\boldsymbol{e}_l+\boldsymbol{e}_k}^{(s)} = \ldots = \left(\boldsymbol{J}^{(s)}\right)^{\boldsymbol{\kappa}} \boldsymbol{v}_{\boldsymbol{\gamma}}^{(s)}, \forall \boldsymbol{\gamma}, \boldsymbol{\kappa} \in \mathbb{N}_0^m$ where $\|\boldsymbol{\gamma}\|_1 = \|\boldsymbol{\kappa}\|_1 = d$. Choosing $\boldsymbol{\gamma} = d\boldsymbol{e}_k$ then ensures that $\left(\boldsymbol{J}^{(s)}\right)^{\boldsymbol{\gamma}} = \left(\boldsymbol{J}^{(s)}\right)^{d\boldsymbol{e}_k}$ is invertible, such that[14]:

$$\left(\boldsymbol{J}^{(s)}\right)^{d\boldsymbol{e}_k} \boldsymbol{v}_{\boldsymbol{\kappa}}^{(s)} = \left(\boldsymbol{J}^{(s)}\right)^{\boldsymbol{\kappa}} \boldsymbol{v}_{d\boldsymbol{e}_k}$$

$$\Leftrightarrow \boldsymbol{v}_{\boldsymbol{\kappa}}^{(s)} = \left(\boldsymbol{J}^{(s)}\right)^{\boldsymbol{\kappa}} \left(\boldsymbol{J}^{(s)}\right)^{-d\boldsymbol{e}_k} \boldsymbol{v}_{d\boldsymbol{e}_k}^{(s)} = \left(\boldsymbol{J}^{(s)}\right)^{\boldsymbol{\kappa}} \boldsymbol{v}^{(s)}.$$

Joining the contributions of over all s and reverting the similarity transform then concludes the proof. □

Note that the solution trajectory $\boldsymbol{x}_{\boldsymbol{\kappa}}$ of the state-transition equations in (2.6) remains a solution under right-multiplication of one of the equations by some

[13] Remark that, for this argument in the general case, all combinations of indices (k, l) should be 'exchangeable' with one another by some equation of the form (2.6). Indeed, eliminating some of the equations for our running example would result in more viable trajectories. When one of the $\boldsymbol{A}_k$ is invertible, e.g., $\boldsymbol{A}_0$ in the affine case, it can be verified that a subset of these equations implies the rest as well.

[14] note that $\boldsymbol{A}\boldsymbol{B} = \boldsymbol{B}\boldsymbol{A} \rightarrow \boldsymbol{B}\boldsymbol{A}^{-1} = \boldsymbol{A}^{-1}\boldsymbol{A}\boldsymbol{B}\boldsymbol{A}^{-1} = \boldsymbol{A}^{-1}\boldsymbol{B}\boldsymbol{A}\boldsymbol{A}^{-1} = \boldsymbol{A}^{-1}\boldsymbol{B}$

invertible $\boldsymbol{T}$. Moreover, the matrices $\boldsymbol{T}\boldsymbol{A}_k$ then do not necessarily commute, providing an alternative model description. However, Theorem 2.4.1 does, of course, not apply when the format of (2.6) is not respected.

We now illustrate the state space model (2.6) for the running example.

Example 2.4.3. Continuing from example 2.4.2 and the equality $\boldsymbol{A}_k\boldsymbol{A}_l\boldsymbol{x}_{\boldsymbol{\kappa}} = \boldsymbol{A}_k\boldsymbol{x}_{\boldsymbol{\kappa}+\boldsymbol{e}_l} = \boldsymbol{A}_l\boldsymbol{x}_{\boldsymbol{\kappa}+\boldsymbol{e}_k}$ leads to the following state-space model:

$$\boldsymbol{A}_0\boldsymbol{x}_{\boldsymbol{\kappa}+\boldsymbol{e}_1} = \begin{bmatrix} 2 & & \\ & 0 & \\ & & 0 \end{bmatrix} \begin{bmatrix} (x_1)_{\boldsymbol{\kappa}+\boldsymbol{e}_1} \\ (x_2)_{\boldsymbol{\kappa}+\boldsymbol{e}_1} \\ (x_3)_{\boldsymbol{\kappa}+\boldsymbol{e}_1} \end{bmatrix} = \begin{bmatrix} 0 & & \\ & -2 & \\ & & 0 \end{bmatrix} \begin{bmatrix} (x_1)_{\boldsymbol{\kappa}+\boldsymbol{e}_0} \\ (x_2)_{\boldsymbol{\kappa}+\boldsymbol{e}_0} \\ (x_3)_{\boldsymbol{\kappa}+\boldsymbol{e}_0} \end{bmatrix} = \boldsymbol{A}_1\boldsymbol{x}_{\boldsymbol{\kappa}+\boldsymbol{e}_0}$$

$$\boldsymbol{A}_0\boldsymbol{x}_{\boldsymbol{\kappa}+\boldsymbol{e}_2} = \begin{bmatrix} 2 & & \\ & 0 & \\ & & 0 \end{bmatrix} \begin{bmatrix} (x_1)_{\boldsymbol{\kappa}+\boldsymbol{e}_2} \\ (x_2)_{\boldsymbol{\kappa}+\boldsymbol{e}_2} \\ (x_3)_{\boldsymbol{\kappa}+\boldsymbol{e}_2} \end{bmatrix} = \begin{bmatrix} 0 & & \\ & 0 & \\ & & 2 \end{bmatrix} \begin{bmatrix} (x_1)_{\boldsymbol{\kappa}+\boldsymbol{e}_0} \\ (x_2)_{\boldsymbol{\kappa}+\boldsymbol{e}_0} \\ (x_3)_{\boldsymbol{\kappa}+\boldsymbol{e}_0} \end{bmatrix} = \boldsymbol{A}_2\boldsymbol{x}_{\boldsymbol{\kappa}+\boldsymbol{e}_0}$$

$$\boldsymbol{A}_2\boldsymbol{x}_{\boldsymbol{\kappa}+\boldsymbol{e}_1} = \begin{bmatrix} 0 & & \\ & 0 & \\ & & 2 \end{bmatrix} \begin{bmatrix} (x_1)_{\boldsymbol{\kappa}+\boldsymbol{e}_1} \\ (x_2)_{\boldsymbol{\kappa}+\boldsymbol{e}_1} \\ (x_3)_{\boldsymbol{\kappa}+\boldsymbol{e}_1} \end{bmatrix} = \begin{bmatrix} 0 & & \\ & -2 & \\ & & 0 \end{bmatrix} \begin{bmatrix} (x_1)_{\boldsymbol{\kappa}+\boldsymbol{e}_2} \\ (x_2)_{\boldsymbol{\kappa}+\boldsymbol{e}_2} \\ (x_3)_{\boldsymbol{\kappa}+\boldsymbol{e}_2} \end{bmatrix} = \boldsymbol{A}_1\boldsymbol{x}_{\boldsymbol{\kappa}+\boldsymbol{e}_2}$$

$$\hat{y}_{\boldsymbol{\kappa}} = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \boldsymbol{x}_{\boldsymbol{\kappa}}. \tag{2.8}$$

It may be verified that trajectories $\boldsymbol{x}_{\boldsymbol{\kappa}} = \boldsymbol{C}\boldsymbol{A}^{\boldsymbol{\kappa}}$ are the only valid solutions when considering these equations over a grid of the form $\mathcal{K} = \{\boldsymbol{\kappa} \in \mathbb{N}_0^3 \mid \|\boldsymbol{\kappa}\|_1 = d\}$.

When different types of grids are considered, however, trajectories other than the multi-exponentials associated with the eigentuples may arise as solutions. For example, let us consider the grid points $(2,0,0), (1,1,0), (1,0,1), (0,1,1)$. Writing out the equations above defining $(x_2)_{\boldsymbol{\kappa}}$, that is, the second row of the state equations in (2.8), over these grid points results in the following linear equations:

$$\begin{bmatrix} 1 & 0 & & \\ & & 1 & 0 \\ 0 & & 0 & \\ & 0 & & 0 \\ & 0 & 1 & \end{bmatrix} \begin{bmatrix} (x_2)_{(2,0,0)} \\ (x_2)_{(1,1,0)} \\ (x_2)_{(1,0,1)} \\ (x_2)_{(0,1,1)} \end{bmatrix} = \mathbf{0}$$

Clearly, the sequence $\begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^\mathsf{T}$ satisfies this set of equations, while this does not comply with the associated multi-exponetial $(x_2)_{\boldsymbol{\kappa}} = (0,1,0)^{\boldsymbol{\kappa}}$. ⋄

2.4.3 Realization theory for descriptor systems

The techniques of affine realization theory have to be slightly adapted to successfully deal with state-transition matrices with eigentuples at infinity. Generally, to account for a non-identity $\boldsymbol{A}_0$ in the definition of $\boldsymbol{\Gamma}$, the right-hand side of equation (2.5) must be multiplied by $\boldsymbol{A}_0$ [12]. Moreover, a similar shift equation holds for all index pairs $(k, l) \in \{0, \ldots, m\}^2$, not just pairs involving the 0th index. This results in an equation with two unknown shift matrices, which can be solved for the single unknown $\boldsymbol{A}_k^{-1}\boldsymbol{A}_l$. However, in general, all the $\boldsymbol{A}_k$ may be singular, as in the running example. Moreover, since $\boldsymbol{\Gamma}_{\boldsymbol{\kappa}+\boldsymbol{e}} = \boldsymbol{\Gamma}_{\boldsymbol{\kappa}}\boldsymbol{A}_k$, a singular state-transition matrix $\boldsymbol{A}_k$ also causes the coefficient matrix on the other side of the equation to be rank-deficient as well.

Nevertheless, as long as there is a finite number of projective roots, these issues can be circumvented using the techniques from [15, 16, 175]. From the definition of the observability matrix for descriptor models, the shift equations (2.5) can be adjusted, from which a set of invertible shift matrices and corresponding coefficient matrices can be obtained through linear combinations:

$$\begin{aligned}\forall k, l \in \{0, 1, \ldots, m\}, \|\boldsymbol{\kappa}\|_1 = d-1 :\ & \boldsymbol{\Gamma}_{\boldsymbol{\kappa}+\boldsymbol{e}_l,:}\boldsymbol{A}_k = \boldsymbol{\Gamma}_{\boldsymbol{\kappa}+\boldsymbol{e}_k,:}\boldsymbol{A}_l \qquad (2.9)\\ \Rightarrow & \boldsymbol{\Gamma}_{\boldsymbol{\kappa}+\boldsymbol{e}_l,:}\sum_{k=0}^{m} f_k\boldsymbol{A}_k = \sum_{k=0}^{m} f_k\boldsymbol{\Gamma}_{\boldsymbol{\kappa}+\boldsymbol{e}_k,:}\boldsymbol{A}_l\\ \Leftrightarrow & \left(\sum_{k=0}^{m} f_k\boldsymbol{\Gamma}_{\boldsymbol{\kappa}+\boldsymbol{e}_k,:}\right)^{\dagger}\boldsymbol{\Gamma}_{\boldsymbol{\kappa}+\boldsymbol{e}_l,:} = \boldsymbol{A}_l\left(\sum_{k=0}^{m} f_k\boldsymbol{A}_k\right)^{-1}\end{aligned}$$

Again, almost all linear combinations suffice to ensure $\boldsymbol{A}_{\boldsymbol{f}} = \sum_{k=0}^{m} f_k\boldsymbol{A}_k$ is invertible.

Example 2.4.4 below illustrates this technique for a null space in the generalized Vandermonde format. Note that using a similarity transform on $\boldsymbol{A}_k\boldsymbol{A}_{\boldsymbol{f}}^{-1}$, these procedures can be readily generalized to general bases for the trajectories, as already highlighted in the affine case [12].

Example 2.4.4. Reconsider $\boldsymbol{\Gamma}$ from Example (2.3.1). Choosing $f(\boldsymbol{\lambda}) = \boldsymbol{f}^{\mathsf{T}}\boldsymbol{\lambda}$ as a linear function of $\boldsymbol{\lambda}$, with $\boldsymbol{f} \in \mathbb{R}^{m+1}$ suffices to ensure that $f(\boldsymbol{\lambda})$ is nonzero in all the eigentuples $\boldsymbol{\lambda}_i$. This causes the associated shift matrix $\boldsymbol{A}_{\boldsymbol{f}} = f(\boldsymbol{A}_0, \boldsymbol{A}_1, \boldsymbol{A}_2)$ to be nonsingular as well. Writing out (2.9) in matrix format for $k = 1, \|\boldsymbol{\kappa}\|_1 = 2$

and $\boldsymbol{f} = \begin{bmatrix}3 & 2 & 1\end{bmatrix}^\mathsf{T}$ then results in:

$$\begin{matrix}3(3,0,0)+2(2,1,0)+(2,0,1)\\3(2,1,0)+2(1,2,0)+(1,1,1)\\3(1,0,1)+2(1,1,1)+(1,0,2)\\3(1,2,0)+2(0,3,0)+(0,2,1)\\3(1,1,1)+2(0,2,1)+(0,1,2)\\3(1,0,2)+2(0,1,2)+(0,0,3)\end{matrix}\begin{bmatrix}24 & 0 & 0\\0 & 0 & 0\\0 & 0 & 0\\0 & -16 & 0\\0 & 0 & 0\\0 & 0 & 8\end{bmatrix}\boldsymbol{A}_1 = \begin{matrix}(2,1,0)\\(1,2,0)\\(1,1,1)\\(0,3,0)\\(0,2,1)\\(0,1,2)\end{matrix}\begin{bmatrix}0 & 0 & 0\\0 & 0 & 0\\0 & 0 & 0\\0 & -8 & 0\\0 & 0 & 0\\0 & 0 & 0\end{bmatrix}\boldsymbol{A}_{\boldsymbol{f}}.$$

Notice that this linear combination causes the left coefficient matrix to be left invertible and $\boldsymbol{A}_{\boldsymbol{f}} = \operatorname{diag}(6,-4,2)$ invertible as well, such that one can solve this system for $\boldsymbol{A}_1\boldsymbol{A}_{\boldsymbol{f}}^{-1}$:

$$\boldsymbol{A}_1\boldsymbol{A}_{\boldsymbol{f}}^{-1} = \begin{bmatrix}0 & & \\ & \frac{1}{2} & \\ & & 0\end{bmatrix}.$$

A similar analysis applies to $\boldsymbol{A}_0\boldsymbol{A}_{\boldsymbol{f}}^{-1}$ and $\boldsymbol{A}_2\boldsymbol{A}_{\boldsymbol{f}}^{-1}$.

It can be seen that the eigentuples of the set of commuting matrices $\boldsymbol{A}_k\boldsymbol{A}_{\boldsymbol{f}}^{-1}, k = 0,1,2$ then correspond to the values of the rational function $\frac{\lambda_k}{f(\boldsymbol{\lambda})}$ in the eigentuples $(2,0,0),(0,-2,0)$ and $(0,0,2)$. Notice that these rational functions, where the numerator is of the same degree as the denominator, are well-defined on the projective eigentuples. That is, evaluating these functions in another representation $a\boldsymbol{\lambda}, a \in \mathbb{C}$, for the same projective point $\boldsymbol{\lambda}_i$ does not affect the resulting value, i.e., $\frac{a\lambda_k}{f(a\boldsymbol{\lambda})} = \frac{\lambda_k}{f(\boldsymbol{\lambda})}$. ◇

In Example 2.4.4 the eigentuples of $\boldsymbol{A}_k\boldsymbol{A}_{\boldsymbol{f}}^{-1}$ are clearly equal to $\frac{\boldsymbol{\lambda}}{f(\boldsymbol{\lambda})}$, an equivalent representation of the projective point $\boldsymbol{\lambda}$. This holds in general: Applying an appropriate joint Schur transformation to $(\boldsymbol{A}_0,\ldots,\boldsymbol{A}_m)$ brings them into a joint upper triangular form $(\boldsymbol{T}_0,\ldots,\boldsymbol{T}_m)$. Constructing $\boldsymbol{T}_f = \sum_{k=0}^m f_k\boldsymbol{T}_k$, one can subsequently see that $\boldsymbol{T}_k\boldsymbol{T}_f^{-1}$ is similar to $\boldsymbol{A}_k\boldsymbol{A}_{\boldsymbol{f}}^{-1}$, where the former matrix has the desired eigenvalues on its diagonal by construction.

In general, the matrices $\boldsymbol{A}_k\boldsymbol{A}_{\boldsymbol{f}}^{-1}$ can be used directly as the state-transition matrices in (2.6). Indeed, multiplying both sides of the equations by $\boldsymbol{A}_{\boldsymbol{f}}^{-1}$ does not alter the solutions $\boldsymbol{x}_{\boldsymbol{\kappa}}$. However, constructing trajectories as $\boldsymbol{C}(\boldsymbol{A}_0\boldsymbol{A}_{\boldsymbol{f}}^{-1},\ldots,\boldsymbol{A}_m\boldsymbol{A}_{\boldsymbol{f}}^{-1})^{\boldsymbol{\kappa}}\boldsymbol{x}$ leads to $\boldsymbol{C}\boldsymbol{A}^{\boldsymbol{\kappa}}\boldsymbol{A}_{\boldsymbol{f}}^{-d}\boldsymbol{x}$, so that the corresponding observability matrix differs from the original one by the invertible factor $\boldsymbol{A}_{\boldsymbol{f}}^{-d}$.

2.5 Data-based realization

2.5.1 Hankel matrix and its decomposition

Now that we have discussed all the state-space representations, we possess the necessary tools to discuss a special case of the image representation: the data-based representation. Indeed, for LTI systems, it is well known that, under some conditions, a single trajectory can be used to infer a basis for all possible trajectories using the shift-invariance property [125, 183]. By commutativity of the shift operators, this also applies to the model class under consideration: $\boldsymbol{r}(\boldsymbol{\sigma})\hat{y}_{\boldsymbol{\kappa}} = \mathbf{0} \rightarrow \boldsymbol{r}(\boldsymbol{\sigma})\hat{y}_{\boldsymbol{\kappa}+\boldsymbol{\gamma}} = \boldsymbol{\sigma}^{\boldsymbol{\gamma}}\boldsymbol{r}(\boldsymbol{\sigma})\hat{y}_{\boldsymbol{\kappa}} = \mathbf{0}$. This observation has been used in the mD case as well to define a generalized Hankel matrix and build a state-space model using multidimensional realization theory [6, 72]. Assuming a state-space representation of the form (2.6), and given a row and column degree d_r and d_c, respectively, the elements of the generalized Hankel matrix can be compactly defined using multi-indices:

$$\forall \|\boldsymbol{\rho}\|_1 = d_r, \|\boldsymbol{\gamma}\|_1 = d_c : \boldsymbol{H}_{\boldsymbol{\rho},\boldsymbol{\gamma}} = \hat{y}_{\boldsymbol{\rho}+\boldsymbol{\gamma}} = \boldsymbol{C}\boldsymbol{A}^{\boldsymbol{\rho}+\boldsymbol{\gamma}}\boldsymbol{x} = \boldsymbol{C}\boldsymbol{A}^{\boldsymbol{\rho}}\boldsymbol{x}_{\boldsymbol{\gamma}}.$$

This matrix can then be decomposed as

$$\boldsymbol{H} = \boldsymbol{\Gamma}\boldsymbol{X}, \tag{2.10}$$

where $\boldsymbol{\Gamma}$ is a degree d_r observability matrix and the columns of $\boldsymbol{X}$ are $\boldsymbol{x}_{\boldsymbol{\gamma}}$. As such, the rank of $\boldsymbol{H}$ is at most n. Moreover, we know that, when rank $\boldsymbol{H} = n$, it holds that range $\boldsymbol{H} =$ range $\boldsymbol{\Gamma}$, so that a basis for its column space can be used as a basis for all model-compliant trajectories of degree d_r. This basis can then be employed to find a kernel representation or a state-space model using the techniques previously described.

Conditions do apply for the rank of $\boldsymbol{H}$ to become n though: first and foremost, $\boldsymbol{X}$ should also be of full rank, which is not the case for all $\boldsymbol{x}$, much like some output matrices $\boldsymbol{C}$ are not suitable either, as mentioned in Example 2.4.2. This will be discussed in depth in the next subsection. In practice, the model order can be detected by building such Hankel-like matrices of increasing row degrees[15], until the rank stabilizes, similarly to building the null space of a Macaulay matrix [44, 188].

[15]Note that decreasing the column degree while increasing the row degree may hamper the growth in rank. However, as will be detailed in the next subsection, the lowest of both degrees is the 'bottleneck' for the rank, so keeping $d_c \geq d_r$ suffices.

Example 2.5.1. Consider a data sequence of degree $\|\boldsymbol{\kappa}\|_1 = 6$, e.g.:

$$\hat{y}_{(6,0,0)} = 1,\ \hat{y}_{(0,6,0)} = 2,\ \hat{y}_{(0,0,6)} = 3,$$

$$\hat{y}_{\boldsymbol{\kappa}} = 0,\ \text{for other } \boldsymbol{\kappa}, \text{with } \|\boldsymbol{\kappa}\|_1 = 6.$$

A generalized Hankel matrix can then be built with column and row indices up to degree 3:

$$\boldsymbol{H} = \begin{array}{c} \begin{matrix} (3,0,0) & (2,1,0) & \dots & (0,0,3) \end{matrix} \\ \begin{bmatrix} \hat{y}_{(6,0,0)} & \hat{y}_{(5,1,0)} & \cdots & \hat{y}_{(3,0,3)} \\ \hat{y}_{(5,1,0)} & \hat{y}_{(4,2,0)} & \cdots & \hat{y}_{(2,1,3)} \\ \vdots & \vdots & \vdots & \vdots \\ \hat{y}_{(3,0,3)} & \hat{y}_{(2,1,3)} & \cdots & \hat{y}_{(0,0,6)} \end{bmatrix} \end{array} \begin{matrix} (3,0,0) \\ (2,1,0) \\ \\ (0,0,3) \end{matrix}$$

$$= \begin{bmatrix} 1 & & & & & & & & & \\ & 0 & & & & & & & & \\ & & 0 & & & & & & & \\ & & & 0 & & & & & & \\ & & & & 0 & & & & & \\ & & & & & 0 & & & & \\ & & & & & & 2 & & & \\ & & & & & & & 0 & & \\ & & & & & & & & 0 & \\ & & & & & & & & & 3 \end{bmatrix}.$$

Clearly, this matrix is of rank 3, as was expected from the model order.

Next, computing a basis $\boldsymbol{\Gamma}$ for range $\boldsymbol{H}$ results in the observability matrix that was previously encountered for the running example:

$$\boldsymbol{\Gamma} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{matrix} (3,0,0) \\ (2,1,0) \\ (2,0,1) \\ (1,2,0) \\ (1,1,1) \\ (1,0,2) \\ (0,3,0) \\ (0,2,1) \\ (0,1,2) \\ (0,0,3) \end{matrix}.$$

The techniques of section 2.4.3 can then be applied to construct a state-space model, as done in Example 2.4.4. ◇

2.5.2 Order of the identified model

It was already mentioned that some values of $\boldsymbol{C}$ may cause the observability matrix to be rank-deficient. Indeed, while an nth order model characterized by difference equations always allows a representation of the form (2.6), the converse is not true: In a similar fashion to LTI systems, a state-space model of this form may describe a lower-order model due to some *unobservable modes*. The model can also be split into an observable and an unobservable part, only the former of which can be reconstructed through realization theory [6].

To provide more insight into the algebraic structure of such models as well as a succinct check for such modes, we now present a multidimensional generalization of the Popov-Belevitch-Hautus (PBH) test for observability [90]. There exist multiple notions of controllability and observability in multidimensional behavioral systems theory [139], which we do not explicitly explore. Instead, we focus on the rank of $\boldsymbol{\Gamma}$. A related highly algebraic account using the behavioral framework for systems theory can be found in [122]. Our presentation strongly differs from this, focusing on a linear-algebra-based approach, though it concerns less general models. A very similar analysis applies to $\boldsymbol{X}$ in (2.10), though with left instead of right eigenvectors.

To state the result, we first note that for any set of $m+1$ commuting matrices $\boldsymbol{A}_0, \dots, \boldsymbol{A}_m \in \mathbb{C}^{n\times n}$, there exists some finite degree d^* so that for all $\boldsymbol{\gamma} : \boldsymbol{A}^{\gamma} = \sum_{\|\boldsymbol{\kappa}\|_1 \leq d^*} l_{\boldsymbol{\kappa}} \boldsymbol{A}^{\boldsymbol{\kappa}}$, i.e., all monomials in these matrices are linear combinations of the monomials up to degree d^*. Indeed, by the Cayley-Hamilton theorem, all monomials in $\boldsymbol{A}_0, \dots, \boldsymbol{A}_m \in \mathbb{C}^{n\times n}$ can be written in terms of the finite set $\{\boldsymbol{A}^{\boldsymbol{\kappa}} \mid \|\boldsymbol{\kappa}\|_\infty \leq n-1\}$, a superset of all linearly independent $\boldsymbol{A}$-monomials. However, smaller sets of monomials exist. For example, [186] provides a smaller superset[16] which includes all degree monomials of $n-1$. To understand the PBH-like property, only the existence of this degree is of importance.

Theorem 2.5.1. *Let $\boldsymbol{\Gamma}$ be the degree $d_r \geq d^*$ observability matrix associated with a state-space model* (2.6)*. It then holds that* $\operatorname{rank} \boldsymbol{\Gamma} < n$ *if and only if there exists a joint eigenvector $\boldsymbol{x}$ of the generalized eigenvalue problems*

$$(\boldsymbol{A}_k - \beta_k \boldsymbol{A}_{\boldsymbol{f}})\boldsymbol{x} = \boldsymbol{0},$$

where $k = 0, \dots, m$ and $\boldsymbol{A}_{\boldsymbol{f}} = \sum_{k=0}^{m} f_k \boldsymbol{A}_k$ is invertible[17], for which $\boldsymbol{C}\boldsymbol{x} = 0$.

Proof. Let us start with the 'if' part. If there exists such a joint eigenvector $\boldsymbol{x}$, associated with some generalized eigentuple $\boldsymbol{\beta} = (\beta_0, \dots, \beta_m)$, we can rewrite

[16] In general, any normal set of a border basis for the ideal $I = \{p(\boldsymbol{\lambda}) \in \mathbb{C}[\boldsymbol{\lambda}] \mid p(\boldsymbol{A}) = \boldsymbol{0}\}$ provides a set of linearly independent matrix monomials. In practice, such a set can be found through what is essentially a 'block Buchberger-Möller' algorithm.

[17] As established in Section 2.4.3, such an $\boldsymbol{f}$ always exists.

the eigenvalue problems as $\boldsymbol{A}_k\boldsymbol{A}_{\boldsymbol{f}}^{-1}\boldsymbol{x} = \beta_k\boldsymbol{x}$. Then, by commutativity, it holds that $\forall\boldsymbol{\kappa}, \|\boldsymbol{\kappa}\|_1 = d_r : \boldsymbol{C}\boldsymbol{A}^{\boldsymbol{\kappa}}\boldsymbol{A}_{\boldsymbol{f}}^{-d_r}\boldsymbol{x} = \boldsymbol{C}(\boldsymbol{A}_0\boldsymbol{A}_{\boldsymbol{f}}^{-1}, \ldots, \boldsymbol{A}_m\boldsymbol{A}_{\boldsymbol{f}}^{-1})^{\boldsymbol{\kappa}}\boldsymbol{x} = \boldsymbol{C}\boldsymbol{x}\boldsymbol{\beta}^{\boldsymbol{\kappa}} = \boldsymbol{0}$ Since $\boldsymbol{\Gamma}_{\boldsymbol{\kappa},:} = \boldsymbol{C}\boldsymbol{A}^{\boldsymbol{\kappa}}$ and $\boldsymbol{A}_{\boldsymbol{f}}$ is invertible, $\boldsymbol{\Gamma}$ has a nontrivial right null space, i.e., $\boldsymbol{A}_{\boldsymbol{f}}^{-d_r}\boldsymbol{x}$.

Now, let us consider the 'only if' part. As discussed in Sections 2.4.2, from any state-space model, another model with the state-transition matrices $\boldsymbol{A}_{\xi_k} = \sum_{l=0}^{m}\boldsymbol{P}_{k,l}\boldsymbol{A}_l$ can be built, so that this new model has no eigentuples at infinity, i.e., $\boldsymbol{A}_{\xi_0}$ is invertible. As in Section 2.3.2, then any monomial $\boldsymbol{A}^{\boldsymbol{\kappa}}$ can be written as a linear combination of monomials $\boldsymbol{A}_{\boldsymbol{\xi}}^{\boldsymbol{\kappa}}$ of the same degree, which implies that there is a linear relation between the observability matrices of both models, that is, $\boldsymbol{\Xi} = \boldsymbol{R}\boldsymbol{\Gamma}$, where $\boldsymbol{R}$ is invertible. Consequently, both these observability matrices have the same right null space. Now consider the change of basis $\boldsymbol{\Xi}^{(2)} = \boldsymbol{\Xi}\boldsymbol{A}_{\xi_0}^{-d_r}$. Let $\boldsymbol{x}$ lie in $\operatorname{null}\boldsymbol{\Gamma} = \operatorname{null}\boldsymbol{\Xi}$, then $\boldsymbol{\Xi}^{(2)}\boldsymbol{A}_{\xi_0}^{d_r}\boldsymbol{x} = \boldsymbol{0}$.

By commutativity, the rows of $\boldsymbol{\Xi}^{(2)}$ consist of all vectors of the form $\boldsymbol{C}\boldsymbol{A}_{\boldsymbol{\xi}}^{\boldsymbol{\kappa}}\boldsymbol{A}_{\xi_0}^{-d_r} = \boldsymbol{C}(\boldsymbol{A}_{\xi_1}\boldsymbol{A}_{\xi_0}^{-1}, \ldots, \boldsymbol{A}_{\xi_m}\boldsymbol{A}_{\xi_0}^{-1})^{\boldsymbol{\kappa}}$, with $\|\boldsymbol{\kappa}\|_1 \leq d_r, \boldsymbol{\kappa} \in \mathbb{N}_0^m$. Since all monomials can be written in terms of the degree $\leq d^*$ monomials[18], any vector of the form $\boldsymbol{C}(\boldsymbol{A}_{\xi_1}\boldsymbol{A}_{\xi_0}^{-1}, \ldots, \boldsymbol{A}_{\xi_m}\boldsymbol{A}_{\xi_0}^{-1})^{\boldsymbol{\kappa}}$ can be written as a linear combination of the rows of $\boldsymbol{\Xi}^{(2)}$. Consequently, for any $\boldsymbol{x} \in \operatorname{null}\boldsymbol{\Xi}$, and any $\boldsymbol{\kappa}$ and k, there exists a vector $\boldsymbol{l}^{\mathsf{T}}$ such that $\boldsymbol{C}\boldsymbol{A}_{\boldsymbol{\xi}}^{\boldsymbol{\kappa}}(\boldsymbol{A}_{\xi_k}\boldsymbol{A}_{\xi_0}^{-1}\boldsymbol{x}) = \boldsymbol{C}(\boldsymbol{A}_{\xi_1}\boldsymbol{A}_{\xi_0}^{-1}, \ldots, \boldsymbol{A}_{\xi_m}\boldsymbol{A}_{\xi_0}^{-1})^{\boldsymbol{\kappa}+\boldsymbol{e}_k}\boldsymbol{A}_{\xi_0}^{d_r}\boldsymbol{x} = \boldsymbol{l}^{\mathsf{T}}\boldsymbol{\Xi}^{(2)}\boldsymbol{A}_{\xi_0}^{d_r}\boldsymbol{x} = 0$. As such, $(\boldsymbol{A}_{\xi_k}\boldsymbol{A}_{\xi_0}^{-1}\boldsymbol{x}) \in \operatorname{null}\boldsymbol{\Xi}$. From this, we conclude that $\forall k : (\boldsymbol{A}_{\xi_k}\boldsymbol{A}_{\xi_0}^{-1})(\operatorname{null}\boldsymbol{\Xi}) \subseteq \operatorname{null}\boldsymbol{\Xi}$, that is, $\operatorname{null}\boldsymbol{\Xi} = \operatorname{null}\boldsymbol{\Gamma}$ is an invariant subspace of all $\boldsymbol{A}_{\xi_k}\boldsymbol{A}_{\xi_0}^{-1}$, and by linearity, of any $\boldsymbol{A}_k\boldsymbol{A}_{\xi_0}^{-1}$ as well.

Now, every joint invariant subspace of a set of commuting matrices contains a joint eigenvector [82]. For didactical purposes, let us briefly elaborate on this statement: Let $\mathcal{X}$ be a $\boldsymbol{P}$- and $\boldsymbol{T}$-invariant subspaces of $\mathbb{C}^n$, where $\boldsymbol{P}$ and $\boldsymbol{T}$ commute. As discussed in Appendix A.2, each $\boldsymbol{P}$-invariant subspace contains an eigenvector of $\boldsymbol{P}$ belonging to some eigenspace $\mathcal{V}_\lambda$. Due to commutativity, it holds that $\boldsymbol{P}\boldsymbol{T}\boldsymbol{v} = \boldsymbol{T}\boldsymbol{P}\boldsymbol{v} = \lambda\boldsymbol{T}\boldsymbol{v}$ for any such eigenvector $\boldsymbol{v}$, so that $\boldsymbol{T}\mathcal{V}_\lambda \subseteq \mathcal{V}_\lambda$. Since $\boldsymbol{T}\mathcal{X} \subseteq \mathcal{X}$ as well, it follows that $\boldsymbol{T}(\mathcal{V}_\lambda \cap \mathcal{X}) \subseteq \mathcal{V}_\lambda \cap \mathcal{X} \neq \{0\}$. That is, the intersection is $\boldsymbol{T}$-invariant as well, and thus, there is a $\boldsymbol{T}$-eigenvector in $\mathcal{V}_\lambda \cap \mathcal{X}$.

Consequently, there exists some joint eigenvector $\boldsymbol{x} \in \operatorname{null}\boldsymbol{\Gamma}$ of the matrices $\boldsymbol{A}_k\boldsymbol{A}_{\xi_0}^{-1}$, associated with an eigentuple $\boldsymbol{\beta}$. By the definition of $\boldsymbol{\Gamma}$, it then holds that $\forall\boldsymbol{\kappa} \in \mathbb{N}_0^{m+1}, \|\boldsymbol{\kappa}\|_1 = d_r : \boldsymbol{0} = \boldsymbol{C}\boldsymbol{A}^{\boldsymbol{\kappa}}\boldsymbol{x} = \boldsymbol{C}(\boldsymbol{A}_0\boldsymbol{A}_{\xi_0}^{-1}, \ldots, \boldsymbol{A}_m\boldsymbol{A}_{\xi_0}^{-1})^{\boldsymbol{\kappa}}(\boldsymbol{A}_{\xi_0}^{d_r}\boldsymbol{x}) = \boldsymbol{C}\boldsymbol{\beta}^{\boldsymbol{\kappa}}(\boldsymbol{A}_{\xi_0}^{d_r}\boldsymbol{x})$, as $\boldsymbol{A}_{\xi_0}^{d_r}\boldsymbol{x}$ is a joint eigenvector

[18] It can be seen that d^* is the same for monomials in $\boldsymbol{A}_{\boldsymbol{\xi}}$ and $\boldsymbol{A}$: Using the coordinate transform, write the $\boldsymbol{A}_{\boldsymbol{\xi}}$-monomial in terms of $\boldsymbol{A}$-monomials of the same degree. Next, write those in terms of the lower degree $\boldsymbol{A}$-monomials. Finally express the result in terms of lower degree $\boldsymbol{A}_{\boldsymbol{\xi}}$-monomials again.

of the $\boldsymbol{A}_k \boldsymbol{A}_{\xi_0}^{-1}$ as well. Since $\boldsymbol{A}_{\xi_0}$ is invertible, the eigentuples are nonzero, so that $\boldsymbol{\beta}^{\boldsymbol{\kappa}}$ is also nonzero for some $\boldsymbol{\kappa}$. Consequently, $\boldsymbol{C}\boldsymbol{A}_{\xi_0}^{d_r}\boldsymbol{x} = 0$. Finally, this results in the eigenvalue problems $(\boldsymbol{A}_k - \beta_k \boldsymbol{A}_{\boldsymbol{f}})(\boldsymbol{A}_{\xi_0}^{d_r}\boldsymbol{x}) = \boldsymbol{0}$, where choosing $\boldsymbol{A}_{\boldsymbol{f}} = \boldsymbol{A}_{\xi_0}$ yields the desired result. □

2.6 Applications

The preceding sections showed that autonomous, scalar, linear, shift-invariant models can output finite sums of multivariate exponential signals. Such signals are common solution components of linear, homogeneous partial differential equations (PDEs) with constant coefficients. Of course, there exist many PDEs that are incompatible with such separable solution components as well. Nonetheless, the considered model class may still serve as approximation tools for PDEs [17, 76, 79, 202]. The formal study of PDEs, the approximation thereof, and partial difference equations is a vast field of research in and of itself, outside the scope of this dissertation. For an overview, we refer the reader to [83, 131, 170]. Let us illustrate how the considered model class can model a finite set of solution components of a sampled solution to the wave equation.

Example 2.6.1. Consider the 1D wave equation with Dirichlet boundary conditions over the x-domain:

$$\frac{\partial y(x,t)}{\partial t^2} = c^2 \frac{\partial y(x,t)}{\partial x^2} \text{ for } 0 < x < L,$$

$$y(0,t) = 0 = y(L,t).$$

For an appropriate set of initial conditions $y(x,0) = \phi(x)$, $\frac{\partial y(x,0)}{\partial t} = \psi(x)$, the solution can be found as [170]:

$$y(x,t) = \sum_{i=1}^{\infty} \left(a_i \cos\left(\frac{i\pi ct}{L}\right) + b_i \sin\left(\frac{i\pi ct}{L}\right)\right) \sin\left(\frac{i\pi x}{L}\right),$$

where a_i and b_i are the coefficients of the Fourier series expansion of $y(x,0)$.

Now, we sample this solution at discrete grid points $(\kappa_1 \Delta x, \kappa_2 \Delta t)$ in time and space and rewrite it as follows:

$$y_{\boldsymbol{\kappa}} = \frac{1}{2} \sum_{i=1}^{\infty} a_i \sin\left(\frac{i\pi(\kappa_1 \Delta x + c\kappa_2 \Delta t)}{L}\right) + a_i \sin\left(\frac{i\pi(\kappa_1 \Delta x - c\kappa_2 \Delta t)}{L}\right)$$

$$-b_i \cos\left(\frac{i\pi(\kappa_1 \Delta x + c\kappa_2 \Delta t)}{L}\right) + b_i \cos\left(\frac{i\pi(\kappa_1 \Delta x - c\kappa_2 \Delta t)}{L}\right).$$

As such, choosing r_i and θ_i such that $b_i + a_i = 2r_i \cos(\theta_i)$ and $b_i - a_i = 2r_i \sin(\theta_i)$, we conclude that the contribution of the ith *solution mode* corresponds to the output of the following fourth-order system:

$$\boldsymbol{A}_1 = \begin{bmatrix} \cos\left(\frac{i\pi\Delta x}{L}\right) & \sin\left(\frac{i\pi\Delta x}{L}\right) & & \\ -\sin\left(\frac{i\pi\Delta x}{L}\right) & \cos\left(\frac{i\pi\Delta x}{L}\right) & & \\ & & \cos\left(\frac{i\pi\Delta x}{L}\right) & \sin\left(\frac{i\pi\Delta x}{L}\right) \\ & & -\sin\left(\frac{i\pi\Delta x}{L}\right) & \cos\left(\frac{i\pi\Delta x}{L}\right) \end{bmatrix},$$

$$\boldsymbol{A}_2 = \begin{bmatrix} \cos\left(\frac{i\pi c\Delta t}{L}\right) & \sin\left(\frac{i\pi c\Delta t}{L}\right) & & \\ -\sin\left(\frac{i\pi c\Delta t}{L}\right) & \cos\left(\frac{i\pi c\Delta t}{L}\right) & & \\ & & \cos\left(\frac{i\pi c\Delta t}{L}\right) & -\sin\left(\frac{i\pi c\Delta t}{L}\right) \\ & & \sin\left(\frac{i\pi c\Delta t}{L}\right) & \cos\left(\frac{i\pi c\Delta t}{L}\right) \end{bmatrix},$$

$$\boldsymbol{C} = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}, \boldsymbol{x}_{\boldsymbol{\kappa}} = \begin{bmatrix} r_i \sin\left(\frac{i\pi(\kappa_1\Delta x + c\kappa_2\Delta t)}{L} + \theta_i\right) \\ r_i \cos\left(\frac{i\pi(\kappa_1\Delta x + c\kappa_2\Delta t)}{L} + \theta_i\right) \\ r_i \sin\left(\frac{i\pi(\kappa_1\Delta x - c\kappa_2\Delta t)}{L} + \theta_i\right) \\ r_i \cos\left(\frac{i\pi(\kappa_1\Delta x - c\kappa_2\Delta t)}{L} + \theta_i\right) \end{bmatrix}.$$

By extension, any solution to this wave equation consisting of $s \in \mathbb{N}$ nonzero *modes* can be described by a $4s$th order state-space model of the form (2.3), by joining s models of the form above. As the general solutions can have an infinite number of modes, these models are almost always an approximation. Then again, good approximations are often the best one can hope for when dealing with PDEs. Note that the sampling can cause aliasing to arise, resulting in some *modes* becoming indistinguishable from one another, so that $y_{\boldsymbol{\kappa}}$ becomes a sum of a finite number of modes. ◇

Since an of array physical phenomena is governed by such linear PDEs, these complex exponentials are a useful tool for analyzing multidimensional signals. Most importantly, the field of multidimensional harmonic retrieval often aims to decompose signals into such components [45, 86, 164, 168, 184, 185]. Many signal processing applications rely on these techniques, for example: analysis of wireless communications channels [69, 128], nuclear magnetic resonance spectroscopy [67, 119, 201], blind source separation [174], and direction of arrival [68], among others.

2.7 Conclusion

This preliminary chapter briefly introduced class of finite-order, autonomous, scalar, linear, shift-invariant models under consideration in this dissertation. Said model class generalizes autonomous LTI models to a multidimensional setting. More precisely, the intricate link with multivariate polynomials with a zero-dimensional set of common roots, and, by extension, the Macaulay framework, was explored. This resulted in a system theoretic interpretation of said framework, which provides a kernel, image, and state-space representation for the model-compliant trajectories. Moreover, the different linear-algebra-based transformations between these representations were also briefly highlighted. Furthermore, descriptor models were discussed as well, that is, models with eigentuples at infinity. The aforementioned representations and algorithms naturally extend to this more general case by associating such descriptor models with affine models using a linear coordinate transform. Here, it is shown that a rational format of state-space equations is required to describe the model-compliant trajectories. A data-based variation on the image representation using a generalized Hankel matrix was highlighted as well, along with a linear-algebra-based account of a generalization of the Popov-Belevitch-Hautus test to this model class. Finally, some potential applications for this class of models are briefly mentioned.

Chapter 3

Constructing Multidimensional Difference Equations from a State-Space Representation using the Generalized Cayley-Hamilton Theorem

Chapter abstract

We show that applying a generalization of the Cayley-Hamilton Theorem to a state-space representation of a multidimensional (mD), linear, shift-invariant, causal, autonomous system, with distinct eigentuples results in an equivalent difference equation representation. The proposed method is also applicable to parameterize a set of mD shift-invariant difference equations in terms of a given set of eigenvalues. Lastly, a closed-form expression in terms of the eigenvalues of the state-transition matrices is derived.

Remark: Up to minor notational and terminology changes for overall consistency, this chapter is identical to the following, previously published work:

- Vanpoucke, L., and De Moor, B. (2024). "Constructing multidimensional difference equations from a state-space representation using the generalized Cayley-Hamilton theorem". IEEE Control Systems Letters 8, pp. 2259-2264.

Big picture

The current chapter provides a novel way of transitioning directly from a state-space representation (or multiplication maps in a polynomial sense) to an equivalent set of difference equations, i.e., a kernel representation, which were discussed in Chapter 2. This can be used as a practical and theoretical tool to manipulate such models. Moreover, when applied to a parametric state-space model, e.g., a model with variable eigentuples, the results of this chapter lead directly to a parameterization for the difference equations in terms of the same variables. This strategy is applied in Chapter 4, to tackle the misfit problem in the kernel representation. While the case of descriptor models is not considered explicitly in this chapter, the insights of Chapter 2 can be used to generalize the construction by associating the given descriptor model with an affine one through a linear coordinate transform, applying the construction, and subsequently reverting the coordinate transform.
Remark that, since only affine models are considered, we step away from the projective setting. As such, all tuples are considered as points in $\mathbb{C}^m$ instead of $\mathbb{P}^m$, and no more 0th index is considered from hereon.

3.1 Introduction

Discrete-time, linear, time-invariant (LTI) models provide a great modeling tool for many engineering applications. Nowadays, multidimensional data, where many other variables besides time are considered, is abundantly available. To this end, many of the tools provided by LTI systems have been generalized to the multidimensional (mD) setting. In this work, we focus on the class of autonomous, scalar, linear, shift-invariant models, as described by the state-space representation in Equation (3.1), where $\boldsymbol{e}_k \in \mathbb{Z}^m$ denotes the kth column of the identity matrix, the multi-index of an m-dimensional grid point is denoted by[1] $\boldsymbol{\kappa} \in \mathbb{N}_0^m$, $\boldsymbol{A}_k \in \mathbb{R}^{n\times n}$, $k = 1, \ldots, m$ and $\boldsymbol{C} \in \mathbb{R}^{1\times n}$:

$$\begin{aligned} &\boldsymbol{x}_{\boldsymbol{\kappa}+\boldsymbol{e}_1} = \boldsymbol{A}_1\boldsymbol{x}_{\boldsymbol{\kappa}};\ \ldots;\ \boldsymbol{x}_{\boldsymbol{\kappa}+\boldsymbol{e}_m} = \boldsymbol{A}_m\boldsymbol{x}_{\boldsymbol{\kappa}};\ \hat{y}_{\boldsymbol{\kappa}} = \boldsymbol{C}\boldsymbol{x}_{\boldsymbol{\kappa}} \\ &\text{where } \forall i, j \in \{1, \ldots, m\} : \boldsymbol{A}_i\boldsymbol{A}_j = \boldsymbol{A}_j\boldsymbol{A}_i. \end{aligned} \tag{3.1}$$

This model class, known by various different names, has been studied extensively in the literature, [6, 8, 44, 58]. Through Equation (3.1), any initial state vector $\boldsymbol{x}_{\boldsymbol{0}} \in \mathbb{R}^n$, where n is the model order, defines an output signal $\hat{y}_{\boldsymbol{\kappa}} : \mathbb{N}_0^m \to$

[1] $\mathbb{N}_0$ denotes the set of natural numbers, including zero and $\mathbb{N}$ is assumed to not include zero.

$\mathbb{R}; \boldsymbol{\kappa} \mapsto \hat{y}_{\boldsymbol{\kappa}}$ over the whole positive orthant $\mathbb{N}_0^m$. Note that the commutativity of the state-transition matrices $\boldsymbol{A}_k \in \mathbb{R}^{n\times n}$ is required for the well-definedness of the state and output signals. For example, in 2D, when stepping from $\boldsymbol{x}_{(0,0)}$ to $\boldsymbol{x}_{(1,1)}$, the outcome should be path-independent, such that $\boldsymbol{A}_2\boldsymbol{A}_1\boldsymbol{x}_{(0,0)} = \boldsymbol{A}_2\boldsymbol{x}_{(1,0)} = \boldsymbol{x}_{(1,1)} = \boldsymbol{A}_1\boldsymbol{x}_{(0,1)} = \boldsymbol{A}_1\boldsymbol{A}_2\boldsymbol{x}_{(0,0)}$. Consequently, by grouping the state-transition matrices in the tuple $\boldsymbol{A} = (\boldsymbol{A}_1, \ldots, \boldsymbol{A}_m)$ and introducing a shorthand notation[2], the model output can be compactly written as: $\hat{y}_{\boldsymbol{\kappa}} = \boldsymbol{C}\boldsymbol{A}^{\boldsymbol{\kappa}}\boldsymbol{x_0} = \boldsymbol{C}\prod_{k=1}^{m}\boldsymbol{A}_k^{\kappa_k}\boldsymbol{x_0}$. This shows that the model dynamics are invariant under a common similarity transform $\boldsymbol{T}$ on the state-transition matrices. Indeed, defining $\boldsymbol{A}_{new} = \boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{T}$, $\boldsymbol{x}_{\boldsymbol{0}}^{new} = \boldsymbol{T}^{-1}\boldsymbol{x_0}$ and $\boldsymbol{C}_{new} = \boldsymbol{C}\boldsymbol{T}$, the output remains: $\hat{y}_{\boldsymbol{\kappa}}^{\text{new}} = \boldsymbol{C}_{\text{new}}\boldsymbol{A}_{\text{new}}^{\boldsymbol{\kappa}}\boldsymbol{x}_{\boldsymbol{0}}^{\text{new}} = \boldsymbol{C}\boldsymbol{A}^{\boldsymbol{\kappa}}\boldsymbol{x_0}$.

In the current work, we consider the problem of constructing an equivalent difference equation representation for a given, possibly parametric, autonomous state-space model (3.1), with distinct eigentuples, that is, eigenvalues of the state-transition matrices grouped into m-tuples[3]. Indeed, akin to the 1D case of LTI systems, it has been shown in [58] that the considered class of mD models can be equivalently described using difference equations with constant coefficients, i.e., shift-invariant, also called partial difference equations. For a minimal, single-output, autonomous, LTI state-space model, the Cayley-Hamilton Theorem implies that the characteristic polynomial of the state-transition matrix $\boldsymbol{A}$, readily expressed in terms of the eigenvalues, forms an equivalent difference equation representation for the model [13]. This contribution aims to generalize this procedure to the mD setting.

A similar problem has been tackled in [13] using the Buchberger-Möller algorithm [129], briefly illustrated in Example 3.1.1. From a system theoretic point of view, the algorithm in [13] constructs a multidimensional *observability matrix* $\boldsymbol{\Gamma}$[4] [44], whose rows are $\boldsymbol{\Gamma}_{l,:} = \boldsymbol{C}\boldsymbol{A}^{\boldsymbol{\kappa}_l}$, for a given set of N grid points $\{\boldsymbol{\kappa}_l |\ l = 1, ..., N\}$ and an ordering on them. The columns of an observability matrix then form a basis for the model-compliant signals $\hat{y}_{\boldsymbol{\kappa}}$ restricted to the given grid points.

The algorithm then sequentially searches for a set of linearly independent grid points $\boldsymbol{\kappa}$ to serve as *initial states*, from which linear recurrence relations, a so-called *Gröbner basis*, can be derived that define the signal values in subsequent grid points. A drawback of this approach is that the resulting set of initial states may vary depending on the precise eigenvalues of the provided system matrices $\boldsymbol{A}_1, \ldots, \boldsymbol{A}_m$ [169]. In cases where the precise values of these eigenvalues are unknown, e.g., when optimizing over them as model parameters to find those

[2] the shorthand notation $\boldsymbol{x}^{\boldsymbol{\alpha}} = \prod_i x_i^{\alpha_i}$ will be used for general multivariate monomials described by the multi-exponent (multi-index) $\boldsymbol{\alpha}$.

[3] This grouping is defined precisely in Section 3.3, after Lemma 3.3.1

[4] The index ':' indicates all rows or columns are selected.

that optimally fit the given data [36], this may lead to invalid parameterizations of the considered models, as illustrated in the 2D Example 3.1.1.

Example 3.1.1. Consider the following state-space model with eigentuples, (a_1, b_1) and $(a_2, b_2) \in \mathbb{C}^2$ and system matrices:

$$\boldsymbol{A}_1 = \begin{bmatrix} a_1 & \\ & a_2 \end{bmatrix}, \boldsymbol{A}_2 = \begin{bmatrix} b_1 & \\ & b_2 \end{bmatrix}, \boldsymbol{C} = \begin{bmatrix} 1 & 1 \end{bmatrix}.$$

To find a set of difference equations that the outputs $\hat{y}_{\boldsymbol{\kappa}}$ of this model must satisfy, we first construct a multidimensional observability matrix $\boldsymbol{\Gamma}$. To this end, we define an appropriate ordering on the grid points, for instance, consider the ordering used in [44], which, in (2D), is : $(0,0) \prec (1,0) \prec (0,1) \prec (2,0) \prec (1,1), \ldots$. Then, we respectively arrange the output signals $\hat{y}_{\boldsymbol{\kappa}} = \boldsymbol{C}\boldsymbol{A}^{\boldsymbol{\kappa}}\boldsymbol{x_0}$ corresponding to the initial states $\boldsymbol{x_0}^\mathsf{T} = \begin{bmatrix} 1 & 0 \end{bmatrix}$ and $\boldsymbol{x_0}^\mathsf{T} = \begin{bmatrix} 0 & 1 \end{bmatrix}$ in the rows of $\boldsymbol{\Gamma}^\mathsf{T}$ below. Note that, due to the linearity of the model, the rows of $\boldsymbol{\Gamma}^\mathsf{T}$ form a basis for the model-compliant output signals, since any initial state can be written as a linear combination of the two considered states. Considering the points $\boldsymbol{\kappa} \in \mathbb{N}_0^2$, where $\|\boldsymbol{\kappa}\|_1 \leq 2$, this results in the following matrix:

$$\boldsymbol{\Gamma}^\mathsf{T} = \begin{array}{c} \begin{matrix} \hat{\boldsymbol{y}}_{(0,0)} & \hat{\boldsymbol{y}}_{(1,0)} & \hat{\boldsymbol{y}}_{(0,1)} & \hat{\boldsymbol{y}}_{(2,0)} & \hat{\boldsymbol{y}}_{(1,1)} & \hat{\boldsymbol{y}}_{(0,2)} \end{matrix} \\ \begin{bmatrix} 1 & a_1 & b_1 & a_1^2 & a_1 b_1 & b_1^2 \\ 1 & a_2 & b_2 & a_2^2 & a_2 b_2 & b_2^2 \end{bmatrix} \end{array}.$$

When $a_1 \neq a_2$, the first and second column of $\boldsymbol{\Gamma}^\mathsf{T}$ are linearly independent. Clearly, all subsequent columns can then be written as linear combinations of these independent columns, e.g., $(a_1 - a_2)\hat{\boldsymbol{y}}_{(0,1)} = (a_1 b_2 - a_2 b_1)\hat{\boldsymbol{y}}_{(0,0)} + (b_1 - b_2)\hat{\boldsymbol{y}}_{(1,0)}$ and $\hat{\boldsymbol{y}}_{(2,0)} = (a_1 + a_2)\hat{\boldsymbol{y}}_{(1,0)} - a_1 a_2 \hat{\boldsymbol{y}}_{(0,0)}$, where $\hat{\boldsymbol{y}}_{\boldsymbol{\kappa}} \in \mathbb{R}^{2\times 1}$. Note that due to the multi-shift invariance of the model outputs, these relations can be 'shifted' throughout the grid $\mathbb{N}_0^m$ by left multiplying by ${\boldsymbol{A}_k}^\mathsf{T}$, leading to linear relations that define $\hat{y}_{\boldsymbol{\kappa}}$ for $\boldsymbol{\kappa} \in \mathbb{N}_0^2$ with $\hat{\boldsymbol{y}}_{(0,0)}, \hat{\boldsymbol{y}}_{(1,0)}$ as initial values. However, in case $a_1 = a_2$, the relation $\hat{\boldsymbol{y}}_{(1,0)} = a_1 \hat{\boldsymbol{y}}_{(0,0)}$ holds, so that $\hat{\boldsymbol{y}}_{(1,0)}$ is linearly dependent on $\hat{\boldsymbol{y}}_{(0,0)}$ and thus cannot be considered an initial value. Instead, $\hat{\boldsymbol{y}}_{(0,0)}$ and $\hat{\boldsymbol{y}}_{(0,1)}$ then act as initial values if $b_1 \neq b_2$.

When considering the eigentuples as variables, it is not known which case applies, though. As it is the generic case, one might use the recurrence equations provided by the case $a_1 \neq a_2$ as a parameterization for the model in terms of these variables. Then, substituting $a_1 = a_2$ into these recurrence relations yields $(b_1 - b_2)\hat{y}_{(1,0)} = a_1(b_1 - b_2)\hat{y}_{(0,0)}$ and $\hat{y}_{(2,0)} = 2a_1 \hat{y}_{(1,0)} - a_1^2 \hat{y}_{(0,0)}$, which acts as a set of 1D difference equations. Consequently, all signals of the form $\hat{y}_{\boldsymbol{\kappa}} = a_1^{\kappa_1} \hat{y}_{\kappa_2}$, where $\hat{y}_{\kappa_2}$ can be any 1D signal, satisfy this set of difference equations, as will be explained in Example 3.1.3. As such, these equations then describe an infinite-order system, instead of the intended second-order system. In essence, the parameterization 'breaks down' in this nongeneric case, describing an infinite-order model, which does not allow a finite-dimensional state-space representation (3.1), instead of the intended second-order model [58, 169]. ⋄

Aside from the theoretical value in offering an alternative transition from a state-space to a difference equation representation, the approach presented in this work provides the advantage of a more general parameterization of models w.r.t. the Gröbner-based methods. The presented construction relies on combining the properties of commuting matrices, with the Generalized Cayley-Hamilton Theorem, introduced in Section 3.2. To this end, we first explain the link between multivariate polynomials and mD difference equations.

Formally, difference equations can be generalized to the multidimensional setting by introducing the linear shift operators: $\sigma_i y_{\boldsymbol{\kappa}} = y_{\boldsymbol{\kappa}+\boldsymbol{e}_i}$, that a set of difference equations can be characterized by a multivariate polynomial matrix in these operators $\boldsymbol{\sigma} = \left[\sigma_1, \ldots, \sigma_m\right]$, as illustrated by the following example of 2D homogeneous difference equations.

Example 3.1.2. Consider the following set of 2D difference equations with constant coefficients in the scalar signal $\hat{y}_{\boldsymbol{\kappa}}$:

$$\begin{bmatrix} \hat{y}_{\boldsymbol{\kappa}} - \hat{y}_{\boldsymbol{\kappa}+(1,1)} + \hat{y}_{\boldsymbol{\kappa}+(2,0)} \\ 6\hat{y}_{\boldsymbol{\kappa}} - 5\hat{y}_{\boldsymbol{\kappa}+(0,1)} + \hat{y}_{\boldsymbol{\kappa}+(0,2)} \end{bmatrix} = \begin{bmatrix} 1 - \sigma_1\sigma_2 + \sigma_1^2 \\ 6 - 5\,\sigma_2 + \sigma_2^2 \end{bmatrix} \hat{y}_{\boldsymbol{\kappa}}$$

$$= \boldsymbol{r}(\boldsymbol{\sigma})\hat{y}_{\boldsymbol{\kappa}} = \boldsymbol{0},\ \forall \boldsymbol{\kappa} \in \mathbb{N}_0^2. \qquad \diamond$$

In the *behavioral framework* for systems theory, model-compliant signals are defined as signals that satisfy such a set of difference equations at each grid point $\boldsymbol{\kappa} \in \mathbb{N}_0^m$, i.e., any $\hat{y}_{\boldsymbol{\kappa}}$ that lies in the kernel of the difference operator: $\boldsymbol{r}(\boldsymbol{\sigma})\hat{y}_{\boldsymbol{\kappa}} = \boldsymbol{0}, \forall \boldsymbol{\kappa} \in \mathbb{N}_0^m$ [139, 159, 197, 205]. Hence, this type of representation is a kernel representation. Due to the linearity of the shift operators $\boldsymbol{\sigma}$, this kernel forms a vector space over $\mathbb{C}$, which can be thought of as the column space of an infinitely long mD observability matrix $\boldsymbol{\Gamma} \in \mathbb{R}^{\infty \times n}$, i.e., constructed using the whole grid $\mathbb{N}_0^m$. Since the model class considered in this paper admits a finite-dimensional state-space system description as described in Equation (3.1), the kernel of the operator $\boldsymbol{r}(\boldsymbol{\sigma})$ that characterizes the system dynamics is also finite-dimensional [58].

As previously stated, we restrict ourselves to scalar, autonomous systems, such that this leads to homogeneous scalar difference equations. For the considered class of difference equations, a basis for the kernel of $\boldsymbol{r}(\boldsymbol{\sigma})$ is readily obtained by filling in an exponential of the form $\hat{y}_{\boldsymbol{\kappa}} = \boldsymbol{\lambda}^{\boldsymbol{\kappa}}$ and subsequently solving $\boldsymbol{r}(\boldsymbol{\sigma})$ as a set of multivariate polynomials, as illustrated in Example 3.1.3 below [83].

Example 3.1.3. Substituting a signal of the form $\hat{y}_{\boldsymbol{\kappa}} = \lambda_1^{\kappa_1}\lambda_2^{\kappa_2}$ into the difference equations from Example 3.1.2 we obtain:

$$\begin{cases} \lambda_1^{\kappa_1}\lambda_2^{\kappa_2}\left(1 - \lambda_1\lambda_2 + \lambda_1^2\right) = 0 \\ \lambda_1^{\kappa_1}\lambda_2^{\kappa_2}\left(6 - 5\lambda_2 + \lambda_2^2\right) = 0 \end{cases} \quad \forall \boldsymbol{\kappa} \in \mathbb{N}_0^2,$$

which corresponds to the set of polynomial equations $\boldsymbol{r}(\lambda_1, \lambda_2)$ after dividing away the factor $\lambda_1^{\kappa_1}\lambda_2^{\kappa_2}$. ◇

Example 3.1.3 illustrates that when the polynomials $\boldsymbol{r}(\boldsymbol{\sigma})$ have a finite number of common roots $\boldsymbol{\lambda}_i, i = 1, \ldots, n$, it is n-dimensional with $\boldsymbol{\lambda}_i^{\boldsymbol{\kappa}}$ as its basis signals. These roots then correspond to the eigentuples of an equivalent state-space model [44, 58]. As such, the considered problem, that is, constructing an equivalent difference equation representation $\boldsymbol{r}(\boldsymbol{\sigma})$ for a given state-space model (3.1), can be thought of as constructing a set of multivariate polynomials that precisely describes a given a set of n mD points as its set of common roots. In general, these basis functions are not necessarily of the form $\boldsymbol{\lambda}^{\boldsymbol{\kappa}}$, for example, when there are eigentuples with a multiplicity greater than one [44]. Such cases are not considered in this work, however.

The rest of this paper is organized as follows: Section 3.2 introduces the generalized Cayley-Hamilton Theorem, and the multivariate polynomials associated with it. Then, Section 3.3 shows that, under an observability-like condition, for systems with distinct eigentuples, these polynomials are a difference equation representation of an autonomous, scalar, linear, shift-invariant system. A closed-form expression for this set of difference equations in terms of the eigentuples is then presented in Section 3.4.

3.2 The generalized Cayley-Hamilton Theorem

Like the classical Cayley-Hamilton Theorem, the Generalized Cayley-Hamilton (GCH) Theorem, as first presented in [126], describe relations between sets of matrices and their higher-order powers. The ideas behind this construction are illustrated in Example 3.2.1 which is quite simple, but it generalizes easily. While the given matrices are not required to commute for the GCH Theorem, since it will be applied to the state-transition matrices, we present a version adapted to commuting matrices in Theorem 3.2.1 to avoid confusion.

Example 3.2.1. Reconsider the state-space model from Example 3.1.1 and define the matrix $\boldsymbol{B} = c_1\boldsymbol{A}_1 + c_2\boldsymbol{A}_2,\ c_k \in \mathbb{C}$. Now compute the characteristic polynomial of $\boldsymbol{B}$: $\chi(x) = x^2 + x(-b_2c_2 - b_1c_2 - a_2c_1 - a_1c_1) + (b_1c_2 + a_1c_1)(b_2c_2 + a_2c_1)$. Introducing the variables λ_1 and λ_2 and substituting $x = c_1\lambda_1 + c_2\lambda_2$, results in the following equation:

$$\begin{aligned}&\chi(c_1\lambda_1 + c_2\lambda_2) = \\ &\underbrace{(\lambda_1^2 - \lambda_1(a_2+a_1) + a_1a_2)}_{r_{(2,0)}(\lambda_1,\lambda_2)} c_1^2 + \underbrace{(\lambda_2^2 - \lambda_2(b_2+b_1) + b_1b_2)}_{r_{(0,2)}(\lambda_1,\lambda_2)} c_2^2 \\ &+ \underbrace{(2\lambda_1\lambda_2 - \lambda_2(a_2+a_1) - \lambda_1(b_2+b_1) + a_1b_2 + a_2b_1)}_{r_{(1,1)}(\lambda_1,\lambda_2)} c_1c_2,\end{aligned} \tag{3.2}$$

where the multi-index $\boldsymbol{\alpha}$ of the coefficient $r_{\boldsymbol{\alpha}}$ refers to the monomial $\boldsymbol{c}^{\boldsymbol{\alpha}}$ with which it is associated. Due to the Cayley-Hamilton Theorem, expression (3.2) evaluated in $(\lambda_1, \lambda_2) = (\boldsymbol{A}_1, \boldsymbol{A}_2)$, equals zero as a polynomial in c_1, c_2, implying that each of its coefficients $r_{\boldsymbol{\alpha}}$ are zero. Consequently, it holds that $\boldsymbol{0} = r_{(2,0)}(\boldsymbol{A}_1, \boldsymbol{A}_2) = r_{(1,1)}(\boldsymbol{A}_1, \boldsymbol{A}_2) = r_{(0,2)}(\boldsymbol{A}_1, \boldsymbol{A}_2)$. Note that $r_{(2,0)}$ and $r_{(0,2)}$ are the characteristic polynomials of $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$, respectively. ◇

This construction can be easily extended to the general case of m matrices $\boldsymbol{A}_k$ and variables c_k, leading to a set of polynomials $r_{\boldsymbol{\alpha}}(\boldsymbol{\lambda})$, giving rise to the generalized Cayley-Hamilton Theorem, Theorem 3.2.1. Note that only monomials $\boldsymbol{c}^{\boldsymbol{\alpha}}$ of exactly degree n occur in $\chi(\sum_k c_k\boldsymbol{\lambda}_k)$, which can be readily deduced from the cofactor expansion of the determinant. To this end, we denote the set of monomials in m variables and of exactly degree n as $\mathcal{A}_n = \{\boldsymbol{\alpha} |\ \boldsymbol{\alpha} \in \mathbb{N}_0^m, \|\boldsymbol{\alpha}\|_1 = n\}$, implying that the GCH construction results in $\#\mathcal{A}_n = \binom{m+n-1}{m-1}$ equations.

Theorem 3.2.1 (GCH [89, 126])**.** *Given m pairwise commuting matrices $\boldsymbol{A}_1, \ldots, \boldsymbol{A}_m \in \mathbb{C}^{n\times n}$. Let χ be the characteristic polynomial of the polynomial matrix $\sum_{k=1}^m c_k\boldsymbol{A}_k = \boldsymbol{B} \in \mathbb{C}^{n\times n}[\boldsymbol{c}]$. Define the variables $\boldsymbol{\lambda}$ and define the polynomials $r_{\boldsymbol{\alpha}}$ by writing $\chi(\sum_k c_k\lambda_k)$ such that:*

$$\chi(\sum_k c_k\lambda_k) = \sum_{\boldsymbol{\alpha}\in\mathcal{A}_n} r_{\boldsymbol{\alpha}}(\lambda_1, ..., \lambda_m)\boldsymbol{c}^{\boldsymbol{\alpha}}. \tag{3.3}$$

It then holds that $\forall\boldsymbol{\alpha} \in \mathcal{A}_n : r_{\boldsymbol{\alpha}}(\boldsymbol{A}_1, \ldots, \boldsymbol{A}_m) = \boldsymbol{0}$.

3.3 Obtaining difference equations through the Generalized Cayley-Hamilton Theorem

As illustrated in Example 3.2.1, the application of the Generalized Cayley-Hamilton Theorem to a set of commuting matrices can be used to build a set

of multivariate polynomials $\{r_{\boldsymbol{\alpha}}(\boldsymbol{\lambda})|\ \boldsymbol{\alpha} \in \mathcal{A}_n\}$ with a finite subset of predefined roots, i.e., the eigentuples of the given matrices. Such a set of multivariate polynomials then naturally leads to a set of homogeneous, scalar mD difference equations by evaluating them in the shift operators $\boldsymbol{\sigma}$, as discussed in Section 3.1. In this section, we show that the eigentuples of the state-transition matrices, when distinct, correspond exactly to the common roots of $r_{\boldsymbol{\alpha}}$, implying that both representations describe the same set of model outputs. This is first illustrated in Example 3.3.1 below.

Example 3.3.1. Reconsider the polynomials $\boldsymbol{r}(\boldsymbol{\sigma})^{\mathsf{T}} = \begin{bmatrix} r_{(2,0)}(\boldsymbol{\sigma}) & r_{(1,1)}(\boldsymbol{\sigma}) & r_{(0,2)}(\boldsymbol{\sigma}) \end{bmatrix}$ as defined in (3.2). While it is obvious from Theorem 3.2.1 that (a_1, b_1) and (a_2, b_2) are common roots of these polynomials, it is not immediately clear that they form the complete set of common roots. Indeed, using the same grid ordering as in Example 3.1.1, writing out $\boldsymbol{r}(\boldsymbol{\sigma})\hat{y}_{(0,0)} = \mathbf{0}$ readily leads to values for $\hat{y}_{(2,0)}, \hat{y}_{(1,1)}, \hat{y}_{(0,2)}$, which thus seems to define a recurrence relation with $\hat{y}_{(0,0)}, \hat{y}_{(1,0)}, \hat{y}_{(0,1)}$ as initial values. This would imply that there are three basis signals instead of the two of the original model in Example 3.1.1. Indeed, the difference equations $\boldsymbol{r}(\boldsymbol{\sigma})$ impose restrictions on the signal at all grid points, i.e., $\forall \boldsymbol{\kappa} \in \mathbb{N}_0^2 : \boldsymbol{r}(\boldsymbol{\sigma})\hat{y}_{\boldsymbol{\kappa}} = \boldsymbol{r}(\boldsymbol{\sigma})\boldsymbol{\sigma}^{\boldsymbol{\kappa}}\hat{y}_{(0,0)} = \boldsymbol{\sigma}^{\boldsymbol{\kappa}}\boldsymbol{r}(\boldsymbol{\sigma})\hat{y}_{(0,0)} = \mathbf{0}$, not only in $\boldsymbol{\kappa} = (0, 0)$. For example, let $(a_1, b_1) = (\frac{1+i}{\sqrt{2}}, i)$ and $(a_2, b_2) = (\frac{1-i}{\sqrt{2}}, -i)$ in Equation (3.2), we can then write out the difference equations at the grid points $\|\boldsymbol{\kappa}\|_1 \leq 1$ as a linear system:

$$\mathbf{0} = \boldsymbol{M}\hat{\boldsymbol{y}} =$$

$$\begin{array}{r} \\ \boldsymbol{r}(\boldsymbol{\sigma}) \\ \\ \\ \sigma_1\boldsymbol{r}(\boldsymbol{\sigma}) \\ \\ \\ \sigma_2\boldsymbol{r}(\boldsymbol{\sigma}) \\ \\ \end{array} \left[\begin{array}{ccccccccccc} 1 & -\sqrt{2} & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \sqrt{2} & 0 & -\sqrt{2} & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & -\sqrt{2} & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & \sqrt{2} & 0 & 0 & -\sqrt{2} & 0 & 0 & 2 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 1 & 0 & -\sqrt{2} & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & \sqrt{2} & 0 & 0 & -\sqrt{2} & 0 & 0 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right] \begin{bmatrix} \hat{y}_{(0,0)} \\ \hat{y}_{(1,0)} \\ \hat{y}_{(0,1)} \\ \hat{y}_{(2,0)} \\ \hat{y}_{(1,1)} \\ \hat{y}_{(0,2)} \\ \hat{y}_{(3,0)} \\ \hat{y}_{(2,1)} \\ \hat{y}_{(1,2)} \\ \hat{y}_{(0,3)} \end{bmatrix}.$$

Clearly, constructing $\hat{\boldsymbol{y}}$ appropriately from any signal $\hat{y}_{\boldsymbol{\kappa}}$ in the kernel of $\boldsymbol{r}(\boldsymbol{\sigma})$ solves this system. Moreover, the right null space of $\boldsymbol{M}$ is of dimension two, such that the basis signals $(\frac{1+i}{\sqrt{2}})^{\kappa_1}(i)^{\kappa_2}$ and $(\frac{1-i}{\sqrt{2}})^{\kappa_1}(-i)^{\kappa_2}$ generate two basis vectors for it. Note that these basis vectors correspond to the columns of a 10×2 extended version of the observability matrix $\boldsymbol{\Gamma}$ from Example 3.1.1, from which it was deduced that for the given ordering, $\hat{y}_{(0,0)}$ and $\hat{y}_{(1,0)}$ act as initial

values. Now, like in Example 3.1.1 consider the case where $a_1 = a_2$ and $b_1 \neq b_2$, e.g., let $(a_1, b_1) = (1, 1)$ and $(a_2, b_2) = (1, 2)$, a similar analysis then reveals that the basis signals $1^{\kappa_1}1^{\kappa_2}$ and $1^{\kappa_1}2^{\kappa_2}$ again lead to two basis vectors $\hat{\boldsymbol{y}}$ spanning null($\boldsymbol{M}$). Thus, as discussed in Example 3.1.1 $\hat{y}_{(0,0)}$ and $\hat{y}_{(0,1)}$ act as initial values and the GCH parameterization does not break down.

In general, as long as enough grid points are considered, i.e., $\|\boldsymbol{\kappa}\|_1 \leq d$ for d 'sufficiently large', it has been shown that for any set of polynomials $\boldsymbol{r}(\boldsymbol{\sigma})$, the kernel of the associated *Macaulay matrix* $\boldsymbol{M}$ is spanned by vectors similar to $\hat{\boldsymbol{y}}$, generated by the basis signals[5] $\boldsymbol{\lambda}_i^{\boldsymbol{\kappa}}$, where $\boldsymbol{\lambda}_i$ are the common roots of the multivariate polynomials $\boldsymbol{r}(\boldsymbol{\sigma})$ [12]. ◇

We now extend the observation of Example 3.3.1 to general n and m. First, note that the state-transition matrices in (3.1) can be brought into upper triangular form through a common Schur transform, as stated in Lemma 3.3.1. Notice that this reveals the eigenvalues of the state-transition matrices on the diagonals, allowing us to group them into eigentuples.

Lemma 3.3.1 ([82])**.** ***Schur triangularization for commuting matrices:*** *Let $\boldsymbol{A}_k \in \mathbb{C}^{n\times n}, k = 1, \ldots, m$ be a set of commuting matrices, then there exists a unitary $\boldsymbol{Q} \in \mathbb{C}^{n\times n}$ such that $\boldsymbol{T}_k = \boldsymbol{Q}^{-1}\boldsymbol{A}_k\boldsymbol{Q}$ are upper triangular.*

For the practical computation of $\boldsymbol{Q}$, we refer the reader to [29]. Now, by Lemma 3.3.1 and since $\boldsymbol{Q}^{-1}\boldsymbol{B}\boldsymbol{Q} = \boldsymbol{Q}^{-1}(c_1\boldsymbol{A}_1 + \cdots + c_m\boldsymbol{A}_m)\boldsymbol{Q} = c_1\boldsymbol{T}_1 + \cdots + c_m\boldsymbol{T}_m$, the characteristic of polynomial $\boldsymbol{B}$ and, by extension, the GCH equations defined in Theorem 3.2.1 are invariant under a common similarity transform on the matrices $\boldsymbol{A}_k$, so that we can assume these to be upper triangular without loss of generality. Denoting the eigentuples, i.e., the m-tuples of ith diagonal entries, as $\boldsymbol{a}_i = \begin{bmatrix}(\boldsymbol{T}_1)_{i,i} & (\boldsymbol{T}_2)_{i,i} & \cdots & (\boldsymbol{T}_m)_{i,i}\end{bmatrix}^\mathsf{T}$, $i = 1, \ldots, n$, the characteristic polynomial of $\boldsymbol{B}$ is then $\chi(x) = \prod_{i=1}^{n}(x - \boldsymbol{c}\cdot\boldsymbol{a}_i)$, where $\cdot$ denotes the dot product, since the determinant of an upper triangular matrix is equal to the product of its diagonal entries. Finally, to construct the GCH equations, the variables $\boldsymbol{\lambda} \in \mathbb{C}^m$ are substituted into the polynomial:

$$\chi(\boldsymbol{c}\cdot\boldsymbol{\lambda}) = \prod_{i=1}^{n}(\boldsymbol{c}\cdot\boldsymbol{\lambda} - \boldsymbol{c}\cdot\boldsymbol{a}_i) = \prod_{i=1}^{n}\boldsymbol{c}\cdot(\boldsymbol{\lambda} - \boldsymbol{a}_i). \tag{3.4}$$

This can then be rewritten to obtain the set of m-variate polynomials $\{r_{\boldsymbol{\alpha}}(\boldsymbol{\lambda}) \mid \boldsymbol{\alpha} \in \mathcal{A}_n\}$, as constructed in Theorem 3.2.1, whose common roots of are described in Theorem 3.3.1.

[5]The common roots of the polynomials are revealed by the eigentuples of the state-transition matrices, which can be obtained through the shift equations $\boldsymbol{\Gamma}_{\boldsymbol{\kappa}_l,:}\boldsymbol{A}_k = \boldsymbol{\Gamma}_{\boldsymbol{\kappa}_l+\boldsymbol{e}_k,:}$ built from the observability matrix $\boldsymbol{\Gamma}$ in the null space of a Macaulay matrix.

Theorem 3.3.1. *Given a set of m pairwise commuting matrices $\mathcal{M} = \{\boldsymbol{A}_k \in \mathbb{C}^{n\times n} \mid k = 1,\dots,m\}$. As in Lemma 3.3.1, jointly triangularize these matrices resulting in $\boldsymbol{T}_k$ and define the eigentuples as $\boldsymbol{a}_i = \begin{bmatrix}(\boldsymbol{T}_1)_{i,i} & \cdots & (\boldsymbol{T}_m)_{i,i}\end{bmatrix}^\mathsf{T}$, where $i = 1,\dots,n$. Let the set of $\binom{m+n-1}{m-1}$ equations $\{r_{\boldsymbol{\alpha}}(\boldsymbol{\lambda}) \mid \boldsymbol{\alpha} \in \mathcal{A}_n\}$ be the GCH polynomials constructed from $\mathcal{M}$, as described in Theorem 3.2.1. The set of affine common solutions[6] of these GCH polynomials is then precisely $V = \{\boldsymbol{a}_i \mid i = 1,\dots,n\}$.*

Moreover, when $\boldsymbol{a}_1,\dots,\boldsymbol{a}_n$ comprise n unique tuples, i.e., $\forall i,j \in \{1,\dots,n\},\ i \neq j:\ \boldsymbol{a}_i \neq \boldsymbol{a}_j$, each solution in V has a multiplicity of precisely one.

Proof. As previously noted, the GCH polynomials are invariant under common similarity transforms applied to $\boldsymbol{A}_k$. We thus assume the upper triangular format without loss of generality. Let $\boldsymbol{\lambda}^*$ be a solution for the set of polynomials $\{r_{\boldsymbol{\alpha}}(\boldsymbol{\lambda}) \mid \boldsymbol{\alpha} \in \mathcal{A}_n\}$. By Equations (3.4) and (3.3), this implies that $\forall\, \boldsymbol{c} \in \mathbb{C}^m$: $\chi(\boldsymbol{c}\cdot\boldsymbol{\lambda}^*) = \sum_{\boldsymbol{\alpha}\in\mathcal{A}_n} r_{\boldsymbol{\alpha}}(\boldsymbol{\lambda}^*)\cdot\boldsymbol{c}^{\boldsymbol{\alpha}} = \prod_{i=1}^{n} \boldsymbol{c}\cdot(\boldsymbol{\lambda}^* - \boldsymbol{a}_i) = 0$. This can only equal zero, regardless of the value of $\boldsymbol{c}$, when $\exists i \in \{1,\dots,n\}:\ \boldsymbol{\lambda}^* - \boldsymbol{a}_i = \boldsymbol{0}$.

To prove the absence of higher-order multiplicities when there are n unique solutions, let $\boldsymbol{v}\cdot\nabla_{\boldsymbol{\lambda}}$ be any linear combination of first-order partial differentiation operators w.r.t. the variables $\boldsymbol{\lambda}$, that is, $\nabla_{\boldsymbol{\lambda}} = \begin{bmatrix}\frac{\partial}{\partial\lambda_1} & \cdots & \frac{\partial}{\partial\lambda_m}\end{bmatrix}^\mathsf{T}$ and $\boldsymbol{v} \in \mathbb{C}^m$. Applying this differentiation operator to $\chi(\boldsymbol{c}\cdot\boldsymbol{\lambda})$ leads to the following expression, where we denote $\frac{\mathrm{d}\chi(x)}{\mathrm{d}x}$ by $\chi'(x)$:

$$\begin{aligned}(\boldsymbol{v}\cdot\nabla_{\boldsymbol{\lambda}})\chi(\boldsymbol{c}\cdot\boldsymbol{\lambda}) &= \sum_{\boldsymbol{\alpha}\in\mathcal{A}_n} (\boldsymbol{v}\cdot\nabla_{\boldsymbol{\lambda}}) r_{\boldsymbol{\alpha}}(\boldsymbol{\lambda})\boldsymbol{c}^{\boldsymbol{\alpha}} \\ &= \chi'(\boldsymbol{c}\cdot\boldsymbol{\lambda})(\boldsymbol{v}\cdot\nabla_{\boldsymbol{\lambda}})(\boldsymbol{c}\cdot\boldsymbol{\lambda}) = \chi'(\boldsymbol{c}\cdot\boldsymbol{\lambda})(\boldsymbol{v}\cdot\boldsymbol{c}).\end{aligned} \tag{3.5}$$

Now assume that the set of equations has a multiplicity of at least two for some solution $\boldsymbol{\lambda}^*$, that is, $\exists \boldsymbol{v} \in \mathbb{C}^m\backslash\{\boldsymbol{0}\}$ such that $\forall\boldsymbol{\alpha} : r_{\boldsymbol{\alpha}}(\boldsymbol{\lambda}^*) = (\boldsymbol{v}\cdot\nabla_{\boldsymbol{\lambda}}) r_{\boldsymbol{\alpha}}(\boldsymbol{\lambda}^*) = 0$ [34]. It then follows that $\forall\boldsymbol{c} \in \mathbb{C}^m : (\boldsymbol{v}\cdot\nabla_{\boldsymbol{\lambda}})\chi(\boldsymbol{c}\cdot\boldsymbol{\lambda}^*) = 0$, due to the first equality in (3.5). Then, due to the third equality in (3.5), it is clear that: $\forall\boldsymbol{c} \in \mathbb{C}^m$ such that $\boldsymbol{c}\cdot\boldsymbol{v} \neq 0$ it holds that $\chi'(\boldsymbol{c}\cdot\boldsymbol{\lambda}^*) = 0$.

Applying the product rule for differentiation to $\chi'(\boldsymbol{c}\cdot\boldsymbol{\lambda})$ results in:

$$\chi'(\boldsymbol{c}\cdot\boldsymbol{\lambda}) = \sum_{j=1}^{n}\prod_{i=1,i\neq j}^{n} \boldsymbol{c}\cdot(\boldsymbol{\lambda} - \boldsymbol{a}_i). \tag{3.6}$$

Since any solution $\boldsymbol{\lambda}^*$ is an element of V, i.e., $\boldsymbol{\lambda}^* = \boldsymbol{a}_k$ for some $k \in \{1,\dots,n\}$. However, when $\boldsymbol{a}_1,\dots,\boldsymbol{a}_n$ comprise n unique tuples, all the factors $\boldsymbol{c}\cdot(\boldsymbol{\lambda} - \boldsymbol{a}_i)$

[6] Additional solutions in projective space, i.e., *solutions at infinity*, are excluded since homogenizing with a variable λ_0 and setting $\lambda_0 = 0$ results in $r_{\boldsymbol{\alpha}}(0,\boldsymbol{\lambda}) = \boldsymbol{\lambda}^{\boldsymbol{\alpha}}$ (see Section 3.4), which only vanishes in $(\lambda_0,\boldsymbol{\lambda}) = \boldsymbol{0}$.

are unique, implying that when we evaluate (3.6) in $\boldsymbol{\lambda} = \boldsymbol{a}_k$, the kth term is the only one without the factor $\boldsymbol{c} \cdot (\boldsymbol{\lambda} - \boldsymbol{a}_k)$ and thus the only one to not vanish. Additionally, when $\boldsymbol{\lambda} = \boldsymbol{a}_k$ it holds that $\boldsymbol{\lambda} - \boldsymbol{a}_i \neq 0$, where $i \neq k$, such that the kth term cannot vanish for all values of $\boldsymbol{c}$ in the open set $\{\boldsymbol{c} \in \mathbb{C}^m : \boldsymbol{c} \cdot \boldsymbol{v} \neq 0\}$, thus contradicting the initial assumption that a solution to the equations with a multiplicity of at least two exists. □

Remark: Note that Equation (3.6) implies that in case there is a multiple in the set of diagonal entries, i.e., $\exists i, j \in \{1, \ldots, n\}, i \neq j : \boldsymbol{a}_i = \boldsymbol{a}_j$, $\chi'(\boldsymbol{c} \cdot \boldsymbol{\lambda})$ will vanish in said solution point $\boldsymbol{a}_i$, regardless of $\boldsymbol{c}$. By (3.5), this implies that all the first-order partial derivatives $(\boldsymbol{v} \cdot \nabla_{\boldsymbol{\lambda}}) r_{\boldsymbol{\alpha}}(\boldsymbol{\lambda})$ vanish in $\boldsymbol{a}_i$, implying a multiplicity of at least $m + 1$, while generally, multiplicities in the range 2 to m can be achieved [34]. As such, parameterizations based on the GCH construction may describe models of an order higher than n in these cases, though not of infinite order. Moreover, the parameterization is more general than those obtained by the Buchberger-Möller algorithm, since for any choice of initial values, the so-called *normal set*, an infinite, measure zero collection of sets of distinct eigentuples exists for which this normal set is invalid, as illustrated in Example 3.1.1. [169]

Lastly, we show that, under a condition akin to observability, the state-space model (3.1) and the difference equations built from the GCH equations describe exactly the same set of output signals. Theorem 3.3.2 shows this equivalence.

Theorem 3.3.2. *Let $\boldsymbol{A}_k \in \mathbb{C}^{n \times n}, k = 1, \ldots, m$ be a set of upper triangular commuting matrices, define a state-space model* (3.1) *with these matrices and $\boldsymbol{C} \in \mathbb{R}^{1 \times n}$, and denote $\boldsymbol{a}_i = \begin{bmatrix} (\boldsymbol{A}_1)_{i,i} & \cdots & (\boldsymbol{A}_m)_{i,i} \end{bmatrix}^\mathsf{T}$. If $\forall i, j \in \{1, \ldots, n\}$, $i \neq j : \boldsymbol{a}_i \neq \boldsymbol{a}_j$, and there exists a finite set of grid points $\boldsymbol{\kappa}_l$, $l = 1, \ldots, N$ such that the matrix $\boldsymbol{\Gamma} \in \mathbb{R}^{N \times n} : \boldsymbol{\Gamma}_{l,:} = \boldsymbol{C}\boldsymbol{A}^{\boldsymbol{\kappa}_l}$, is of rank n, then, the set of signals described by the GCH difference equations constructed from $\boldsymbol{A}_k$ in Theorem 3.3.1 equals the set of signals described by the state-space model.*

Proof. Let $\boldsymbol{r}(\boldsymbol{\sigma}) \in \mathbb{C}^{\#\mathcal{A}_n \times 1}[\boldsymbol{\sigma}]$ denote the GCH difference equations. As established in Section 3.1 and due to Theorem 3.3.1 any signal $\hat{y}_{\boldsymbol{\kappa}} : \mathbb{N}_0^m \to \mathbb{R}; \boldsymbol{\kappa} \mapsto \hat{y}_{\boldsymbol{\kappa}}$ that is, model-compliant, i.e., $\boldsymbol{r}(\boldsymbol{\sigma})\hat{y}_{\boldsymbol{\kappa}} = \boldsymbol{0}$, can be written as $\hat{y}_{\boldsymbol{\kappa}} = \sum_{i=1}^{n} \gamma_i \boldsymbol{a}_i^{\boldsymbol{\kappa}}$, where $\gamma_i \in \mathbb{C}$, such that the associated kernel is the span of signals $\mathcal{Y}_{GCH} = \operatorname{span}(\boldsymbol{a}_1^{\boldsymbol{\kappa}}, \ldots, \boldsymbol{a}_n^{\boldsymbol{\kappa}})$. The outputs of the state-space model lie in the span of signals $\operatorname{span}(\boldsymbol{C}\boldsymbol{A}^{\boldsymbol{\kappa}}\boldsymbol{e}_1, \ldots, \boldsymbol{C}\boldsymbol{A}^{\boldsymbol{\kappa}}\boldsymbol{e}_n) = \mathcal{Y}_{ss}$, where $\boldsymbol{e}_i$ is the ith column of the identity matrix. All entries of the vector signal $\boldsymbol{C}\boldsymbol{A}^{\boldsymbol{\kappa}}$ then lie in $\mathcal{Y}_{GCH}$, since by Theorem 3.2.1, $\forall \boldsymbol{\kappa} \in \mathbb{N}_0^m : \boldsymbol{r}(\boldsymbol{\sigma})\boldsymbol{C}\boldsymbol{A}^{\boldsymbol{\kappa}} = \boldsymbol{C}\boldsymbol{r}(\boldsymbol{A}_1, \ldots, \boldsymbol{A}_m)\boldsymbol{A}^{\boldsymbol{\kappa}} = \boldsymbol{0}$. Consequently, $\mathcal{Y}_{ss}$ is a linear subspace of $\mathcal{Y}_{GCH}$, more precisely: $\exists \boldsymbol{V} \in \mathbb{C}^{n \times n} : \forall \boldsymbol{\kappa} \in \mathbb{N}_0^m : \begin{bmatrix} \boldsymbol{a}_1^{\boldsymbol{\kappa}} & \cdots & \boldsymbol{a}_n^{\boldsymbol{\kappa}} \end{bmatrix} \boldsymbol{V} = \boldsymbol{C}\boldsymbol{A}^{\boldsymbol{\kappa}}$. Now, $\boldsymbol{V}$ is of rank n and thus invertible, because $\boldsymbol{\Gamma}$ is of rank n and $\forall l : \boldsymbol{\Gamma}_{l,:} = \begin{bmatrix} \boldsymbol{a}_1^{\boldsymbol{\kappa}_l} & \cdots & \boldsymbol{a}_n^{\boldsymbol{\kappa}_l} \end{bmatrix} \boldsymbol{V}$. This implies that

$\forall \boldsymbol{\kappa} \in \mathbb{N}_0^m : \begin{bmatrix} \boldsymbol{a}_1^{\boldsymbol{\kappa}} & \cdots & \boldsymbol{a}_n^{\boldsymbol{\kappa}} \end{bmatrix} = \boldsymbol{C}\boldsymbol{A}^{\boldsymbol{\kappa}}\boldsymbol{V}^{-1}$, such that $\mathcal{Y}_{GCH}$ is a linear subspace[7] of $\mathcal{Y}_{ss}$, and thus $\mathcal{Y}_{GCH} = \mathcal{Y}_{ss}$. □

3.4 Closed-form expression

In this section, we derive an explicit expression for the GCH polynomials $r_{\boldsymbol{\alpha}}(\boldsymbol{\lambda})$ in terms of their given set of roots. The construction of a specific equation $r_{\boldsymbol{\alpha}}$ can then be viewed as a combinatorial problem, which will first be outlined using the Example 3.4.1, Afterwards, we derive a general formulation in a similar fashion. For an easier exposition, we first introduce the function $\text{count}(\boldsymbol{k}, j) : \mathbb{N}_0^n \times \mathbb{N} \to \mathbb{N}_0; (\boldsymbol{k}, j) \mapsto \#\{i \in \{1, \ldots, n\} |\ k_i = j\}$, which counts the number of occurrences of j in the n-tuple $\boldsymbol{k}$.

Now, let $a_{i,k}$ denote the kth entry of $\boldsymbol{a}_i$, then by (3.3) and (3.4), it holds that:

$$\sum_{\boldsymbol{\alpha} \in \mathcal{A}_n} r_{\boldsymbol{\alpha}}(\boldsymbol{\lambda}) \cdot \boldsymbol{c}^{\boldsymbol{\alpha}} = \prod_{i=1}^{n} \left(\sum_{k=1}^{m} c_k (\lambda_k - a_{i,k}) \right). \tag{3.7}$$

Example 3.4.1. Consider the GCH polynomials constructed from the matrices $\boldsymbol{A}_1 = \text{diag}(1, 2, 3)$ and $\boldsymbol{A}_2 = \text{diag}(3, 2, 1)$, such that $\boldsymbol{a}_1 = \begin{bmatrix} 1 & 3 \end{bmatrix}^\mathsf{T}, \boldsymbol{a}_2 = \begin{bmatrix} 2 & 2 \end{bmatrix}^\mathsf{T}$ and $\boldsymbol{a}_3 = \begin{bmatrix} 3 & 1 \end{bmatrix}^\mathsf{T}$ and $\boldsymbol{c} = \begin{bmatrix} c_1 & c_2 \end{bmatrix}^\mathsf{T}$. To keep the exposition brief, we focus on building the single polynomial $r_{(2,1)}$, i.e., for $\boldsymbol{\alpha} = (2, 1)$. For notational conciseness, denote $d_{i,k} = (\lambda_k - a_{i,k})$ and expand the right-hand side of (3.7):

$$(c_1 d_{1,1} + c_2 d_{1,2})(c_1 d_{2,1} + c_2 d_{2,2})(c_1 d_{3,1} + c_2 d_{3,2}). \tag{3.8}$$

This expression can be expanded into eight individual terms, each corresponding to one of the four third-degree monomials in $\boldsymbol{c}$. To construct $r_{(2,1)}$, we filter out the terms associated with $\boldsymbol{c}^{(2,1)} = c_2^2 c_1$, as stated in the construction procedure in Theorem 3.2.1, resulting in:

$$r_{(2,1)} = \underbrace{d_{1,1} d_{2,1} d_{3,2}}_{(1,1,2)} + \underbrace{d_{1,1} d_{2,2} d_{3,1}}_{(1,2,1)} + \underbrace{d_{1,2} d_{2,1} d_{3,1}}_{(2,1,1)}. \tag{3.9}$$

The $(1, 1, 2)$-term in (3.9) then corresponds to selecting the first term in the first factor of (3.8), the first term in the second factor and the second term in the third factor, as encoded by the tuple $(1, 1, 2)$. The selection pattern is similarly encoded for the other terms. Notice that for each tuple $\boldsymbol{k}$ associated with a term in (3.9), it holds that $\text{count}(\boldsymbol{k}, 1) = 2$ and $\text{count}(\boldsymbol{k}, 2) = 1$, ensuring the product involves a factor $\boldsymbol{c}^{(2,1)}$.

[7] Note that $\boldsymbol{V}^{-1}$ essentially transforms the state-space outputs into the outputs of a model defined by the diagonal matrices $\boldsymbol{D}_k = \text{diag}((\boldsymbol{a}_1)_k, \ldots, (\boldsymbol{a}_n)_k)$, implying that a similarity transform exists that jointly diagonalizes the matrices $\boldsymbol{A}_k$.

Now, we further expand each of these terms. Let us focus on the $(1,1,2)$-term. Filling in $d_{i,k} = (\lambda_k - a_{i,k})$ results in:

$$d_{1,1}d_{2,1}d_{3,2} = (\lambda_1 - 1)(\lambda_1 - 2)(\lambda_2 - 1). \tag{3.10}$$

To further expand this in terms of monomials in (λ_1, λ_2), one has to choose between the constant or the λ_k-term in each of these three factors. We focus on the λ_k terms first. Notice that λ_1 can achieve any exponent $\beta_1 \in \{0,1,2\}$ and, similarly, λ_2 can achieve any exponent $\beta_2 \in \{0,1\}$ in the terms of the expansion of Equation (3.10). In short, $\boldsymbol{\beta} = (\beta_1, \beta_2) \leq \boldsymbol{\alpha} = (2,1)$, where $\leq$ denotes elementwise comparison. A similar observation holds for the other selections patterns.

The latter observation shows which (λ_1, λ_2)-monomials occur in $r_{(2,1)}$. As such, we now consider a specific monomial, for instance, $\boldsymbol{\lambda}^{(1,1)}$, and attempt to construct its coefficient. Note that, by the full expansion of Equations (3.8), (3.7), this monomial can only be obtained through a factor of the form $\boldsymbol{c}^{(1,1)}\boldsymbol{\lambda}^{(1,1)}$, which still needs to be multiplied by a factor involving c_1 to arrive at the required factor $\boldsymbol{c}^{(2,1)}$. Since we fixed the monomial $\boldsymbol{\lambda}^{(1,1)}$, we can only choose factors of the form $-c_k a_{i,k}$ to achieve this. As such, disregarding the λ_k-part of $d_{i,k}$, we can inspect the eligible parts of the individual factors of (3.8):

$$(-c_1 - 3c_2), (-2c_1 - 2c_2), (-3c_1 - c_2) \tag{3.11}$$

Clearly, to achieve the goal of multiplying by c_1, we can select the first term in either the first, second or third factor, selection patterns that can be encoded as $(1,0,0), (0,1,0)$ and $(0,0,1)$, respectively. Notice that for these selections $\boldsymbol{k}$: $\text{count}(\boldsymbol{k}, 1) = 1$ and $\boldsymbol{k} : \text{count}(\boldsymbol{k}, 2) = 0$. Summing over all these contributions to the coefficient of $\boldsymbol{\lambda}^{(1,1)}$ then yields: $-1 - 2 - 3 = -6$. However, fully expanding (3.9) results in:

$$r_{(2,1)} = -26 + 26\lambda_1 + 11\lambda_2 - 6\lambda_1^2 - 12\lambda_1\lambda_2 + 3\lambda_1^2\lambda_2$$

The missing factor two results from the different permutations by which the monomial $\boldsymbol{\lambda}^{(1,1)}$ can be obtained. For example, selecting the term $-c_1$ corresponds to the selection pattern $(1,0,0)$ in (3.11). For this pattern, one can either select $c_1\lambda_1$ as the second factor and $c_2\lambda_2$ as the third, or vice versa, as both choices result in a monomial $\boldsymbol{c}^{(2,1)}$ when multiplied by $-c_1$. ◇

We now derive an expression for the general case. Using distributivity, Equation (3.7) can be expanded into m^n terms of the form $\prod_i^n c_{k_i}(\lambda_{k_i} - a_{i,k_i})$ for some selection of k-indices: $(k_1, \ldots, k_n) \in \mathbb{N}^n$ and $k_i \leq m$. To arrive at a specific monomial $\boldsymbol{c}^{\boldsymbol{\alpha}}$ in this expansion, one has to make a selection such that: for every $j = 1, \ldots, m$, the jth term is chosen precisely α_j times over all the n

factors. This set of $\boldsymbol{\alpha}$-compatible selections can thus be formally described as $\mathcal{S}^n_{\boldsymbol{\alpha}} = \{\boldsymbol{k} \in \mathbb{N}_0^n \mid k_i \leq m, \forall j \in \{1, \ldots, m\} : \text{count}(\boldsymbol{k}, j) = \alpha_j\}$.

A term corresponding to some $\boldsymbol{\alpha}$-compatible selection can then be expanded further by choosing between the $c_k\lambda_k$ and $-c_k a_{i,k}$ terms in each of the n factors. First, we focus on the former type of terms. Observe that the products of subselections of the $c_k\lambda_k$-type terms lead to contributions of the form $f(\boldsymbol{a}_1, ..., \boldsymbol{a}_n)\boldsymbol{c}^{\boldsymbol{\beta}}\boldsymbol{\lambda}^{\boldsymbol{\beta}}$ for any $\boldsymbol{\beta} \in \mathbb{N}_0^m$ and $\boldsymbol{\beta} \leq \boldsymbol{\alpha}$. Now, given that a factor $\boldsymbol{c}^{\boldsymbol{\beta}}\boldsymbol{\lambda}^{\boldsymbol{\beta}}$ has been obtained through some subselection of terms, it is clear that a factor $\boldsymbol{c}^{\boldsymbol{\alpha}-\boldsymbol{\beta}}$ is still required to arrive at the monomial $\boldsymbol{c}^{\boldsymbol{\alpha}}$. Any $(\boldsymbol{\alpha} - \boldsymbol{\beta})$-compatible subselection of $n - \|\boldsymbol{\alpha} - \boldsymbol{\beta}\|_1$ factors $-c_k a_{i,k}$ provides the required factor $\boldsymbol{c}^{\boldsymbol{\alpha}-\boldsymbol{\beta}}$. Such a selection can be encoded by a tuple $\boldsymbol{k} \in \mathcal{S}^n_{\boldsymbol{\alpha}-\boldsymbol{\beta}}$, in which a zero entry k_i indicates selection of a $c_k\lambda_k$-type factor at position i.

Finally, note that there are multiple subselections by which $\boldsymbol{\beta}$ can be formed. To account for this, one has to count the number of $\boldsymbol{\beta}$-compatible selections out of $\|\boldsymbol{\beta}\|_1$ positions, i.e., $\#S^{\|\boldsymbol{\beta}\|_1}_{\boldsymbol{\beta}}$. To do so, first consider the number of selections of β_1 positions out of $\|\boldsymbol{\beta}\|_1$, i.e., $\binom{\sum_{s\geq 1}\boldsymbol{\beta}_s}{\boldsymbol{\beta}_1}$; next, consider the number of selections of β_2 positions out of $\|\boldsymbol{\beta}\|_1 - \beta_1$, i.e., $\binom{\sum_{s\geq 2}^{m}\beta_s}{\boldsymbol{\beta}_2}$, and so on. Taking the product then results in a total of $\prod_{j=1}^{m}\binom{\sum_{s\geq j}^{m}\beta_s}{\beta_j}$ possibilities.

Combining all the observations above then leads to the following formula in terms of the parameters for the equation associated with multi-index $\boldsymbol{\alpha}$:

$$r_{\boldsymbol{\alpha}}(\boldsymbol{\lambda}) =$$

$$\sum_{\mathbf{0}\leq\boldsymbol{\beta}\leq\boldsymbol{\alpha}}\prod_{j=1}^{m}\binom{\sum_{s\geq j}^{m}\beta_s}{\beta_j}\boldsymbol{\lambda}^{\boldsymbol{\beta}}\sum_{\boldsymbol{k}\in\mathcal{S}_{\boldsymbol{\alpha}-\boldsymbol{\beta}}}(-1)^{\|\boldsymbol{\alpha}-\boldsymbol{\beta}\|_1}\prod_{i:k_i\neq 0}^{n}a_{i,k_i}.$$

To construct the set $\mathcal{S}^n_{\boldsymbol{\alpha}-\boldsymbol{\beta}}$, first construct all selections of $(\boldsymbol{\alpha} - \boldsymbol{\beta})_1$ out of n positions and assign '1' to the selected positions; then, combine this with all possible selections of $(\boldsymbol{\alpha} - \boldsymbol{\beta})_2$ out of the remaining $n - (\boldsymbol{\alpha} - \boldsymbol{\beta})_1$ positions and assign '2'; and so on.

3.5 Conclusion and future work

It was shown that the Generalized Cayley-Hamilton Theorem of [126] can be used to construct a difference equation representation of a given state-space representation of an autonomous, scalar, linear, shift-invariant system with simple eigentuples. Additionally, the proposed methodology allows for the parameterization of a difference equation representation in terms of a general,

finite set of n affine eigentuples of multiplicity precisely one—or equivalently, the parameterization of a set of multivariate polynomial equations in terms of its n solutions with multiplicity one. This generality distinguishes it from the existing approaches in the literature [13, 129], which make a choice of initial states that depends on the precise eigentuples at hand, leading to discontinuities for parameterizations when considering these tuples as model parameters [169]. Additionally, we derived a closed-form expression for the resulting set of equations, expanding each equation in terms of its monomials and the corresponding coefficients.

Since general systems of multivariate polynomials can have both so-called *solutions at infinity* in projective space and solutions with some multiplicity structure [44], we aim to extend the presented methodology to deal with these more general cases in future work. Additionally, we hope to extend this approach to multiple-output systems and systems with inputs. Furthermore, we aim to apply this parameterization to the data-driven estimation of the difference equations [36, 188], To this end, it is imperative that we investigate the numerical accuracy and scalability of this approach.

Chapter 4

Globally optimal least-squares misfit identification of multidimensional, autonomous, linear, shift-invariant systems

Chapter abstract

The problem of least-squares misfit identification of autonomous, scalar, linear, shift-invariant, multidimensional systems from data is considered. The local minima of this optimization problem are characterized as common roots of two different sets of multivariate polynomial equations, one for the image- and one for the kernel-based representation of the model-compliant sequence. Moreover, it is shown that lower-order models cannot be locally optimal, allowing their polynomial characterization to be simplified. It is demonstrated that these optimal models constitute isolated solutions, such that all minima can be found through polynomial root-finding techniques, resulting in a deterministic methodology to obtain the global minimum. The presence of an infinite number of lower-order solutions to these polynomial characterizations is shown as well. Finally, the effectiveness of the proposed methodology is illustrated with numerical examples.

Remark: Much of this chapter generalizes previously published work, which focuses on the 1D case (LTI models):

- Lagauw, S., **Vanpoucke, L.**, and De Moor, B. "Exact characterization of the global optima of least squares realization of autonomous LTI models as a multiparameter eigenvalue problem". 3446–3451. DOI: 10.23919/ECC64448.2024.10591182
- Lagauw, S., **Vanpoucke, L.**, and De Moor, B. "Exact solution to the least squares realization problem as a multiparameter eigenvalue problem". Automatica 185 (2026), 112792. DOI: 10.1016/j.automatica.2025.112792

Most of the current chapter was submitted for review to *Automatica*:

- **Vanpoucke L.**, Lagauw S., De Moor B. "Globally optimal least squares misfit identification of multidimensional, autonomous, linear, shift-invariant systems", Tech. rep. `https://ftp.esat.kuleuven.be/pub/stadius//lvanpouc/misfit.pdf`, Leuven, Belgium: KU Leuven 2026. *Submitted to Automatica*

Big picture

The results in this chapter build on the polynomial nature of the models, as discussed in Chapter 2, which details the different model representations and the transition between them. For the sake of self-sufficiency, the most important concepts are repeated in the introductory sections of this chapter. While we only consider affine models in this chapter, the characterizations of descriptor models and their outputs discussed in 2 may be used to generalize this problem to the case of descriptor models. These insights may also serve to compute the solutions at infinity of the polynomial characterization, which we conjecture to be closely connected to the optimal descriptor models. The construction of Chapter 3 is also used to provide a parametric description of the difference equations in terms of the eigentuples. Unfortunately, the algorithms related to algebraic root finding to be presented in Part II of this dissertation are not generally applicable to the considered misfit problem, due to the infinite number of stationary points therein, as will be discussed. They are, however, applicable to the misfit problems regarding LTI models, i.e., the 1D case, as these do have finite solution sets [36, 103, 104].

4.1 Introduction

The relative simplicity of linear dynamical models, combined with the insight that such models, in many practical cases, provide a sufficiently accurate approximation of the true system dynamics, caused them to become popular modeling tools. The current work focuses on multidimensional (mD) generalizations of the well-known linear time-invariant (LTI) models. More precisely, we focus on autonomous models. Multiple such *linear, shift-invariant model* classes have been proposed in the literature, for example, the state-space models in [6, 8, 22, 56, 88, 100, 162], some more general than others. On the other hand, the behavioral framework for systems theory focuses on the study of models governed by mD difference equations [58, 139, 159, 197, 205]. The latter point of view is particularly relevant to this paper.

As it is not always feasible to obtain such a model through first principles, a common approach is to fit them to given input-output data, especially for LTI models. Moreover, in the one-dimensional case, it has been shown for a variety of modeling approaches that such a fitting problem can be solved exactly when using a least-squares distance measure, that is, there exists a deterministic algorithm guaranteed to find the globally optimal model w.r.t. a given data sequence. Such approaches rely on characterizing stationary points of the underlying optimization problem as common roots of a set of polynomial equations, which can all be found through well-established computational tools [9, 36, 189]. The current work extends this methodology to the least-squares misfit identification of autonomous, scalar, linear, shift-invariant models, extending insights from the 1D autonomous case [36, 104] to the mD setting. Note that the contribution is of a theoretical nature, providing insight into the properties of the underlying optimization problem, and, more precisely, its stationary points.

This approach stands in stark contrast to the identification methods for mD systems present in engineering practice, which all involve some form of heuristics. They either rely on local optimization algorithms—which can at best guarantee that a local optimum is found and the results of which depend strongly on the chosen initialization—or on statistical assumptions that are often unverifiable in practice. Such approaches focus on providing a sufficiently good approximation for engineering applications, rather than completely solving the underlying optimization problem.

Among these ‘heuristic’ multidimensional identification methods, there exist different branches. Most notably, subspace identification methods, which, based on statistical assumptions, construct a state-space model, through efficient linear algebra operations, [59, 84, 154–156]. Optimization-based methods to estimate

the model parameters have also been considered in [28, 51, 52, 71, 182], extending the well-established prediction-error-methods and errors-in-variables methods for LTI models [116, 120, 173]. Another common approach for modeling mD systems 'circumvents' the multidimensionality through distributed parameter models, that is, the models are described as a set of linked 1D models or a single, larger 1D model, to be subsequently identified using 1D techniques, see e.g., [28, 59, 71].

These techniques have been successful in generating useful models in many signal processing applications [22], for example, image processing [115, 155] and texture analysis [41] among others, as well as for multidimensional control techniques [40, 163, 200], and for the parameter estimation of PDEs [51, 144]. For the autonomous models considered here, the problem of multidimensional harmonic retrieval is particularly relevant [45, 86, 164, 168, 184, 185], the techniques of which have been applied in nuclear magnetic resonance spectroscopy [67, 119, 201], the analysis of wireless communications channels [69, 128], and direction of arrival problems [68].

Because a different problem is tackled altogether, it should be emphasized that the current work does not aim to be computationally competitive with such 'heuristic' mD system identification methods. Rather, we strive for theoretical insight into such problems, which may serve to improve these methods—as was done in the one-dimensional case[206]—and provide a means to benchmark them.

4.1.1 Problem formulation

We consider the problem of the least-squares misfit identification of autonomous, scalar, linear, shift-invariant models from a given, finite multi-indexed data sequence. An nth order model in this class allows state-space representation of the form [44, 58, 72]:

$$\begin{aligned} \boldsymbol{x}_{\boldsymbol{\kappa}+\boldsymbol{e}_1} &= \boldsymbol{A}_1 \boldsymbol{x}_{\boldsymbol{\kappa}} \\ \boldsymbol{x}_{\boldsymbol{\kappa}+\boldsymbol{e}_2} &= \boldsymbol{A}_2 \boldsymbol{x}_{\boldsymbol{\kappa}} \\ &\vdots \\ \boldsymbol{x}_{\boldsymbol{\kappa}+\boldsymbol{e}_m} &= \boldsymbol{A}_m \boldsymbol{x}_{\boldsymbol{\kappa}} \\ \hat{y}_{\boldsymbol{\kappa}} &= \boldsymbol{C} \boldsymbol{x}_{\boldsymbol{\kappa}} \\ \text{where} \quad \boldsymbol{A}_k \boldsymbol{A}_l &= \boldsymbol{A}_l \boldsymbol{A}_k \quad \forall k, l \in \{1, \ldots, m\}, \end{aligned} \tag{4.1}$$

and $\boldsymbol{A}_k \in \mathbb{C}^{n\times n}$, $k \in \{1,\ldots,m\}$, $\boldsymbol{C} \in \mathbb{C}^{1\times n}$, $\boldsymbol{x_0} \in \mathbb{C}^n$ and $\boldsymbol{e}_k$ denotes the kth column of the identity matrix. For an initial state $\boldsymbol{x} = \boldsymbol{x_0}$, the output of such a system can thus be written as $\hat{y}_{\boldsymbol{\kappa}} = \boldsymbol{C}\boldsymbol{A}^{\boldsymbol{\kappa}}\boldsymbol{x_0}$, with[1] $\boldsymbol{A}^{\boldsymbol{\kappa}} = \prod_{k=1}^{m} \boldsymbol{A}_k^{\kappa_k}$. Furthermore, the output trajectory remains unaltered under a common similarity transform of the form $\boldsymbol{A}_k^{\text{new}} = \boldsymbol{P}^{-1}\boldsymbol{A}_k\boldsymbol{P}, \boldsymbol{C}^{\text{new}} = \boldsymbol{C}\boldsymbol{P}, \boldsymbol{x}_{\boldsymbol{\kappa}}^{\text{new}} = \boldsymbol{P}^{-1}\boldsymbol{x}_{\boldsymbol{\kappa}}$, where $\boldsymbol{P} \in \mathbb{R}^{n\times n}$ is an invertible matrix. Also note that $\boldsymbol{C}$ and $\boldsymbol{x}$ can be scaled interchangeably, so that one can choose $\boldsymbol{C} = \begin{bmatrix}1 & \ldots & 1\end{bmatrix}$ without loss of generality.

The commuting set of state-transition matrices $\boldsymbol{A}_1,\ldots,\boldsymbol{A}_m$ can be brought into an upper triangular form through a common unitary similarity transformation [82]. This triangular format allows the eigenvalues of $\boldsymbol{A}_1,\ldots,\boldsymbol{A}_m$ on the diagonal entries to be grouped into n m-tuples, which we refer to as eigentuples $\boldsymbol{\lambda}_1,\ldots,\boldsymbol{\lambda}_n \in \mathbb{C}^m$. For notational convenience, the eigentuples are sometimes referred to in matrix format; to this end, let $\begin{bmatrix}\boldsymbol{\lambda}_1 & \ldots & \boldsymbol{\lambda}_n\end{bmatrix}^{\mathsf{T}} = \boldsymbol{\Lambda} \in \mathbb{C}^{n\times m}$.

By lack of a more general parameterization, we confine ourselves to fitting models with n distinct eigentuples, that is, eigentuples of multiplicity one. This assumption implies that the state transition matrices can be chosen as the diagonal matrices $(\boldsymbol{A}_k)_{i,i} = (\boldsymbol{\lambda}_i)_k$ [82]. Models with eigentuples at infinity are also excluded by the model class (4.1) [44].

As will be detailed in Section 4.2, these models also adhere to a set of difference equations with constant coefficients. These equations are essentially a set of multivariate polynomials, whose common roots correspond to the eigentuples [44]. In this work, we are interested in real models, that is, those governed by difference equations with real coefficients. For such difference equations, or rather, multivariate polynomials $\boldsymbol{r}(\boldsymbol{\lambda})$, it holds that $\overline{\boldsymbol{r}(\boldsymbol{\lambda})} = \boldsymbol{r}(\overline{\boldsymbol{\lambda}})$, such that its eigentuples and initial states arise in complex-conjugate pairs. Definition 4.1.1 formalizes this notion for the model parameters $(\boldsymbol{\Lambda}, \boldsymbol{x})$.

Definition 4.1.1 (Real models)**.** Let $\mathcal{R}$ be the set of model parameters $(\boldsymbol{\Lambda}, \boldsymbol{x})$ for which each subset of indices $\mathcal{I} \subseteq \{1,\ldots,n\}$ that satisfies $\forall i, s \in \mathcal{I} : (\boldsymbol{\lambda}_i, x_i) = (\boldsymbol{\lambda}_s, x_s)$, there exists a subset of indices $\mathcal{J} \subseteq \{1,\ldots,n\}$ such that $\forall i \in \mathcal{I}, \exists j \in \mathcal{J} : (\boldsymbol{\lambda}_j, x_j) = \overline{(\boldsymbol{\lambda}_i, x_i)}$, and $\#\mathcal{I} = \#\mathcal{J}$. $(\boldsymbol{\Lambda}, \boldsymbol{x})$ is said to define *a real model* if $(\boldsymbol{\Lambda}, \boldsymbol{x}) \in \mathcal{R}$. △

Given a finite multidimensional data sequence[2] $y_{\boldsymbol{\kappa}} : \mathcal{K} \to \mathbb{R} : \boldsymbol{\kappa} \mapsto y_{\boldsymbol{\kappa}}$ over some finite set of grid points $\mathcal{K} \subseteq \mathbb{N}_0^m$, where $\#\mathcal{K} = N$, and a predefined model order n, the goal of misfit identification is to find model parameters $(\boldsymbol{\Lambda}, \boldsymbol{x})$ associated with the model-compliant signal $\hat{y}_{\boldsymbol{\kappa}}$ that lies nearest to $y_{\boldsymbol{\kappa}}$. Using

[1] *Notation:* A similar multi-exponent notation will be used throughout this work: $\boldsymbol{z}^{\boldsymbol{\kappa}} = \prod_{k=1}^{m} z_k^{\kappa_k}$, where $\boldsymbol{x} \in \mathbb{C}^m$

[2] $\mathbb{N}_0$ denotes the set of natural numbers including zero.

a least-squares measure for distance, the misfit identification problem can be formulated as [116]:

$$\min_{\boldsymbol{\Lambda},\boldsymbol{x}} \|\tilde{\boldsymbol{y}}\|_2^2 \tag{4.2}$$

$$\text{subject to: } \forall \boldsymbol{\kappa} \in \mathcal{K} : \hat{y}_{\boldsymbol{\kappa}} = \prod_{k=1}^{m} \boldsymbol{C}\boldsymbol{A}^{\boldsymbol{\kappa}}\boldsymbol{x}$$

$$\hat{\boldsymbol{y}} = \boldsymbol{y} - \tilde{\boldsymbol{y}}$$

$$(\boldsymbol{\Lambda}, \boldsymbol{x}) \in \mathcal{R},$$

where $\boldsymbol{y} = \begin{bmatrix} y_{\boldsymbol{\kappa}_1} \dots y_{\boldsymbol{\kappa}_N} \end{bmatrix}^{\mathsf{T}} \in \mathbb{R}^N$ is the sequence of given data points and $\tilde{\boldsymbol{y}}, \hat{\boldsymbol{y}} \in \mathbb{R}^N$ are defined similarly. Here, $\tilde{\boldsymbol{y}}$ is the sequence of residuals at each grid point, also referred to as the *misfit*.

The particular ordering[3] of these sequences is of no importance for this problem. Rather, to refer to specific elements of these vectors, we employ the multi-index $\boldsymbol{\kappa}$ directly instead of a sequential index over $\mathcal{K}$.

4.1.2 Contributions and outline

For the considered model class, (4.2) amounts to a non-convex optimization problem, such that there can be many local minima. This complicates the search for the strict solution of this problem, that is, its global minimum. The approach presented in this work characterizes the local minima of this optimization problem as common roots of a set of multivariate polynomial equations. All solutions to these polynomial problems can then be computed using numerical polynomial solvers. By extension, all local minima can thus be obtained, including the global minimum. This methodology provides a deterministic means of solving the misfit problem (4.2).

A brief overview of the main sections of the chapter and the contributions therein is provided in Figure 4.1, which we now briefly elaborate on. First, Section 4.2 introduces the necessary preliminaries regarding the considered model class and its representations.

Next, a relaxed version of the problem is considered in Section 4.3. More precisely, the constraint that the model parameters should define a real model is disregarded, as it is hard to impose analytically. Through manipulation of

[3] In this text, we prefer the use of graded inverse lexicographic ordering when needed. For example, in 2D, this amounts to $(0,0), (1,0), (0,1), (2,0), \dots$ [32].

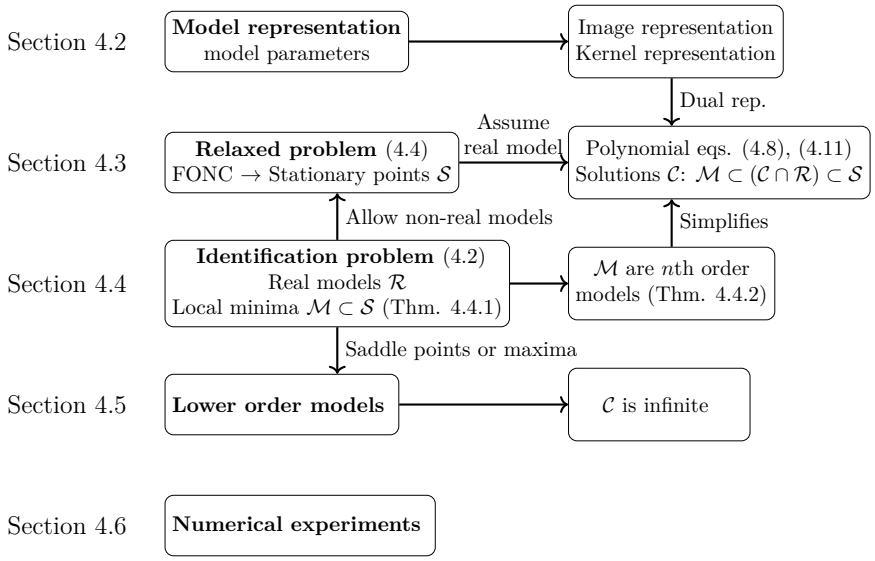


Figure 4.1: Overview of the main sections of this chapter and the contributions therein.

the first-order necessary conditions for optimality (FONC) that follow from this, the aforementioned novel polynomial characterization for the local minima of (4.2) is obtained. A formal proof that these indeed include all local minima of the non-relaxed problem (4.2) is deferred to Section 4.4. Finally, by exploiting the duality between the kernel and image representation of a model-compliant sequence, an alternative polynomial characterization is derived.

Section 4.4 then formally relates the local minima of the original problem (4.2) to the stationary points of the relaxed problem. Moreover, it is shown that these local minima cannot be described using a lower model order, providing theoretical insight into the system-theoretic properties of the minima akin to the LTI case [36, 157]. This property allows us to eliminate part of the non-optimal stationary points from the solution sets of the polynomial characterizations, similar to the approaches in [102, 104]. Lastly, this section shows that these local minima are almost always a finite set of isolated points, allowing them to be found using off-the-shelf numerical polynomial root-finding techniques.

In Section 4.5, the non-optimal, *lower-order* stationary points are investigated. It is illustrated that these arise as infinite sets of equivalent models leading to a positive-dimensional solution component in the polynomial characterizations. Finally, additional numerical experiments are presented in Section 4.6, after which a conclusion is drawn in Section 4.7.

4.2 Autonomous, shift-invariant, linear models

In this section, we briefly explore the properties of the considered model class (4.1). More specifically, an alternative difference equations-based representation of the model is considered, and it is explored how the model-compliant signals can be represented over a finite set of grid points using observability and Macaulay matrices.

4.2.1 Observability matrix

By the linearity of the model class (4.1), the set of output signals generated by such a model forms a finite-dimensional vector space over $\mathbb{C}$ of dimension at most n, referred to as the *behavior* of the model (4.1) [58, 139, 197]. When considering signals over a finite subset of grid points $\mathcal{K} \subset \mathbb{N}_0^m$, this vector space of model-compliant signals is the image of an *observability matrix* $\boldsymbol{\Gamma} \in \mathbb{R}^{N \times n}$, defined as[4] $\boldsymbol{\Gamma}_{\boldsymbol{\kappa},:} = \boldsymbol{C}\boldsymbol{A}^{\boldsymbol{\kappa}}, \boldsymbol{\kappa} \in \mathcal{K}$. For our choice of $\boldsymbol{C}$ and diagonal $\boldsymbol{A}_i$, this can be written out in terms of the model parameters, i.e., the eigentuples $\boldsymbol{\lambda}_i$, as follows:

$$\boldsymbol{\Gamma} = \begin{bmatrix} \boldsymbol{\lambda}_1^{\boldsymbol{\kappa}_1} & \dots & \boldsymbol{\lambda}_n^{\boldsymbol{\kappa}_1} \\ \boldsymbol{\lambda}_1^{\boldsymbol{\kappa}_2} & \dots & \boldsymbol{\lambda}_n^{\boldsymbol{\kappa}_2} \\ \vdots & & \vdots \\ \boldsymbol{\lambda}_1^{\boldsymbol{\kappa}_N} & \dots & \boldsymbol{\lambda}_n^{\boldsymbol{\kappa}_N} \end{bmatrix}.$$

The image of this matrix then describes all model-compliant sequences over $\mathcal{K}$, that is, $\hat{\boldsymbol{y}} = \boldsymbol{\Gamma}\boldsymbol{x}$. As such, this will be referred to as the *image representation*.

4.2.2 Difference equations

Linear, shift-invariant models can also be represented using difference equations with constant coefficients. These multidimensional difference equations can be succinctly written out using *shift operators* $\boldsymbol{\sigma} = (\sigma_1, \dots, \sigma_m)$, which act upon a signal as follows: $\sigma_k y_{\boldsymbol{\kappa}} = y_{\boldsymbol{\kappa}+\boldsymbol{e}_k}$. A model can then be characterized by a set of difference equations, represented by a set of multivariate polynomials in these shift operators $\boldsymbol{r}(\boldsymbol{\sigma}) \in \mathbb{C}^p[\boldsymbol{\sigma}]$, in the sense that a signal $\hat{y}_{\boldsymbol{\kappa}}$ is model-complaint if and only if it satisfies these equations everywhere, i.e., $\boldsymbol{r}(\boldsymbol{\sigma})\hat{y} = 0, \forall \boldsymbol{\kappa} \in \mathbb{N}_0^m$ [58, 139, 159, 205]. The vector space of model-compliant signals then corresponds to the kernel of the difference equations $\boldsymbol{r}(\boldsymbol{\sigma})$. Because of this, it will be referred to as the *kernel representation*.

[4] The subscript ":" denotes that all rows or columns are selected.

Example 4.2.1. Consider the following set of difference equations:

$$\boldsymbol{r}(\boldsymbol{\sigma})\hat{y}_{\boldsymbol{\kappa}} = \begin{bmatrix} \sigma_1^2 - 3\sigma_1 + 2 \\ \sigma_1 - \sigma_2 + 2 \end{bmatrix} \hat{y}_{\boldsymbol{\kappa}} = \begin{bmatrix} \hat{y}_{\boldsymbol{\kappa}+(2,0)} - 3\hat{y}_{\boldsymbol{\kappa}+(1,0)} + 2\hat{y}_{\boldsymbol{\kappa}} \\ \hat{y}_{\boldsymbol{\kappa}+(1,0)} - \hat{y}_{\boldsymbol{\kappa}+(0,1)} + 2\hat{y}_{\boldsymbol{\kappa}} \end{bmatrix} = \boldsymbol{0}, \forall \boldsymbol{\kappa} \in \mathbb{N}_0^2.$$

Any signal $y_{\boldsymbol{\kappa}}$ satisfying this set of equations is then a valid output of the model characterized by the difference equations $\boldsymbol{r}(\boldsymbol{\sigma}) \in \mathbb{C}^2[\boldsymbol{\sigma}]$, which are essentially polynomials in $\boldsymbol{\sigma}$. The polynomial nature of these equations can be leveraged to construct solutions in the form of 2D exponentials $\hat{y}_{\boldsymbol{\kappa}} = \boldsymbol{\lambda}^{\boldsymbol{\kappa}}, \boldsymbol{\lambda} \in \mathbb{C}^2$ [83]:

$$\begin{bmatrix} \boldsymbol{\lambda}^{(2,0)} - 3\boldsymbol{\lambda}^{(1,0)} + 2 \\ \boldsymbol{\lambda}^{(1,0)} - \boldsymbol{\lambda}^{(0,1)} + 2 \end{bmatrix} \boldsymbol{\lambda}^{\boldsymbol{\kappa}} = \boldsymbol{r}(\boldsymbol{\lambda})\boldsymbol{\lambda}^{\boldsymbol{\kappa}} = \boldsymbol{0}, \forall \boldsymbol{\kappa} \in \mathbb{N}_0^2.$$

Clearly, these equations can only be satisfied when $\boldsymbol{\lambda}$ is a solution of $\boldsymbol{r}(\boldsymbol{\lambda}) = \boldsymbol{0}$. The common roots of the difference equations $\boldsymbol{r}(\boldsymbol{\sigma})$ when viewed as a set of multivariate polynomials, $(1, 3)$ and $(2, 4)$ in this case, thus generate a set of exponential basis signals that comply with the model. It can be shown that these basis signals span the full kernel of $\boldsymbol{r}(\boldsymbol{\sigma})$, so that all model-compliant signals are of the form $\hat{y}_{\boldsymbol{\kappa}} = x_1(1, 3)^{\boldsymbol{\kappa}} + x_2(2, 4)^{\boldsymbol{\kappa}}$ [44, 172].

Notice that these model-compliant signals correspond precisely to those generated by a state space model (4.1) with $\boldsymbol{A}_1 = \mathrm{diag}(1, 2)$, $\boldsymbol{A}_2 = \mathrm{diag}(3, 4)$ and $\boldsymbol{C} = \begin{bmatrix} 1 & 1 \end{bmatrix}$, which thus provides an alternative description for this model. Moreover, notice that the eigentuples $\boldsymbol{\lambda}_1 = (1, 3)$ and $\boldsymbol{\lambda}_2 = (2, 4)$ of this state-space model are exactly the common roots of $\boldsymbol{r}(\boldsymbol{\lambda})$. ◇

Example 4.2.1 illustrates how a basis for the signal space can be obtained through solving the difference equations $\boldsymbol{r}(\boldsymbol{\sigma})$ as if they were a set of multivariate polynomials. Due to this close connection with multivariate polynomials, difference equations and multivariate polynomials will often be treated interchangeably. The illustrated relation between the common roots of the difference equations and the eigentuples of an equivalent minimal state-space model also holds in general [44, 72]. Because of this, the state-transition matrices always satisfy the polynomials representing the difference equations[5]: $\boldsymbol{r}(\boldsymbol{A}_1, \ldots, \boldsymbol{A}_m) = \boldsymbol{0}$, such that any output $\hat{y}_{\boldsymbol{\kappa}}$ of (4.1) satisfies the difference equations: $\boldsymbol{0} = \boldsymbol{r}(\boldsymbol{\sigma})\hat{y}_{\boldsymbol{\kappa}} = \boldsymbol{r}(\boldsymbol{\sigma})\boldsymbol{C}\boldsymbol{A}^{\boldsymbol{\kappa}}\boldsymbol{x}_{\boldsymbol{0}} = \boldsymbol{C}\boldsymbol{r}(\boldsymbol{A}_1, \ldots, \boldsymbol{A}_m)\boldsymbol{A}^{\boldsymbol{\kappa}}\boldsymbol{x}_{\boldsymbol{0}}$ [44].

The number of basis signals in the kernel of $\boldsymbol{r}(\boldsymbol{\sigma})$, i.e., the model order n, then equals its total number of common roots, counting multiplicities [12, 172]. Indeed, multiple roots result in additional basis signals of a different format: For example, it can be worked out that[6] $\boldsymbol{r}(\boldsymbol{\sigma})\kappa_k\boldsymbol{\lambda}^{\boldsymbol{\kappa}-\boldsymbol{e}_k} = \boldsymbol{r}(\boldsymbol{\lambda})\kappa_i\boldsymbol{\lambda}^{\boldsymbol{\kappa}-\boldsymbol{e}_k} + \frac{\partial \boldsymbol{r}(\boldsymbol{\lambda})}{\partial \lambda_k}\boldsymbol{\lambda}^{\boldsymbol{\kappa}}$,

[5] For models with simple eigentuples, the $\boldsymbol{A}_k$ are jointly diagonalizable [143]. Applying this diagonalizing similarity transform then yields the desired result: $\boldsymbol{P}^{-1}\boldsymbol{r}(\boldsymbol{A}_1, \ldots, \boldsymbol{A}_m)\boldsymbol{P} = \boldsymbol{r}(\boldsymbol{P}^{-1}\boldsymbol{A}_1\boldsymbol{P}, \ldots, \boldsymbol{P}^{-1}\boldsymbol{A}_m\boldsymbol{P}) = \boldsymbol{0}$, since the eigentuples are common roots of $\boldsymbol{r}(\boldsymbol{\sigma})$.

[6] It suffices to look at the action of σ_k: $\sigma_k\kappa_k\boldsymbol{\lambda}^{\boldsymbol{\kappa}-\boldsymbol{e}_k} = (\kappa_k + 1)\boldsymbol{\lambda}^{\boldsymbol{\kappa}} = \kappa_k\lambda_k\boldsymbol{\lambda}^{\boldsymbol{\kappa}-\boldsymbol{e}_k} + \boldsymbol{\lambda}^{\boldsymbol{\kappa}}$.

which vanishes if and only if $\forall \boldsymbol{\kappa} \in \mathbb{N}_0^m : \boldsymbol{r}(\boldsymbol{\lambda}) = \frac{\partial \boldsymbol{r}(\boldsymbol{\lambda})}{\partial \lambda_i} = 0$, so that $\boldsymbol{\lambda}$ is a common root of multiplicity at least two. Moreover, linear combinations of differential operators can also lead to multiplicities: if $\boldsymbol{r}(\boldsymbol{\lambda}) = \boldsymbol{0}$ and there exists a $\boldsymbol{v} \in \mathbb{C}^m \backslash \{\boldsymbol{0}\} : \boldsymbol{J}_{\boldsymbol{r}}(\boldsymbol{\lambda})\boldsymbol{v} = \boldsymbol{0}$, where $\boldsymbol{J}_{\boldsymbol{r}}$ is the Jacobian of $\boldsymbol{r}(\boldsymbol{\lambda})$, then $\sum_{k=1}^m v_k \kappa_k \boldsymbol{\lambda}^{\boldsymbol{\kappa}-\boldsymbol{e}_k}$ satisfies $\boldsymbol{r}(\boldsymbol{\sigma})$. The vanishing of higher-order partial derivative operators in a common root results in even higher multiplicities [169]. When such multiple roots occur, the state-transition matrices of an equivalent state-space model are not jointly diagonalizable, violating the assumptions made in Section 4.1.1.

4.2.3 Generalized Cayley-Hamilton equations

Given a set of distinct eigentuples $\boldsymbol{\Lambda}$, it is not so straightforward to construct a set of multidimensional difference equations that precisely describe these points. To this end, the *generalized Cayley-Hamilton equations* are introduced in Definition 4.2.1 and their common roots are characterized in Theorem 4.2.1 below.

Definition 4.2.1 (Generalized Cayley-Hamilton (GCH) equations, [89])**.** Given m pairwise commuting matrices $\boldsymbol{A}_k \in \mathbb{C}^{n\times n}, k = 1, \ldots, m$, introduce the variables $\boldsymbol{q} \in \mathbb{C}^m$ and consider the characteristic polynomial $\chi(\alpha, \boldsymbol{q}) = \det(\sum_{k=1}^m q_i \boldsymbol{A}_i - \alpha \boldsymbol{I})$. Then, $\chi(\boldsymbol{q}^\mathsf{T}\boldsymbol{\sigma}, \boldsymbol{q})$ can be written as a polynomial in $\boldsymbol{q}$ with polynomial coefficients in $\boldsymbol{\sigma}$:

$$\chi(\boldsymbol{q}^\mathsf{T}\boldsymbol{\sigma}, \boldsymbol{q}) = \sum_{\boldsymbol{\alpha}\in\mathcal{A}_n} r_{\boldsymbol{\alpha}}(\boldsymbol{\sigma})\boldsymbol{q}^{\boldsymbol{\alpha}}, \tag{4.3}$$

where $\mathcal{A}_n = \{\boldsymbol{\alpha} \in \mathbb{N}_0^m \mid \|\boldsymbol{\alpha}\|_1 = n\}$. The (polynomial) coefficients $r_{\boldsymbol{\alpha}}(\boldsymbol{\sigma})$ then form a set of difference equations called the GCH equations of order n: $\boldsymbol{r}(\boldsymbol{\sigma}) = \begin{bmatrix} r_{\boldsymbol{\alpha}_1} & r_{\boldsymbol{\alpha}_2} & \ldots \end{bmatrix}^\mathsf{T}$. △

Lemma 4.2.1 ([186], Thm. 2)**.** *Given diagonal matrices $\boldsymbol{A}_k \in \mathbb{C}^{n\times n}, k = 1, \ldots, m$, with eigentuples $\boldsymbol{\lambda}_i$. The common roots of the GCH equations $\boldsymbol{r}(\boldsymbol{\sigma})$, as constructed in* (4.3)*, are then given precisely by the set $\{\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_n\}$. Moreover, each solution has a multiplicity of one whenever this set consists of n distinct tuples.*

Using the model parameters $\boldsymbol{\Lambda}$, one can construct a set of diagonal, parametric state-transition matrices $\boldsymbol{A}_i$ as discussed in Section 4.1.1. Applying Lemma 4.2.1 to these matrices then results in a set of parametric difference equations in terms of $\boldsymbol{\Lambda}$. The resulting polynomials then precisely describe the given eigentuples as their solution set, as long as $\boldsymbol{\Lambda}$ consists of distinct eigentuples. This construction is illustrated in Example 4.2.2 below.

Example 4.2.2. Consider a second-order parametric state-space model with eigentuples $\boldsymbol{\lambda}_1 = (\lambda_{1,1}, \lambda_{1,2})$ and $\boldsymbol{\lambda}_2 = (\lambda_{2,1}, \lambda_{2,2})$, resulting in the state-transition matrices:

$$\boldsymbol{A}_1 = \begin{bmatrix} \lambda_{1,1} & \\ & \lambda_{2,1} \end{bmatrix}, \boldsymbol{A}_2 = \begin{bmatrix} \lambda_{1,2} & \\ & \lambda_{2,2} \end{bmatrix}, \boldsymbol{C} = \begin{bmatrix} 1 & 1 \end{bmatrix}.$$

Now, consider the kernel representation. The GCH equations for $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ then lead to the following set of difference equations:

$$\boldsymbol{r}(\boldsymbol{\sigma}) = \begin{bmatrix} \boldsymbol{\sigma}^{(2,0)} - \boldsymbol{\sigma}^{(1,0)} a + b \\ 2\boldsymbol{\sigma}^{(1,1)} - \boldsymbol{\sigma}^{(0,1)} a - \boldsymbol{\sigma}^{(1,0)} c + d \\ \boldsymbol{\sigma}^{(0,2)} - \boldsymbol{\sigma}^{(0,1)} c + e \end{bmatrix},$$

where we introduced the auxiliary variables $a = \lambda_{1,1} + \lambda_{1,2}, b = \lambda_{1,1}\lambda_{1,2}, c = \lambda_{2,1} + \lambda_{2,2}, d = \lambda_{1,1}\lambda_{2,2} + \lambda_{1,2}\lambda_{2,1}, e = \lambda_{2,1}\lambda_{2,2}$. ◇

4.2.4 Macaulay matrix

Over a finite set of grid points $\mathcal{K}$, the difference equations define a system of linear equations and can thus be written in matrix format as $\boldsymbol{M}_{\boldsymbol{r}}\hat{\boldsymbol{y}} = \boldsymbol{0}$. The coefficient matrix $\boldsymbol{M}_{\boldsymbol{r}}$ then is a *Macaulay matrix*, originating from linear algebra techniques for manipulation of and root finding for multivariate polynomials. Definition 4.2.2 compactly describes this type of matrix using multi-indices [12, 44, 123]. The construction is subsequently illustrated in Example 4.2.3.

Definition 4.2.2 (Macaulay matrix)**.** Given a set of difference equations $\boldsymbol{r}(\boldsymbol{\sigma}) = \sum_{\boldsymbol{\alpha}\in\mathcal{A}} \boldsymbol{v}_{\boldsymbol{\alpha}} \boldsymbol{\sigma}^{\boldsymbol{\alpha}} \in \mathbb{C}^p[\boldsymbol{\sigma}], \boldsymbol{v}_{\boldsymbol{\alpha}} \in \mathbb{C}^p$, where $\mathcal{A} \subseteq \mathbb{N}_0^m$ is a finite set of a total degree $d = \max_{\boldsymbol{\alpha}\in\mathcal{A}} \|\boldsymbol{\alpha}\|_1$. Using multi-indices, the Macaulay matrix[7] $\boldsymbol{M}_{\boldsymbol{r}}^{pq_1\times q_2}$ of degree $d_t \geq d$ associated with $r(\boldsymbol{\sigma})$ is then defined block-elementwise as:

$$(\boldsymbol{M}_r)_{\boldsymbol{\rho},\boldsymbol{\kappa}} = \boldsymbol{v}_{\boldsymbol{\kappa}-\boldsymbol{\rho}}, \text{ where } \boldsymbol{\kappa}, \boldsymbol{\rho} \in \mathbb{N}_0^m; \|\boldsymbol{\kappa}\|_1 \leq d_t, \|\boldsymbol{\rho}\|_1 \leq d_t - d.$$

Here, whenever $\boldsymbol{\kappa} - \boldsymbol{\rho} \notin \mathcal{A}$, the coefficient $\boldsymbol{v}_{\boldsymbol{\kappa}-\boldsymbol{\rho}}$ is assumed to be zero. △

Example 4.2.3. Reconsider the GCH equations constructed in Example 4.2.2. The associated Macaulay matrix of degree three can then be used to define

[7] When the difference equations within $\boldsymbol{r}(\boldsymbol{\sigma})$ are of different degree, some additional rows are to be considered. Though this is not the case in the current chapter.

model-compliant sequences $\hat{\boldsymbol{y}}$:

$$\mathbf{0} = \begin{matrix} \\ \boldsymbol{r}(\boldsymbol{\sigma}) \\ \\ \\ \\ \\ \\ \sigma_1\boldsymbol{r}(\boldsymbol{\sigma}) \\ \\ \\ \sigma_2\boldsymbol{r}(\boldsymbol{\sigma}) \\ \\ \end{matrix} \begin{bmatrix} \overset{(0,0)}{b} & \overset{(1,0)}{a} & \overset{(0,1)}{} & \overset{(2,0)}{1} & \overset{(1,1)}{} & \overset{(0,2)}{} & \overset{(3,0)}{} & \overset{(2,1)}{} & \overset{(1,2)}{} & \overset{(3,0)}{} \\ d & c & a & & 2 & & & & & \\ e & & c & & & 1 & & & & \\ & b & & a & & & 1 & & & \\ & d & & c & a & & & 2 & & \\ & e & & & c & & & & 1 & \\ & & b & & a & & & 1 & & \\ & & d & & c & a & & & 2 & \\ & & e & & & c & & & & 1 \end{bmatrix} \begin{bmatrix} y_{(0,0)} \\ y_{(1,0)} \\ y_{(0,1)} \\ y_{(2,0)} \\ y_{(1,1)} \\ y_{(0,2)} \\ y_{(3,0)} \\ y_{(2,1)} \\ y_{(1,2)} \\ y_{(0,3)} \end{bmatrix} = \boldsymbol{M}_{\boldsymbol{r}}\hat{\boldsymbol{y}}$$

Note that for these GCH equations, $\boldsymbol{M}_{\boldsymbol{r}}$ depends on the model parameters $\boldsymbol{\Gamma}$. $\diamond$

While different sets of grid points can be considered, we confine ourselves to those up to some fixed degree: $\mathcal{K} = \{\boldsymbol{\kappa} \mid \boldsymbol{\kappa} \in \mathbb{N}_0^m, \|\boldsymbol{\kappa}\|_1 \leq d\}$, as in Definition 4.2.2. Given such a set of grid points of a large enough degree, the null space of a Macaulay matrix corresponds precisely to an observability matrix, formalized in Theorem 4.2.2 below. Indeed, the kernel of $\boldsymbol{M}_{\boldsymbol{r}}$ describes the model-compliant sequences $\hat{\boldsymbol{y}}$, just like range $\boldsymbol{\Gamma}$ in the image representation. For too small a degree, range $\boldsymbol{\Gamma}$ may be a strict subspace of null $\boldsymbol{M}_{\boldsymbol{r}}$. For example, if signals were considered over $\|\boldsymbol{\kappa}\|_1 \leq 2$ in Example 4.2.3, the Macaulay matrix would be of size 3×6 so that its nullity would be at least three, which is larger than the model order $n = 2$ in this case.

Lemma 4.2.2 ([12], Thm. 4.1)**.** *Let $\boldsymbol{r}(\boldsymbol{\sigma}) \in \mathbb{C}^p[\boldsymbol{\sigma}]$ be a set of mD difference equations, with a finite number n of common roots, counting multiplicities, and without roots at infinity and let $\boldsymbol{M}_{\boldsymbol{r}}$ be the corresponding Macaulay matrix of degree d, as per Definition 4.2.2. Let $\boldsymbol{\Gamma}_{\boldsymbol{\kappa},:} = \boldsymbol{C}\boldsymbol{A}^{\boldsymbol{\kappa}}, \|\boldsymbol{\kappa}\|_1 \leq d$ be the observability matrix associated with any equivalent nth order state-space model* (4.1)*. Then for some finite degree of regularity d^*, it holds that: when $d \geq d^*$, then*

$$\operatorname{null} \boldsymbol{M}_{\boldsymbol{r}} = \operatorname{range} \boldsymbol{\Gamma}, \quad \textit{and} \quad \dim(\operatorname{range} \boldsymbol{\Gamma}) = n.$$

Moreover, $d^* \leq \min_{\mathcal{I} \subseteq \{1,\ldots,p\}, \#\mathcal{I}=m} \left(\sum_{i \in \mathcal{I}} d_i - m + 1\right)$, where d_i is the total degree of the ith element of $\boldsymbol{r}(\boldsymbol{\sigma})$.

4.3 Polynomial characterization of the stationary points

We now derive the polynomial description for the local minima of the least-squares misfit problem (4.2) through a relaxation of the problem. First, a characterization based on the image representation is considered in Section 4.3.1. This characterization implies a decomposition of the ambient space for each optimal model, which is detailed in Section 4.3.2. From this decomposition, another characterization based on the kernel representation is derived in Section 4.3.3.

4.3.1 Image representation

Consider the following relaxed version of (4.2), where the model parameters $(\boldsymbol{\Lambda}, \boldsymbol{x}) \notin \mathcal{R}$ are allowed as well. Using the insights from Section 4.2, the model-compliancy constraint is expressed simultaneously for all grid points through an observability matrix:

$$\min_{\boldsymbol{\Lambda}, \boldsymbol{x}} \|\tilde{\boldsymbol{y}}\|_2^2 \tag{4.4}$$

$$\text{subject to } \hat{\boldsymbol{y}} = \boldsymbol{y} - \tilde{\boldsymbol{y}}$$

$$\hat{\boldsymbol{y}} = \boldsymbol{\Gamma}\boldsymbol{x}.$$

Substituting in the constraints leads to the unconstrained objective function (4.5):

$$\|\tilde{\boldsymbol{y}}\|_2^2 = \|\boldsymbol{y} - \boldsymbol{\Gamma}\boldsymbol{x}\|^2 = \boldsymbol{y}^{\mathsf{H}}\boldsymbol{y} + \boldsymbol{x}^{\mathsf{H}}\boldsymbol{\Gamma}^{\mathsf{H}}\boldsymbol{\Gamma}\boldsymbol{x} - \boldsymbol{x}^{\mathsf{H}}\boldsymbol{\Gamma}^{\mathsf{H}}\boldsymbol{y} - \boldsymbol{y}^{\mathsf{H}}\boldsymbol{\Gamma}\boldsymbol{x}. \tag{4.5}$$

As this objective is generally a multivariate polynomial of degree larger than two, it is non-convex and can thus have multiple local minima.

Since we are interested in real models after all, we focus on the stationary points of this relaxed problem that do in fact define real models. Indeed, it is obvious that any stationary point of (4.4) that lies in $\mathcal{R}$ is also a stationary point of (4.2), since, if the derivatives vanish in all directions, they also vanish in the directions that respect the realness constraint. However, the converse is not immediately obvious: if the derivatives vanish in a restricted set of 'real' directions, they do not necessarily vanish in all directions. As such, it has yet to be established that all local minima of (4.2) are also stationary points of (4.4). Theorem 4.4.1 later asserts that this is indeed the case.

The FONC for this relaxed problem can now be derived by equating all partial derivatives of the objective function (4.5) w.r.t. the decision variables to zero. Wirtinger calculus provides a straightforward methodology to deal with the intricacies of optimization problems over complex-valued decision variables [2]: Treat the complex conjugates of the decision variables, i.e., $\overline{\boldsymbol{\Lambda}}, \overline{\boldsymbol{x}}$, as independent w.r.t. $\boldsymbol{\Lambda}, \boldsymbol{x}$. This results in:

$$\frac{\partial \|\tilde{\boldsymbol{y}}\|_2^2}{\partial \boldsymbol{x}} = (\boldsymbol{x}^{\mathsf{H}}\boldsymbol{\Gamma}^{\mathsf{H}} - \boldsymbol{y}^{\mathsf{H}})\boldsymbol{\Gamma} = \boldsymbol{0}, \tag{4.6a}$$

$$\frac{\partial \|\tilde{\boldsymbol{y}}\|_2^2}{\partial \lambda_{i,k}} = (\boldsymbol{x}^{\mathsf{H}}\boldsymbol{\Gamma}^{\mathsf{H}} - \boldsymbol{y}^{\mathsf{H}})\frac{\partial \boldsymbol{\Gamma}}{\partial \lambda_{i,k}}\boldsymbol{x} = 0 \qquad \forall (i,k) \in \{1,\dots,n\} \times \{1,\dots,m\}.$$

We denote the set of stationary points, that is, the common solutions of (4.6), as $\mathcal{S}$ from here on.

Note that $\frac{\partial \boldsymbol{\Gamma}}{\partial (\lambda_i)_k}\boldsymbol{x} = \begin{bmatrix} \frac{\partial \boldsymbol{\lambda}_i^{\boldsymbol{\kappa}_1}}{\partial \lambda_{i,k}} & \dots & \frac{\partial \boldsymbol{\lambda}_i^{\boldsymbol{\kappa}_N}}{\partial \lambda_{i,k}} \end{bmatrix}^{\mathsf{T}} x_i$. To more compactly express the following equations, we introduce the following $N \times nm$ polynomial matrix, which stacks these derivatives along one another:

$$\boldsymbol{\Gamma}^{(\boldsymbol{\lambda})} = \begin{bmatrix} \kappa_1 \boldsymbol{\lambda}_1^{\boldsymbol{\kappa}_1 - \boldsymbol{e}_1} & \dots & \kappa_m \boldsymbol{\lambda}_1^{\boldsymbol{\kappa}_m - \boldsymbol{e}_m} & \dots & \kappa_1 \boldsymbol{\lambda}_n^{\boldsymbol{\kappa}_1 - \boldsymbol{e}_1} & \dots & \kappa_m \boldsymbol{\lambda}_n^{\boldsymbol{\kappa}_m - \boldsymbol{e}_m} \\ \vdots & & \vdots & & \vdots & & \vdots \\ \kappa_1 \boldsymbol{\lambda}_1^{\boldsymbol{\kappa}_N - \boldsymbol{e}_1} & \dots & \kappa_m \boldsymbol{\lambda}_1^{\boldsymbol{\kappa}_N - \boldsymbol{e}_m} & \dots & \kappa_1 \boldsymbol{\lambda}_n^{\boldsymbol{\kappa}_N - \boldsymbol{e}_1} & \dots & \kappa_m \boldsymbol{\lambda}_n^{\boldsymbol{\kappa}_N - \boldsymbol{e}_m} \end{bmatrix}.$$

Now, for any model $(\boldsymbol{\Lambda}, \boldsymbol{x}) \in \mathcal{R}$, it holds that $\boldsymbol{\Lambda}\boldsymbol{x} = \hat{\boldsymbol{y}} \in \mathbb{R}^N$, and consequently, $(\boldsymbol{\Lambda}\boldsymbol{x})^{\mathsf{H}} = (\boldsymbol{\Lambda}\boldsymbol{x})^{\mathsf{T}}$. As such, the subset of stationary points that define real models, i.e., $\mathcal{S} \cap \mathcal{R}$, satisfies the following polynomial equations, obtained by exchanging $^{\mathsf{H}}$ for $^{\mathsf{T}}$ in (4.6), where $\otimes$ denotes the Kronecker product:

$$\boldsymbol{0} = (\boldsymbol{x}^{\mathsf{T}}\boldsymbol{\Gamma}^{\mathsf{T}} - \boldsymbol{y}^{\mathsf{T}})\boldsymbol{\Gamma}, \tag{4.7a}$$

$$\boldsymbol{0} = (\boldsymbol{x}^{\mathsf{T}}\boldsymbol{\Gamma}^{\mathsf{T}} - \boldsymbol{y}^{\mathsf{T}})\boldsymbol{\Gamma}^{(\boldsymbol{\lambda})}(\operatorname{diag}(\boldsymbol{x}) \otimes \boldsymbol{I}_m). \tag{4.7b}$$

For general 'complex' models, it is clear that this substitution of $^{\mathsf{H}}$ by $^{\mathsf{T}}$ does not retain the equality. Because of this, the solutions of (4.7) outside of $\mathcal{R}$ do not necessarily belong to $\mathcal{S}$.

Finally, if one is only interested in the local minima of the misfit problem (4.2), one can apply Theorem 4.4.2, which states that all x_i are nonzero for these minima. This ensures $(\operatorname{diag}(\boldsymbol{x}) \otimes \boldsymbol{I}_m)$ is invertible, allowing us to simplify (4.7b) by right multiplying with the inverse of this matrix. The preceding observations then result in the simplified characterization of the local minima (4.8) and Theorem 4.3.1 below.

Theorem 4.3.1. *If* $(\boldsymbol{\Lambda}, \boldsymbol{x}) \in \mathcal{S}$ *is such that* $\forall i : x_i \neq 0$ *and* $(\boldsymbol{\Lambda}, \boldsymbol{x}) \in \mathcal{R}$*, then it is a solution to:*

$$\begin{bmatrix} \boldsymbol{\Gamma}^{\mathsf{T}}\boldsymbol{\Gamma} & \boldsymbol{\Gamma}^{\mathsf{T}}\boldsymbol{y} \\ \left(\boldsymbol{\Gamma}^{(\boldsymbol{\lambda})}\right)^{\mathsf{T}}\boldsymbol{\Gamma} & \left(\boldsymbol{\Gamma}^{(\boldsymbol{\lambda})}\right)^{\mathsf{T}}\boldsymbol{y} \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ -1 \end{bmatrix} = \boldsymbol{0}. \tag{4.8}$$

We denote the set of common roots of (4.8) *as* $\mathcal{C}$*. Conversely,* $(\mathcal{C} \cap \mathcal{R}) \subseteq \mathcal{S}$*.*

Proof. The first statement was derived in the text leading up to this theorem. To prove the second part of the statement, the steps taken in the derivation leading up to (4.8) can be reverted. □

By Theorem 4.3.1, some stationary points with $x_i = 0$ might not arise as solutions to (4.8). In such models, the ith eigentuple does not contribute to $\hat{\boldsymbol{y}}$, so that it may just as well be expressed with a model of order $n-1$. However, such *lower-order* solutions may persist in $\mathcal{C}$. Indeed, take $\boldsymbol{y} \perp \text{range}\begin{bmatrix}\boldsymbol{\Gamma} & \boldsymbol{\Gamma}^{(\boldsymbol{\lambda})}\end{bmatrix}$ for some fixed $\boldsymbol{\Lambda}$. The model $(\boldsymbol{\Lambda}, \boldsymbol{0})$ is then a solution to (4.8).

Let us now consider a first-order, 2D example below to illustrate the presented methodology.

Example 4.3.1. We consider the misfit problem (4.2) for fitting a first-order model, i.e., $n = 1$, to the following 2D data sequence:

$$\boldsymbol{y}^{\mathsf{T}} = \begin{bmatrix} \overset{(0,0)}{0.7060} & \overset{(1,0)}{0.0318} & \overset{(0,1)}{0.2769} & \overset{(2,0)}{0.0462} & \overset{(1,1)}{0.0971} & \overset{(0,2)}{0.8235} \end{bmatrix}.$$

We now set up the equations (4.8) for this problem. There is one eigentuple, the components of which we denote as $\boldsymbol{\lambda}_1 = (\alpha, \beta)$ for this example, to avoid too many super- and subscripts, resulting in:

$$\begin{aligned} 0 &= (1 + \beta^2 + \beta^4 + \alpha^2 + \alpha^2\,\beta^2 + \alpha^4)x \\ &\quad - (0.70 + 0.28\,\beta + 0.82\,\beta^2 + 0.03\,\alpha + 0.097\,\alpha\beta + 0.046\,\alpha^2), \\ 0 &= (\alpha + \alpha\,\beta^2 + 2\,\alpha^3)x - (0.0318 + 0.0971\,\beta + 0.0924\,\alpha), \\ 0 &= (\beta + 2\,\beta^3 + \alpha^2\,\beta)x - (0.2769 + 1.6470\,\beta + 0.0971\,\alpha). \end{aligned} \tag{4.9}$$

Solving this using MacaulayLab [188] results in 13 affine complex solutions for $\boldsymbol{\lambda}$, as shown in Tables 4.1 and 4.2. By Theorem 4.3.1, the three real-valued solutions in Table 4.1 are then stationary points of the relaxed problem (4.4), and by extension of (4.2). Note that existence of multiple stationary points

index	$\boldsymbol{\lambda}_1 = (\alpha, \beta)$	x	$\|\boldsymbol{y} - \hat{\boldsymbol{y}}\|_2$
1	$(0.1268, 1.1478)$	0.5201	0.3990
2	$(0.0027, -0.3106)$	0.6325	0.9074
3	$(-87.8665, 5.1478)$	$5.5823 \cdot 10^{-6}$	1.1243

Table 4.1: Solutions of (4.9) that correspond to real models, ordered by misfit norm. Since they are real models, these are stationary points of (4.4), that is, they satisfy (4.6). By extension, they are stationary points of (4.2).

index	$\boldsymbol{\lambda}_1 = (\alpha, \beta)$	x	$\|\boldsymbol{f}(\boldsymbol{\lambda}_1, \boldsymbol{x})\|_2$
4-5	$(-0.1579 \mp 1.8369i, -2.4644 \pm 0.4170i)$	$0.1404 \pm 0.0581i$	7.6877
6-7	$(-0.0540 \mp 1.0321i, 1.1074 \mp 0.0214i)$	$0.7924 \mp 0.0413i$	0.5863
8-9	$(-0.1372 \mp 0.1135i, -1.4475 \mp 0.8220i)$	$0.1648 \mp 0.1835i$	3.1840
10-11	$(0.2252 \mp 1.1657i, -1.4219 \mp 0.3115i)$	$0.3782 \mp 0.0993i$	5.0424
12-13	$(-0.0168 \mp 0.6927i, -0.2462 \pm 0.0686i)$	$0.8521 \pm 0.0038i$	1.3986

Table 4.2: Complex-valued solutions of (4.9), where $\boldsymbol{f}(\boldsymbol{\lambda}_1, \boldsymbol{x})$ is the left-hand-side of (4.6). This function evaluation shows that these solutions are not part of $\mathcal{S}$.

indicates the nonconvexity of the problem. From this, we conclude that the globally optimal model is corresponds to the eigentuple $\boldsymbol{\lambda} = (0.1268, 1.1478)$.

The remaining ten solutions in Table 4.2 do not define real models. As such, these models are not necessarily stationary points. This is evidenced by the last column of the table, where the norm of the FONC evaluated in these points is given.

Notice how all the solutions satisfy $x \neq 0$. However, if we turn our attention to (4.7) instead and search for solutions with $x = 0$, these equations reduce to:

$$0 = y^{\mathsf{T}}\boldsymbol{\Gamma} = 0.7060 + 0.0318\alpha + 0.2769\beta + 0.0462\alpha^2 + 0.0971\alpha\beta + 0.8235\beta^2.$$

Clearly, there is an infinite set of solutions to this bivariate polynomial equation, which gets eliminated in the simplified characterization (4.9). In Section 4.5, we briefly discuss how such infinite solution sets may hinder the convergence of some polynomial solvers. Note that this observation can be easily generalized to higher values of n. ◇

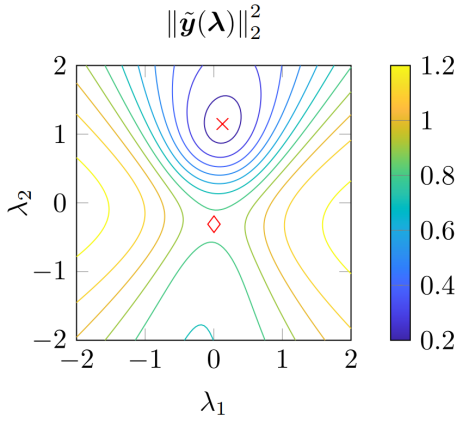


Figure 4.2: Objective for the misfit problem in Example 4.3.1. The optimal value of x is determined based on $\boldsymbol{\lambda}$ through a linear projection of $\boldsymbol{y}$ onto range $\boldsymbol{\Gamma}$. The global minimum is marked by $\times$ and the saddle point by $\diamond$.

4.3.2 Misfit structure

As previously discussed, range $\boldsymbol{\Gamma} \subseteq \mathbb{R}^N$ forms the linear subspace of model-compliant sequences over $\mathcal{K}$. For a fixed value of $\boldsymbol{\Lambda}$, the closest sequence $\hat{\boldsymbol{y}}$ in a least-squares sense is thus readily obtained by orthogonally projecting $\boldsymbol{y}$ onto this subspace, defining $\boldsymbol{x}$ in terms of $\boldsymbol{\Lambda}$ in the process. This is reflected in (4.6a), which constitute the normal equations for this projection. Using this projection to define x in terms of $\boldsymbol{\lambda}$, a contour plot of the objective function underlying Example 4.3.1 is plotted in Figure 4.2, clearly indicating the non-convexity of the optimization problem.

In this sense, $\boldsymbol{\Lambda}$ are the only decision variables of importance for optimality, while $\boldsymbol{x}$, on which the objective only depends linearly, can be considered auxiliary variables. Each value of $\boldsymbol{\Lambda}$ thus decomposes the ambient space $\mathbb{R}^N$ into two orthogonal parts: a model-compliant subspace, range $\boldsymbol{\Gamma}$, and its orthogonal complement, containing the residuals. In case of distinct eigentuples and an appropriate set of grid points, this complementary space is row $\boldsymbol{M_r}$ by Lemmas 4.2.1 and 4.2.2, such that $\tilde{\boldsymbol{y}} \in \operatorname{row} \boldsymbol{M_r}$.

To also characterize the local optimality of $\boldsymbol{\Lambda}$, the whole of (4.8) should be considered, which can be rewritten as:

$$\tilde{\boldsymbol{y}}^\mathsf{T} \begin{bmatrix} \boldsymbol{\Gamma} & \boldsymbol{\Gamma}^{(\boldsymbol{\lambda})} \end{bmatrix} = \tilde{\boldsymbol{y}}^\mathsf{T} \boldsymbol{G} = \boldsymbol{0}. \tag{4.10}$$

For real models, this condition further restricts the misfit $\tilde{\boldsymbol{y}}$ to lie in the orthogonal complement of range $\boldsymbol{G}$, a strict subspace of $(\operatorname{range} \boldsymbol{\Gamma})^\perp$ by its

construction. Similarly to the 1D case [36], each optimal model $\boldsymbol{\Lambda}$ then gives rise to a further decomposition of $\mathbb{R}^N$ into three parts: the model-compliant space range $\boldsymbol{\Gamma}$, the misfit space $(\text{range}\,\boldsymbol{G})^\perp$, and the complementary subspace range $\boldsymbol{\Gamma}^{(\boldsymbol{\lambda})}$.

Finally, define the polynomial $\tilde{y}(\boldsymbol{\lambda}) = \sum_{\boldsymbol{\kappa}\in\mathcal{K}} \tilde{y}_{\boldsymbol{\kappa}} \boldsymbol{\lambda}^{\boldsymbol{\kappa}}$. Equation (4.10) can then be re-expressed as interpolation condition $\forall i, k : \tilde{y}(\boldsymbol{\lambda}_i) = 0 = \frac{\partial \tilde{y}(\boldsymbol{\lambda})}{\partial \lambda_k}|_{\boldsymbol{\lambda}=\boldsymbol{\lambda}_i}$ [36, 157], or, in a system-theoretic sense, the signals $\boldsymbol{\lambda}^{\boldsymbol{\kappa}}, \kappa_k \boldsymbol{\lambda}^{\boldsymbol{\kappa}-\boldsymbol{e}_k}$ satisfy $\tilde{y}(\boldsymbol{\sigma})$. Indeed, $\tilde{\boldsymbol{y}}$ lies in the orthogonal complement of the observability matrix $\boldsymbol{G}$ associated with these signals. By the discussion in Section 4.2, this complement is spanned by the rows of a Macaulay matrix generated by a set of polynomials with this specific set of common roots and multiplicity structure, given a suitable $\mathcal{K}$.

4.3.3 Kernel representation

The Macaulay matrix whose row space forms the orthogonal complement of the observability matrix $\boldsymbol{G}$ can be used to characterize $\hat{\boldsymbol{y}}$ through the kernel representation, leading to an alternative set of polynomial equations which are satisfied by the minima of (4.2). Let us now look into how one can obtain difference equations with the required multiplicity structure, that is, polynomials for which all first-order partial derivatives vanish in the eigentuples. To this end, consider Example 4.3.2.

Example 4.3.2. To illustrate the procedure, let us constrain ourselves to a simple 2D, first-order example. First, consider a set of difference equations describing precisely one exponential mode $\boldsymbol{\lambda}_1^{\boldsymbol{\kappa}} = (\alpha, \beta)^{\boldsymbol{\kappa}}$, which is readily constructed as:

$$\boldsymbol{r}(\boldsymbol{\sigma}) y_{\boldsymbol{\kappa}} = \begin{bmatrix} r_1(\boldsymbol{\sigma}) \\ r_2(\boldsymbol{\sigma}) \end{bmatrix} y_{\boldsymbol{\kappa}} = \begin{bmatrix} \sigma_1 - \alpha \\ \sigma_2 - \beta \end{bmatrix} y_{\boldsymbol{\kappa}} = \boldsymbol{0}.$$

Now consider the difference equations resulting from all pairwise products of the elements of this $\boldsymbol{r}(\boldsymbol{\sigma})$:

$$\boldsymbol{r}^2(\boldsymbol{\sigma}) y_{\boldsymbol{\kappa}} = \begin{bmatrix} r_1(\boldsymbol{\sigma})^2 \\ r_1(\boldsymbol{\sigma}) r_2(\boldsymbol{\sigma}) \\ r_2(\boldsymbol{\sigma})^2 \end{bmatrix} y_{\boldsymbol{\kappa}} = \begin{bmatrix} \alpha^2 - 2\alpha\sigma_1 + \sigma_1^2 \\ \alpha\beta - \alpha\sigma_2 - \beta\sigma_1 + \sigma_1\sigma_2 \\ \beta^2 - 2\beta\sigma_2 + \sigma_2^2 \end{bmatrix} y_{\boldsymbol{\kappa}} = \boldsymbol{0}.$$

Clearly, $\boldsymbol{r}^2$ then has the same solution $\boldsymbol{\lambda}_1$, though with an altered multiplicity. Indeed, the difference equations $\boldsymbol{r}^2(\boldsymbol{\sigma})$ precisely describe the signal space spanned by $\boldsymbol{\lambda}_1^{\boldsymbol{\kappa}}, \kappa_1 \boldsymbol{\lambda}_1^{\boldsymbol{\kappa}-\boldsymbol{e}_1}$ and $\kappa_2 \boldsymbol{\lambda}_1^{\boldsymbol{\kappa}-\boldsymbol{e}_2}$.

Let us verify this using the Macaulay framework: For $\|\boldsymbol{\kappa}\|_2 \leq 2$, the null space of the associated Macaulay matrix constitutes signals in the range of the

observability matrix $\boldsymbol{G}$:

$$\boldsymbol{M}_{\boldsymbol{r}^2}\boldsymbol{G} = \begin{bmatrix} \alpha^2 & -2\,\alpha & 0 & 1 & 0 & 0 \\ \alpha\,\beta & -\beta & -\alpha & 0 & 1 & 0 \\ \beta^2 & 0 & -2\,\beta & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ \alpha & 1 & 0 \\ \beta & 0 & 1 \\ \alpha^2 & 2\,\alpha & 0 \\ \alpha\,\beta & \beta & \alpha \\ \beta^2 & 0 & 2\,\beta \end{bmatrix} = \boldsymbol{0},$$

Clearly, $\boldsymbol{M}_{\boldsymbol{r}^2}$ is of rank 3, such that this $\boldsymbol{G}$ spans its full null space, indicating no further multiplicities. It can be verified that a similar observation holds for higher degree Macaulay matrices.

Now, since $\operatorname{range}\boldsymbol{\Gamma} \subset \operatorname{range}\boldsymbol{G}$, the converse holds for their orthogonal complements: $\operatorname{row}\boldsymbol{M}_{\boldsymbol{r}} \supset \operatorname{row}\boldsymbol{M}_{\boldsymbol{r}^2}$. This indicates that $\boldsymbol{M}_{\boldsymbol{r}^2}$ can be expressed as linear combinations of the rows of $\boldsymbol{M}_{\boldsymbol{r}}$. Indeed, notice that the polynomial multiplications can be expressed using Macaulay matrices $\boldsymbol{P}_{r_1}, \boldsymbol{P}_{r_2}$ matching the degree of the row shifts in $\boldsymbol{M}_{\boldsymbol{r}}$, i.e., one in this case:

$$\boldsymbol{P}_{r_1} = \begin{bmatrix} -\alpha & 1 & 0 \end{bmatrix}, \boldsymbol{P}_{r_2} = \begin{bmatrix} -\beta & 0 & 1 \end{bmatrix},$$

$$\boldsymbol{M}_{\boldsymbol{r}^2} = \begin{bmatrix} \boldsymbol{P}_{r_1} & \\ \boldsymbol{P}_{r_2} & \\ & \boldsymbol{P}_{r_2} \end{bmatrix} \boldsymbol{M}_{\boldsymbol{r}} = \boldsymbol{P}\boldsymbol{M}_{\boldsymbol{r}}$$

$$= \left[\begin{array}{ccc|ccc} -\alpha & 1 & 0 & 0 & 0 & 0 \\ -\beta & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\beta & 0 & 1 \end{array}\right] \begin{bmatrix} -\alpha & 1 & 0 & 0 & 0 & 0 \\ 0 & -\alpha & 0 & 1 & 0 & 0 \\ 0 & 0 & -\alpha & 0 & 1 & 0 \\ -\beta & 0 & 1 & 0 & 0 & 0 \\ 0 & -\beta & 0 & 0 & 1 & 0 \\ 0 & 0 & -\beta & 0 & 0 & 1 \end{bmatrix}.$$

◇

We now generalize the observations of Example 4.3.2 to general multidimensional difference equations.

Lemma 4.3.1. *Let $\boldsymbol{r}(\boldsymbol{\sigma}) \in \mathbb{R}^p[\boldsymbol{\sigma}], p \geq m$, be a set of difference equations with a kernel that is spanned by n basis signals $\mathcal{B} = \{\boldsymbol{\lambda}_1^{\boldsymbol{\kappa}}, \ldots, \boldsymbol{\lambda}_n^{\boldsymbol{\kappa}}\}$. Define the set of polynomials $\boldsymbol{r}^2(\boldsymbol{\sigma}) = \left[r_s(\boldsymbol{x})r_t(\boldsymbol{x}) \mid s = 1, \ldots, n; t \leq s\right]$. The kernel of $\boldsymbol{r}^2(\boldsymbol{\sigma})$ is then spanned by the basis $\mathcal{B} \cup \{\kappa_k \boldsymbol{\lambda}_i^{\boldsymbol{\kappa}-\boldsymbol{e}_k} \mid i = 1, \ldots, n; k = 1, \ldots, m\}$.*

Proof. Since all signals in the kernel are of the form $\boldsymbol{\lambda}_i^{\boldsymbol{\kappa}}$, the common roots $\boldsymbol{\lambda}_i$ are multiplicity-free. This means that $\boldsymbol{r}(\boldsymbol{\lambda}_i) = 0$ and $\forall i, \boldsymbol{v} \in \mathbb{C}^m \backslash \{\boldsymbol{0}\} : \boldsymbol{J}_{\boldsymbol{r}}(\boldsymbol{\lambda}_i)\boldsymbol{v} \neq \boldsymbol{0}$, where $\boldsymbol{J}_{\boldsymbol{r}}$ is the Jacobian of $\boldsymbol{r}(\boldsymbol{\lambda})$ [34, 172].

Since $\boldsymbol{r}(\boldsymbol{\lambda}_i) = \mathbf{0}$, $\frac{\partial r_s r_t}{\partial \lambda_k}|_{\boldsymbol{\lambda}=\boldsymbol{\lambda}_i} = \frac{\partial r_s}{\partial \lambda_k} r_t(\boldsymbol{\lambda}_i) + \frac{\partial r_t}{\partial \lambda_k} r_s(\boldsymbol{\lambda}_i) = 0$ for any pair (s,t), so that $\kappa_k \boldsymbol{\lambda}_i^{\boldsymbol{\kappa}-\boldsymbol{e}_k}$ satisfies $\boldsymbol{r}^2(\boldsymbol{\sigma})$.

Finally, due to the closedness property of the dual space, to show the absence of all higher-order multiplicities it suffices to show that nontrivial linear combinations of second-order partial derivatives vanish in $\boldsymbol{\lambda}_i$ [169]. It can be worked out that: $\sum_k^m \sum_{l\geq k}^m h_{(k,l)} \frac{\partial^2 r_s r_t}{\partial \lambda_k \partial \lambda_l}(\boldsymbol{\lambda}_i) = (\nabla r_s(\boldsymbol{\lambda}_i))^{\mathsf{T}} \boldsymbol{H} (\nabla r_t(\boldsymbol{\lambda}_i))$, where $\boldsymbol{H}$ is a symmetric matrix containing the $h_{(k,l)}$. To vanish for all pairs (s,t), it should thus hold that $\boldsymbol{J_r}(\boldsymbol{\lambda}_i) \boldsymbol{H} \boldsymbol{J_r}(\boldsymbol{\lambda}_i)^{\mathsf{T}} = \mathbf{0}$. However, since $\boldsymbol{H}$ is nonzero and $\boldsymbol{J_r}(\boldsymbol{\lambda}_i)$ is of full column rank, this cannot be the case. □

Corollary 4.3.1. *Let $\boldsymbol{r}(\boldsymbol{\sigma})$ be the parametric m GCH equations of order n constructed from the eigentuples $\boldsymbol{\Lambda}$ and let $\boldsymbol{r}^2(\boldsymbol{\sigma})$ be as defined in Lemma 4.3.1. Whenever $\boldsymbol{\Lambda}$ comprises n distinct eigentuples, the right null space of the degree $d > (2n-1)m$ Macaulay matrix $\boldsymbol{M}_{\boldsymbol{r}^2}$ is then* $\operatorname{range} \begin{bmatrix} \boldsymbol{\Gamma} & \boldsymbol{\Gamma}^{(\boldsymbol{\lambda})} \end{bmatrix} = \operatorname{range} \boldsymbol{G}$.

Proof. By Lemma 4.2.1, these equations meet the conditions of Lemma 4.3.1 when the eigentuples are distinct. As discussed in Section 4.2.2, signals of the form $\kappa_k \boldsymbol{\lambda}_i^{\boldsymbol{\kappa}-\boldsymbol{e}_k}$ are then model-compliant for all i and k, so that the statement follows from Lemma 4.2.2. Note that the observability matrix $\boldsymbol{G}$ is to be constructed as in Lemma 4.2.2 with $\mathcal{K} = \{\boldsymbol{\kappa} \in \mathbb{N}_0 \mid \|\boldsymbol{\kappa}\|_1 \leq d\}$. The degree bound follows from the upper bound on the degree of regularity of $\boldsymbol{r}^2(\boldsymbol{\sigma})$, defined in Theorem 4.2.2. □

Note that, as was already illustrated in Example 4.3.1, the row space of $\boldsymbol{M}_{\boldsymbol{r}^2}$ is a subspace of row $\boldsymbol{M}_{\boldsymbol{r}}$. Indeed, it is readily verified that the multiplication property holds in general: Given a Macaulay matrix $\boldsymbol{M}_{r_t}$ of degree d, with r_t of degree d_t and let $\boldsymbol{M}_{r_s}$ be another Macaulay matrix of degree $d - d_t$, then it holds that $\boldsymbol{M}_{r_s r_t} = \boldsymbol{M}_{r_s} \boldsymbol{M}_{r_t}$ is the Macaulay matrix of degree d associated with $r_s(\boldsymbol{\sigma}) r_t(\boldsymbol{\sigma})$. From this relation it can be worked out that $\boldsymbol{M}_{\boldsymbol{r}^2} = \boldsymbol{P} \boldsymbol{M}_{\boldsymbol{r}} = \operatorname{blockdiag}(\boldsymbol{M}_{\boldsymbol{r}_{1:p}}, \boldsymbol{M}_{\boldsymbol{r}_{2:p}}, \ldots, \boldsymbol{M}_{r_p}) \boldsymbol{M}_{\boldsymbol{r}}$ for an appropriate choice of degrees of the Macaulay matrices in $\boldsymbol{P} \in \mathbb{C}^{q \times t}$.

Lemma 4.3.1 provides us with an explicit expression for the left kernel of $\boldsymbol{G} = \begin{bmatrix} \boldsymbol{\Gamma} & \boldsymbol{\Gamma}^{(\boldsymbol{\lambda})} \end{bmatrix}$ in terms of the eigentuples given a suitable $\mathcal{K}$. This allows (4.10) to be expressed as $\tilde{\boldsymbol{y}} \in \operatorname{row} \boldsymbol{M}_{\boldsymbol{r}^2}$. Using this property, Theorem 4.3.1 is rephrased by leveraging the kernel representation $\boldsymbol{M}_{\boldsymbol{r}} \hat{\boldsymbol{y}} = \mathbf{0}$, similarly to the 1D misfit and model-order-reduction problems [102, 104].

Theorem 4.3.2. *Let $\mathcal{K}$ be as defined in Corollary 4.3.1 and let $(\boldsymbol{\Lambda}^*, \boldsymbol{x}^*) \in \mathcal{C}$ be any model with distinct eigentuples.*

Then, there exists a vector $\boldsymbol{g}^* \in \mathbb{C}^q$ *such that* $(\boldsymbol{\Lambda}^*, \boldsymbol{g}^*)$ *satisfies:*

$$\begin{bmatrix} \boldsymbol{M_r} \boldsymbol{M_r}^\mathsf{T} \boldsymbol{P}^\mathsf{T} & \boldsymbol{M_r} \boldsymbol{y} \end{bmatrix} \begin{bmatrix} \boldsymbol{g} \\ -1 \end{bmatrix} = \boldsymbol{0}. \tag{4.11}$$

Conversely, for any point $(\boldsymbol{\Lambda}^*, \boldsymbol{g}^*)$ *with distinct eigentuples satisfying* (4.11)*, there exists a vector* $\boldsymbol{x}^*$ *such that* $(\boldsymbol{\Lambda}^*, \boldsymbol{x}^*) \in \mathcal{C}$*.*

Proof. Both sides of the proof rely on $\operatorname{null} \boldsymbol{M_r} = \operatorname{range} \boldsymbol{\Gamma}$, which holds by Theorems 4.2.1 and 4.2.2 .

We first prove that any solution of (4.8) with distinct eigentuples satisfies (4.11). Note that (4.10) follows from (4.8). By Corollary 4.3.1 the former implies that $\exists \boldsymbol{g} : \tilde{\boldsymbol{y}} = \boldsymbol{M_r}^\mathsf{T} \boldsymbol{P}^\mathsf{T} \boldsymbol{g}$. This results in:

$$\boldsymbol{y} = \hat{\boldsymbol{y}} + \tilde{\boldsymbol{y}} = \boldsymbol{\Gamma} \boldsymbol{x} + \boldsymbol{M_r}^\mathsf{T} \boldsymbol{P}^\mathsf{T} \boldsymbol{g} \Rightarrow \boldsymbol{M_r} \boldsymbol{M_r}^\mathsf{T} \boldsymbol{P}^\mathsf{T} \boldsymbol{g} = \boldsymbol{M_r} \boldsymbol{y}.$$

Conversely, choosing $\hat{\boldsymbol{y}} = \boldsymbol{y} - \boldsymbol{M_r}^\mathsf{T} \boldsymbol{P}^\mathsf{T} \boldsymbol{g}$ in (4.11) results in $\boldsymbol{M_r} \hat{\boldsymbol{y}} = \boldsymbol{0}$. As such $\hat{\boldsymbol{y}}$ lies in the kernel of $\boldsymbol{M_r}$, i.e., $\exists \boldsymbol{x}^* : \hat{\boldsymbol{y}} = \boldsymbol{\Gamma} \boldsymbol{x}^*$. Consequently, by Corollary 4.3.1, it holds that:

$$\hat{\boldsymbol{y}} + \tilde{\boldsymbol{y}} = \boldsymbol{\Gamma} \boldsymbol{x}^* + \boldsymbol{M_r}^\mathsf{T} \boldsymbol{P}^\mathsf{T} \boldsymbol{g} = \boldsymbol{y} \Rightarrow \boldsymbol{G}^\mathsf{T} \boldsymbol{\Gamma} \boldsymbol{x}^* = \boldsymbol{G}^\mathsf{T} \boldsymbol{y}.$$

□

Example 4.3.3. Reconsider the same data and parameters as in Example 4.3.1. Working out (4.11) results in the system of polynomial equations:

$$\begin{bmatrix} 0.0318 - 0.7060\,\alpha & 2\,\alpha + \alpha^3 & \beta + \alpha^2\,\beta & \alpha\,\beta^2 \\ 0.0462 - 0.0318\,\alpha & -1 - 2\,\alpha^2 & -\alpha\,\beta & 0 \\ 0.0971 - 0.2769\,\alpha & 0 & -1 - \alpha^2 & -2\,\alpha\,\beta \\ 0.2769 - 0.7060\,\beta & \alpha^2\,\beta & \alpha + \alpha\,\beta^2 & 2\,\beta + \beta^3 \\ 0.0971 - 0.0318\,\beta & -2\,\alpha\,\beta & -1 - \beta^2 & 0 \\ 0.8235 - 0.2769\,\beta & 0 & -\alpha\,\beta & -1 - 2\,\beta^2 \end{bmatrix} \begin{bmatrix} 1 \\ \boldsymbol{g} \end{bmatrix} = \boldsymbol{0}.$$

The solutions are precisely the same eigentuples as those from Example 4.3.1—up to numerical errors, that is—given in Tables 4.1 and 4.2. The corresponding values for $\boldsymbol{g}$ and the misfit norms are given in Table 4.3. ◇

4.4 Real models

Now that the proposed methodology has been outlined, it remains to be shown that the local minima of (4.2) are in fact part of the solutions $\mathcal{C}$. Indeed,

index	g_1	g_2	g_3
1	0.0378	0.0214	0.1383
2	0.0462	0.0976	0.7625
3	0.0031	0.0996	0.8234
4-5	$0.5502 \mp 0.1130\mathrm{i}$	$0.1939 \pm 0.6935\mathrm{i}$	$-0.1242 \pm 0.0539\mathrm{i}$
6-7	$0.8834 \pm 0.1322\mathrm{i}$	$0.2092 \mp 0.9014\mathrm{i}$	$-0.1459 \mp 0.0883\mathrm{i}$
8-9	$0.0395 \pm 0.0040\mathrm{i}$	$0.0289 \pm 0.0263\mathrm{i}$	$0.1530 \pm 0.1317\mathrm{i}$
10-11	$0.5930 \mp 0.0687\mathrm{i}$	$0.1979 \pm 0.6682\mathrm{i}$	$0.0076 \pm 0.1439\mathrm{i}$
12-13	$0.4549 \pm 0.0180\mathrm{i}$	$0.0538 \pm 0.1446\mathrm{i}$	$0.7757 \mp 0.0286\mathrm{i}$

Table 4.3: $\boldsymbol{g}$-values of the solutions of (4.12). The index refers to the entries row in Tables 4.1 and 4.2, which contain the corresponding values of $\boldsymbol{\lambda}$ and $\|\tilde{\boldsymbol{y}}\|$.

the polynomial characterizations were obtained under the assumptions that local minima of (4.2) are also stationary points $\mathcal{S}$ of the relaxation (4.4), that $x_i \neq 0$ for $i = 1, \ldots, n$, and for (4.11), that the eigentuples are distinct. In the following subsections, we formally prove these statements. Moreover, it is also shown that the solutions of interest are almost always isolated solutions of the polynomial characterization, such that numerical solvers can indeed find these points. Finally, we briefly highlight a limitation of the proposed methodology: the absence of a global minimum and how it can be detected.

4.4.1 Locally minimal real models

It was already highlighted in Section 4.3 that it is not immediately obvious that all local minima of (4.2) are stationary points of (4.4), while all real models that are stationary points of the latter, relaxed problem are clearly stationary points of the former. This is due to the additional constraint that restricts the 'movement' of the parameters to directions that respect the complex-conjugate structure of real models. With respect to this restricted set of directions, the gradient of the objective function may vanish, while not necessarily vanishing in general directions. To show that this is not an issue for the misfit problem (4.2), we first formalize what it means to be a local minimum of this problem.

Definition 4.4.1. A real model $(\boldsymbol{\Lambda}^*, \boldsymbol{x}^*) \in \mathcal{R}$ is a *locally minimal real model* of (4.2) when there exists a $\delta \in \mathbb{R}^+$ such that for any $(\boldsymbol{\Lambda}, \boldsymbol{x})$, where $\|\boldsymbol{\Lambda}^* - \boldsymbol{\Lambda}\|_{\mathrm{F}} < \delta$ and $(\boldsymbol{\Lambda}, \boldsymbol{x}) \in \mathcal{R}$, it holds that $\|\tilde{\boldsymbol{y}}^*\|_2 \leq \|\tilde{\boldsymbol{y}}\|_2$. The set of locally minimal real models will be denoted as $\mathcal{M}$. △

λ_1, λ_2	x_1, x_2	$\|\tilde{\boldsymbol{y}}\|_2^2$
$-0.5774, 1.0948$	$3.098, 4.793$	18.155
$1.2088, 15.4705$	$4.8414, -9.3275 \times 10^{-5}$	23.326
$1.0306, 1.0306e3$	$4.1903, 1.6417$	30.425
$-0.2463 \pm 0.9047i$	$1.4926 \mp 0.9746i$	197.55
$-2.1340, -2.1340$	$0.2510, -0.1245$	214.19

Table 4.4: Real models that satisfy the characterization (4.8) for the data $\boldsymbol{y}$ presented in Example 4.4.1

A globally minimal real model is, of course, also a locally minimal real model as it satisfies this definition for $\delta = +\infty$.

Observe that the real local minimality of a model only imposes that no better value of the eigentuples $\boldsymbol{\Lambda}$ can be found arbitrarily close, disregarding $\boldsymbol{x}$. Indeed, recall that given $\boldsymbol{\Lambda}$, the corresponding optimal value for $\boldsymbol{x}$ is readily obtained by solving a linear projection, essentially making $\boldsymbol{\Lambda}$ the only decision variable. However, in case $\boldsymbol{\Gamma}$ is not of full column rank, e.g., when $\boldsymbol{\lambda}_i = \boldsymbol{\lambda}_j$, this projection does not define $\boldsymbol{x}$ uniquely, though it does uniquely define the objective value. As such, to correctly handle these 'singular' cases, the local minimality of a model should only depend on the optimality of $\boldsymbol{\Lambda}$.

Example 4.4.1 briefly highlights why we need to discern between locally minimal real models and the minima of the relaxed optimization problem (4.4). Here, the globally optimal real model is a saddle point of (4.4), rather than a minimum. This shows that, in general, $\mathcal{M}$ are not minima of the relaxed problem.

Example 4.4.1. Consider the problem of fitting an order $n = 2$, 1D model to the data sequence $\boldsymbol{y}^\mathsf{T} = \begin{bmatrix} 7 & 2 & 9 & 8 & 5 \end{bmatrix}$. Up to permutation of the poles, the real models that solve the corresponding set of equations (4.8) are listed in Table 4.4.

Using the projection discussed in 4.3.2 to determine $\boldsymbol{x}$, Figure 4.3 then plots the objective $\|\tilde{\boldsymbol{y}}\|_2^2$ around the global minimum $(\lambda_1, \lambda_2) = (-0.5774, 1.0948)$. The left figure, where only real eigentuples (λ_1, λ_2) are considered, demonstrates that this point is indeed a local minimum for this parameterization. Since no models with complex-conjugate eigentuples are directly adjacent to this model, this suffices to conclude that it is a local minimum of the misfit problem (4.2).

However, relaxing the problem as in (4.4) creates additional dimensions to achieve a better fit: the complex part of (λ_1, λ_2). The right-hand-side figure plots the objective function over eigentuples of the form $(-0.5774 + \mathrm{i}a, 1.0948 + \mathrm{i}b)$ for varying $a, b \in \mathbb{R}$. This figure clearly shows that $(-0.5774, 1.0948)$ is not a

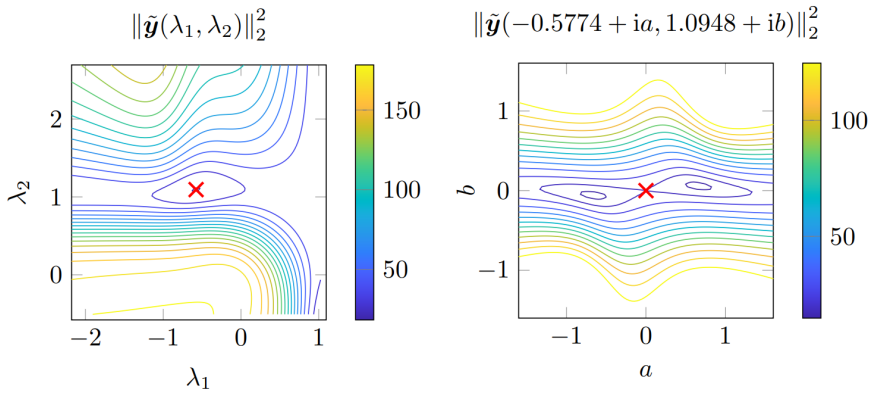


Figure 4.3: Plots of the misfit $\|\tilde{\boldsymbol{y}}\|_2^2$ for varying real (left) and imaginary (right) values of (λ_1, λ_2) associated with the data $\boldsymbol{y}$ given in Example 4.4.1. The globally minimal real model $(-0.5774, 1.0948)$ is indicated by $\times$.

local minimum of the relaxed problem. Indeed, it corresponds to a saddle point in this plot, and can thus be locally improved upon using such complex-valued eigentuples. $\diamond$

From Example 4.4.1, it can be concluded that, while the best-fitting real model can be found through solving (4.8), the global minimum of the relaxation (4.5) is not always found in this way. Moreover, the global minimum of (4.2), while a stationary point, may not even be a local minimum of this relaxation. Intuitively, this makes sense, as the absence of the realness constraint enlarges the search space and provides more search directions to improve the fit.

Notice that, using the proposed methodology, the globally minimal real model was indeed found in Example 4.4.1. The following theorem ensures that this is always the case, such that it makes sense to search for them amongst the stationary points $\mathcal{S}$ by solving (4.8) or (4.11).

Theorem 4.4.1. *Any locally minimal real model satisfies* (4.6)*, i.e.,* $\mathcal{M} \subseteq \mathcal{S}$.

Proof. The proof is by contradiction. Suppose that $(\boldsymbol{\Lambda}^*, \boldsymbol{x}^*) \in \mathcal{M}$ does not satisfy the FONC (4.6), then, by abuse of notation, there exists some subset of variables $\boldsymbol{z} \subseteq (\boldsymbol{\Lambda}^*, \boldsymbol{x}^*)$ such that $\nabla_{\boldsymbol{z}} \|\tilde{\boldsymbol{y}}^*\|_2^2 \neq \boldsymbol{0}$. Due to the nonzero gradient, there then exists a direction $\Delta\boldsymbol{z}$ such that the directional derivative w.r.t $\underline{\Delta\boldsymbol{z}}$ is: $\Delta\boldsymbol{z}^\mathsf{T} \nabla_{\boldsymbol{z}} \|\tilde{\boldsymbol{y}}^*\|_2^2 + \overline{\Delta\boldsymbol{z}}^\mathsf{T} \nabla_{\overline{\boldsymbol{z}}} \|\tilde{\boldsymbol{y}}^*\|_2^2 = \gamma < 0$. Also note that $\nabla_{\overline{\boldsymbol{z}}} \|\tilde{\boldsymbol{y}}^*\|_2^2 = \overline{\nabla_{\boldsymbol{z}} \|\tilde{\boldsymbol{y}}^*\|_2^2}$ [2].

Now, since $(\boldsymbol{\Lambda}^*, \boldsymbol{x}^*)$ defines a real model, there exists a set of complex-conjugate variables $\boldsymbol{w} \subseteq (\boldsymbol{\Lambda}^*, \boldsymbol{x}^*)$ such that $\bar{\boldsymbol{z}} = \boldsymbol{w}$. From (4.6), it can be worked out that $\nabla_{\boldsymbol{w}} \|\tilde{\boldsymbol{y}}^*\|_2^2 = \overline{\nabla_{\boldsymbol{z}} \|\tilde{\boldsymbol{y}}^*\|_2^2}$. As such, choosing $\Delta \boldsymbol{w} = \overline{\Delta \boldsymbol{z}}$ results in a directional derivative of $\Delta \boldsymbol{w}^{\mathsf{T}} \nabla_{\boldsymbol{w}} \|\tilde{\boldsymbol{y}}^*\|_2^2 + \overline{\Delta \boldsymbol{w}}^{\mathsf{T}} \nabla_{\overline{\boldsymbol{w}}} \|\tilde{\boldsymbol{y}}^*\|_2^2 + \gamma = 2\gamma < 0$ w.r.t. the direction $\begin{bmatrix} \Delta \boldsymbol{z}^{\mathsf{T}} & \Delta \boldsymbol{w}^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}}$.

Consequently, this is a descent direction which retains the realness of the model. There thus exist arbitrarily small steps along this direction that improve the fit $\|\tilde{\boldsymbol{y}}\|_2^2$, contradicting its local minimality. □

4.4.2 Model order of real minima

While Theorem 4.4.1 shows that $\mathcal{M}$ can be found as solutions of the FONC (4.6), it remains to be shown that the x_i are nonzero, as was assumed in the derivation of (4.8). Moreover, it also remains unclear whether the condition of distinct eigentuples posed in Theorem 4.3.2 is satisfied by such a model. The following theorem states that both these conditions are indeed met for locally minimal real models $\mathcal{M}$. This finally proves that all locally minimal real models, including the globally minimal one, can be found by solving (4.8) or, by Theorem 4.3.2, solving (4.11).

Theorem 4.4.2. *Any* $(\boldsymbol{\Lambda}^*, \boldsymbol{x}^*) \in \mathcal{M}$ *for which* $\|\tilde{\boldsymbol{y}}^*\| \neq 0$ *satisfies:* $\forall i \in \{1, \ldots, n\} : x_i \neq 0$ *and* $\forall i, j \in \{1, \ldots, n\}$ *with* $i \neq j : \boldsymbol{\lambda}_i \neq \boldsymbol{\lambda}_j$.

Proof. The proof is by contradiction. Note that, by Definition 4.4.1, to contradict the local real minimality of the model, it suffices to show that there always exists a better choice of $\boldsymbol{\Lambda}$ arbitrarily close to $\boldsymbol{\Lambda}^*$, while $\boldsymbol{x}$ can be arbitrarily far from $\boldsymbol{x}^*$.

Suppose that for some i it holds that $x_i^* = 0$ and define the Vandermonde vector $\boldsymbol{v}(\boldsymbol{z}) \in \mathbb{C}^N[\boldsymbol{z}]$ as $\boldsymbol{v}_{\boldsymbol{\kappa}}(\boldsymbol{z}) = \boldsymbol{z}^{\boldsymbol{\kappa}}$. Consider the polynomial $\tilde{y}^*(\boldsymbol{z}) = (\tilde{\boldsymbol{y}}^*)^{\mathsf{T}} \boldsymbol{v}(\boldsymbol{z}) = \sum_{\boldsymbol{\kappa} \in \mathcal{K}} \tilde{\boldsymbol{y}}_{\boldsymbol{\kappa}} \boldsymbol{z}^{\boldsymbol{\kappa}}$, which is nonzero since since $\tilde{\boldsymbol{y}}^* \neq \boldsymbol{0}$. Akin to the interpolation condition (4.10), it holds that $0 = \tilde{y}^*(\boldsymbol{\lambda}_i^*)$, as the real model satisfies (4.7a). There exists an arbitrarily small, real perturbation $\boldsymbol{\epsilon} \in \mathbb{R}^m$ so that $\tilde{y}^*(\boldsymbol{\lambda}_i^* + \boldsymbol{\epsilon}) \neq 0$.

For the sake of brevity, we presume that $\mathrm{Re}(\tilde{y}^*(\boldsymbol{\lambda}_i^* + \boldsymbol{\epsilon})) \neq 0$. The argument can be easily adjusted to cover the case where only $\mathrm{Im}(\tilde{y}^*(\boldsymbol{\lambda}_i^* + \boldsymbol{\epsilon})) \neq 0$.

Since $(\boldsymbol{\Lambda}^*, \boldsymbol{x}^*)$ defines a real model, there is a complex-conjugate eigentuple to $\boldsymbol{\lambda}_i$, that is, $\exists j : x_j = 0$ and $(\boldsymbol{\lambda}_i^*, x_i^*) = \overline{(\boldsymbol{\lambda}_j^*, x_j^*)}$. Then consider $\tilde{y}^*(\boldsymbol{\lambda}_i^* + \boldsymbol{\epsilon}) + \tilde{y}^*(\boldsymbol{\lambda}_j^* + \overline{\boldsymbol{\epsilon}}) = \tilde{y}^*(\boldsymbol{\lambda}_i^* + \boldsymbol{\epsilon}) + \overline{\tilde{y}^*(\boldsymbol{\lambda}_i^* + \boldsymbol{\epsilon})} = 2\mathrm{Re}(\tilde{y}^*(\boldsymbol{\lambda}_i^* + \boldsymbol{\epsilon}))$. When $\mathrm{Re}(\tilde{y}^*(\boldsymbol{\lambda}_i^* + \boldsymbol{\epsilon})) \neq 0$,

the component along $\mathrm{Re}(\boldsymbol{v}(\boldsymbol{\lambda}_i^* + \boldsymbol{\epsilon}))$ can be projected away so that:

$$\|\tilde{\boldsymbol{y}}^{\mathrm{new}}\|_2^2 = \left\| \tilde{\boldsymbol{y}}^* - 2\mathrm{Re}(\tilde{y}^*(\boldsymbol{\lambda}_i^* + \boldsymbol{\epsilon}))\frac{\mathrm{Re}(\boldsymbol{v}(\boldsymbol{\lambda}_i^* + \boldsymbol{\epsilon}))}{\|\mathrm{Re}(\boldsymbol{v}(\boldsymbol{\lambda}_i^* + \boldsymbol{\epsilon}))\|_2^2} \right\|_2^2 < \|\tilde{\boldsymbol{y}}^*\|_2^2. \tag{4.12}$$

It can be verified that $\|\tilde{\boldsymbol{y}}^{\mathrm{new}}\|_2^2$ is attained by the real model:

$$\boldsymbol{\lambda}_i^{\mathrm{new}} = \boldsymbol{\lambda}_i^* + \boldsymbol{\epsilon}, \quad x_i^{\mathrm{new}} = \frac{\mathrm{Re}(\tilde{y}^*(\boldsymbol{\lambda}_i^* + \boldsymbol{\epsilon}))}{\|\mathrm{Re}(\boldsymbol{v}(\boldsymbol{\lambda}_i^* + \boldsymbol{\epsilon}))\|_2^2}$$

$$\boldsymbol{\lambda}_j^{\mathrm{new}} = \boldsymbol{\lambda}_j^* + \overline{\boldsymbol{\epsilon}}, \quad x_j^{\mathrm{new}} = \frac{\mathrm{Re}(\tilde{y}^*(\boldsymbol{\lambda}_i^* + \boldsymbol{\epsilon}))}{\|\mathrm{Re}(\boldsymbol{v}(\boldsymbol{\lambda}_i^* + \boldsymbol{\epsilon}))\|_2^2},$$

while keeping other model parameters the same as in $(\boldsymbol{\Lambda}^*, \boldsymbol{x}^*)$. Since $\|\boldsymbol{\Lambda}^{\mathrm{new}} - \boldsymbol{\Lambda}^*\|_{\mathrm{F}}^2 = 2\|\boldsymbol{\epsilon}\|^2$ can be arbitrarily small, this contradicts that $(\boldsymbol{\Lambda}^*, \boldsymbol{x}^*) \in \mathcal{M}$.

Lastly, suppose $\exists i, j$ with $i \neq j$ for which $\boldsymbol{\lambda}_i^* = \boldsymbol{\lambda}_j^*$. This case can be reduced to an equivalent model with $x_i = 0$: Let (s, t) be such that $\overline{(\boldsymbol{\lambda}_i^*, x_i)} = (\boldsymbol{\lambda}_s^*, x_s)$ and $\overline{(\boldsymbol{\lambda}_j^*, x_j)} = (\boldsymbol{\lambda}_t^*, x_t)$, and choose $x_j^{\mathrm{new}} = x_j^* + x_i^*$ and $x_i^{\mathrm{new}} = 0$ and $x_t^{\mathrm{new}} = x_t^* + x_s^*$ and $x_s^{\mathrm{new}} = 0$, while keeping the rest of the parameters the same. By the first part of the proof, this then contradicts the local minimality of the model. □

Notice that a locally minimal real model thus has n nonzero contributions of n distinct eigentuples, ensuring that a model of order at least n is needed to describe the corresponding model-compliant data sequence $\hat{\boldsymbol{y}}$. Informally, one could state that the model has *nth order dynamics*. Indeed, any model that does not meet these restrictions can be equivalently described using a model of order at most $n - 1$. Either remove the eigentuple with no contribution, i.e., for which $x_i = 0$, or merge the contributions of a pair of equal eigentuples, as was done in the proof of Theorem 4.4.2. This proof also indicates that such lower-order models are saddle points, in the sense that there is always a better choice of $\boldsymbol{\Lambda}$ arbitrarily close by.

Consequently, such models can just as well be found by solving (4.8) for a lower model order. Indeed, substituting $\boldsymbol{\lambda}_i = \boldsymbol{\lambda}_j$ into these equations results in a set of equations equivalent to the $(n-1)$th order version of (4.8), so that optimal solutions to a lower-order problem 'carry over' to saddle points of the nth order problem.

Remark: Since a perfect fit, i.e., $\tilde{\boldsymbol{y}} = \boldsymbol{0}$, cannot be improved upon, posing a problem for a proof by contradiction, this case is excluded in Theorem 4.4.2. In case a real model with $\tilde{\boldsymbol{y}} = \boldsymbol{0}$ exists and has *nth order dynamics*, it still

satisfies both (4.8) and (4.11). If not, it can be equivalently described using a lower model order, so that at least one equivalent model appears as a solution to (4.8).

For (4.11), adding any eigentuple $\boldsymbol{\lambda}_n$ to a $(n-1)$th order model ensures that, by Theorems 4.2.1 and 4.2.2, $\hat{\boldsymbol{y}}$ still lies in the kernel of the corresponding Macaulay matrix. Since $\tilde{\boldsymbol{y}} = \mathbf{0}$ lies in the row space of any Macaulay matrix, choosing $\boldsymbol{g} = \mathbf{0}$ then provides a solution to the higher order problem as well. As such, any globally minimal real model is always a solution to both characterizations.

4.4.3 Real, local minima are isolated

In case of a continuum of solutions, numerical polynomial solvers, of course, only compute a finite number of 'representatives', called a *witness set*, for each irreducible part of the solution set [24]. While we now know that models in $\mathcal{M}$ are indeed solutions to our polynomial characterizations, this does not guarantee that they do not lie on a continuum of complex solutions, for which such a witness set can be provided at best. This witness set may not contain real models, while the continuum they represent may contain the real model of interest. Thus, to ensure that this is not the case so that the models of interest can reliably be found using polynomial root-finding techniques, it remains to be shown that they are isolated solutions. Theorem 4.4.3 shows that, for *almost all* $\boldsymbol{y}$, this is indeed the case.

Theorem 4.4.3. *There exists a dense, open subset $\mathcal{O} \subseteq \mathbb{C}^N$, such that for $\boldsymbol{y} \in \mathcal{O}$, any $(\boldsymbol{\Lambda}, \boldsymbol{x}) \in \mathcal{M}$ is an isolated solution of* (4.8) *when $\mathcal{K}$ meets the condition of Theorem 4.3.2. Secondly, the corresponding $\boldsymbol{\Lambda}$ are isolated points of the set $\{\boldsymbol{\Lambda} \in \mathbb{C}^{n\times m} \mid \exists \boldsymbol{g} \in \mathbb{C}^q : (\boldsymbol{\Lambda}, \boldsymbol{g})$ satisfies* (4.11)$\}$.

Proof. To show the first statement, it suffices to show that for a dense, open subset of $\boldsymbol{y}$-values, the Jacobian of (4.8) has a nullity of zero in the solutions with distinct eigentuples and nonzero x_i, as this upper bounds the local dimension of the solution set [75]. By Theorem 4.4.2, these solutions include $\mathcal{M}$.

First, we eliminate the solutions where $\boldsymbol{\lambda}_i = \boldsymbol{\lambda}_j$ or $x_i = 0$ by applying Rabinowitsch's trick [Chapter 4, Theorem 2 of 32, 152]. Choose any[8] $\mathbf{0} \neq \boldsymbol{v} \in \mathbb{C}^m$ and introduce the new variables u_{ij} and w_i. Then, by abuse of notation, define the set of equations $\boldsymbol{c} = \{\forall i, j; i \neq j : 1 - u_{ij}\boldsymbol{v}^\mathsf{T}(\boldsymbol{\lambda}_i - \boldsymbol{\lambda}_j) = 0\} \cup \{\forall i : 1 - w_i x_i = 0\}$ and construct the extended set of equations: $\boldsymbol{p} = \{(4.8)\} \cup \boldsymbol{c}$. For every solution $(\boldsymbol{\Lambda}, \boldsymbol{x}, \boldsymbol{y})$ of (4.8) with $x_i \neq 0$ and $\boldsymbol{v}^\mathsf{T}(\boldsymbol{\lambda}_i - \boldsymbol{\lambda}_j) \neq 0$, there then exist

[8] As the following construction holds for any nonzero value of $\boldsymbol{v}$, this essentially only eliminates cases with $\boldsymbol{\lambda}_i = \boldsymbol{\lambda}_j$.

unique values for u_{ij} and w_i that solve $\boldsymbol{p}$. Conversely, any solution of $\boldsymbol{p}$ implies a corresponding solution of (4.8) by construction. Now considering $\boldsymbol{y}$ as variables, this allows us to define the variety $V = \{\boldsymbol{z} = (\boldsymbol{\Lambda}, \boldsymbol{x}, \boldsymbol{u}, \boldsymbol{w}, \boldsymbol{y}) \in \mathbb{C}^t \mid \boldsymbol{p}(\boldsymbol{z}) = \boldsymbol{0}\}$.

Next, consider a model $(\boldsymbol{\Lambda}^*, \boldsymbol{x}^*)$ with $x_i^* \neq 0$ and eigentuples so that $v^\mathsf{T}(\boldsymbol{\lambda}_i - \boldsymbol{\lambda}_j) \neq 0$. Selecting $\boldsymbol{y}^* = \boldsymbol{\Gamma}^*\boldsymbol{x}^*$, there then exist values $\boldsymbol{u}^*, \boldsymbol{w}^*$ so that $\boldsymbol{z}^* = (\boldsymbol{\Lambda}^*, \boldsymbol{x}^*, \boldsymbol{u}^*, \boldsymbol{w}^*, \boldsymbol{y}^*) \in V$. Then, up to row and column permutations, the Jacobian $\boldsymbol{J}_{\boldsymbol{p}}^*$ in $\boldsymbol{z}^* = (\boldsymbol{\Lambda}^*, \boldsymbol{x}^*, \boldsymbol{u}^*, \boldsymbol{w}^*, \boldsymbol{y}^*)$ of $\boldsymbol{p}$ is of the form:

$$\boldsymbol{J}_{\boldsymbol{p}}^* = \begin{array}{c} \begin{array}{ccc} \nabla_{(\boldsymbol{\Lambda},\boldsymbol{x})} & \nabla_{(\boldsymbol{u},\boldsymbol{w})} & \nabla_{\boldsymbol{y}} \end{array} \\ \begin{bmatrix} \boldsymbol{J}_1^* & \boldsymbol{0} & (\boldsymbol{G}^*)^\mathsf{T} \\ \boldsymbol{B}_1 & \boldsymbol{D} & \boldsymbol{B}_2 \end{bmatrix} \end{array} \begin{array}{l} \boldsymbol{J}_{(4.8)} \\ \boldsymbol{J}_{\boldsymbol{c}} \end{array},$$

where $\boldsymbol{J}_1^* = (\boldsymbol{G}^*)^\mathsf{T} \begin{bmatrix} \boldsymbol{\Gamma}^* & \left(\boldsymbol{\Gamma}^{(\boldsymbol{\lambda})}\right)^* (\boldsymbol{I}_m \otimes \operatorname{diag}(\boldsymbol{x}^*)) \end{bmatrix} = (\boldsymbol{G}^*)^\mathsf{T}\boldsymbol{H}$ and $\boldsymbol{D}$ is diagonal and full rank.

Remark that for any $\boldsymbol{z} \in V$, the corresponding $\boldsymbol{G}$ is of full column rank, since its columns span the null space of[9] $\boldsymbol{M}_{\boldsymbol{r}^2}$, which has a nullity of $(m+1)n$. Since the columns of $\boldsymbol{H}^*$ are scaled versions of the columns of $\boldsymbol{G}^*$, $\boldsymbol{J}_1^*$ is of full rank. Together with $\boldsymbol{D}$ being invertible, this implies that $\boldsymbol{J}_{\boldsymbol{p}}^*$ is of full row-rank as well.

From the implicit function theorem it now follows that there exists an open subset $U \subseteq \mathbb{C}^N$ around $\boldsymbol{y}^*$ and a function $\boldsymbol{f}$ such that $\forall \boldsymbol{y} \in U : (\boldsymbol{\Lambda}, \boldsymbol{x}, \boldsymbol{u}, \boldsymbol{w}) = \boldsymbol{f}(\boldsymbol{y})$ and $\boldsymbol{p}(\boldsymbol{f}(\boldsymbol{y}), \boldsymbol{y}) = \boldsymbol{0}$. This implies that $\{(\boldsymbol{f}(\boldsymbol{y}), \boldsymbol{y}) \mid \boldsymbol{y} \in U\} \subseteq V$ so that the variety is N-dimensional at $\boldsymbol{z}^*$.

Now define the morphism $\pi : V \to \pi(V) : \boldsymbol{z} \mapsto \boldsymbol{y}$. As $U \subseteq \pi(V)$, it is N-dimensional. By the closure theorem [32], its closure $\overline{\pi(V)}$ then is an N-dimensional variety in $\mathbb{C}^N$, implying $\overline{\pi(V)} = \mathbb{C}^N$ and that $\pi(V)$ is a dense, open subset of[10] $\mathbb{C}^N$.

Finally, the Jacobian of (4.8) w.r.t the model parameters $(\boldsymbol{\Lambda}, \boldsymbol{x})$ is a square, polynomial matrix in $(\boldsymbol{\Lambda}, \boldsymbol{x}, \boldsymbol{y})$, so that its determinant is a polynomial $\delta(\boldsymbol{z})$ which defines the variety $W = \{\boldsymbol{z} \in \mathbb{C}^t \mid \delta(\boldsymbol{z}) = 0\}$. Since in $\boldsymbol{z}^*$, the Jacobian $\boldsymbol{J}_1^*$ is of full rank, it holds that $V \not\subseteq W$, so that $V \cap W$ is a variety of dimension $d < N$. Thus, $\dim \pi(V \cap W) < N$, implying that it is a closed subset of $\pi(V)$, so that its complement is a dense, open subset of $\overline{\pi(V)} = \mathbb{C}^N$, where this Jacobian has a nullity of zero.

By Theorem 4.3.2, the second statement follows directly from the first statement. The solutions of (4.11) are not isolated in $\boldsymbol{g}$, however, since $\boldsymbol{M}_{\boldsymbol{r}^2}$ generally has a nontrivial null space. □

[9] Apply Lemma 4.2.2: $\boldsymbol{r}^2$ has precisely $(m+1)n$ solutions, counting multiplicities, and $\mathcal{K}$ is such that the degree of regularity is exceeded.

[10] Note that for any $\boldsymbol{y} \in \pi(V)$ there is at least one corresponding point in V, i.e., there exists a solution of (4.8) that satisfies the assumptions on $(\boldsymbol{\Lambda}, \boldsymbol{x})$.

4.4.4 Limitations

To conclude this section, we highlight a key limitation of focusing on the stationary points of the optimization problem. It has been shown that all locally minimal real models can be found through the proposed approach, and, clearly, a global minimum is necessarily a local one. However, keep in mind that a globally minimal real model may not exist at all. In such cases, solving (4.8) or (4.11) only leads to the locally minimal real models, which may still be improved upon by another real model, as illustrated in Example 4.4.2 below. To detect such cases, a sum-of-squares decomposition [20, 114] can be employed to get a tight lower bound for polynomial optimization problems.

Example 4.4.2. Given $\boldsymbol{y} = \begin{bmatrix} 0.1 & 0 & 0 & 1 \end{bmatrix}^\mathsf{T}$, consider the problem of fitting a first-order 1D model this data. The objective function is then:

$$\|\tilde{\boldsymbol{y}}\|^2 = |0.1 - x|^2 + |\lambda x|^2 + \left|\lambda^2 x\right|^2 + \left|1 - \lambda^3 x\right|^2.$$

Choosing $x = \frac{1}{\lambda^3}$ and taking the limit then results in $\lim_{\lambda \to \pm\infty} \left\|\tilde{\boldsymbol{y}}(\lambda, \frac{1}{\lambda^3})\right\|^2 = 0.01$. This provides an upper bound on the infimum of $\|\tilde{\boldsymbol{y}}\|^2$. If a better fit could be obtained for a finite, real model (λ, x), there would have to be a local real minimum corresponding to such a point. However, solving (4.8), yields two real models that do not improve upon this bound.

λ	x	$\|\tilde{\boldsymbol{y}}\|^2$
0.0334	0.0999	1.0000
0.0000	0.1000	1.0000

Even though, by Theorem 4.4.1, these models include all locally minimal real models, neither of them define a globally minimal real model, as there exist finite parameter values leading to a lower objective function, e.g., $\lambda = 1, x = 0.1$, leading to $\|\tilde{\boldsymbol{y}}\|^2 = 0.83$. ◇

4.5 Lower-order models and infinite solutions

A polynomial characterization of the local minima is desirable because there exists a variety of numerical algorithms that can obtain all solutions to such problems. Broadly speaking, there are two classes of polynomial solvers: algebraic methods [44, 72, 169] and homotopy-based methods [24, 192]. We now illustrate the infinite number of *lower-order* solutions that occur in the presented polynomial-based characterizations. This observation limits the choice of viable

solvers, as currently, only homotopy-based root-finding methods can handle an infinite number of solutions. Remark that, at the cost of additional variables and the entailing computational load, these infinite solution components may be removed using Rabinowitsch's trick mentioned in the proof of Theorem 4.4.3. Finally, we briefly discuss rephrasing these problems as multiparameter eigenvalue problems (MEPs).

4.5.1 Polynomial formulations

Example 4.3.1 already highlighted how an infinite number of solutions with $x_i = 0$ is excluded by leveraging Theorem 4.4.2 to simplify the equations. Excluding non-interesting solutions in this way is of interest as it reduces the computational load of solving the equations. However, the solution sets of (4.8) and (4.11) do still contain other instances of lower-order models: those for which $\exists i, j, i \neq j : \boldsymbol{\lambda}_i = \boldsymbol{\lambda}_j$.

Let us first take a closer look at how these lower-order models appear as solutions to (4.8). In case $\boldsymbol{\lambda}_i = \boldsymbol{\lambda}_j, i \neq j$, $\boldsymbol{\Gamma}$ is not of full column rank and does not define a unique optimum for $\boldsymbol{x}$ through the projection discussed in Section 4.3.2. Instead, the optimal value for $\boldsymbol{x}$ is only defined up to null $\boldsymbol{\Gamma}$. Furthermore, such solutions always exist: Each solution of the order $n-1$ problem (4.8) can be extended to a solution of the corresponding nth order problem by choosing[11] $\boldsymbol{\lambda}_n = \boldsymbol{\lambda}_i$ and $x_i = 0$ for some i, thus inducing an infinite number of equivalent solutions in the higher-order problem. This observation is illustrated in Example 4.5.1 below.

Example 4.5.1. Consider the problem of fitting a second-order 1D model to the following data:

$$\boldsymbol{y}^\mathsf{T} = \begin{bmatrix} 1 & \frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} & -1 & -\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} & 1 & \frac{\sqrt{2}}{2} \end{bmatrix}.$$

Now, substitute in a lower-order solution of the form $\lambda_1 = \lambda_2 = 0.2984$ into (4.8), resulting in:

$$\boldsymbol{G}\boldsymbol{\Gamma}\boldsymbol{x} - \boldsymbol{G}\boldsymbol{y} = \begin{bmatrix} 1.0977 & 1.0977 & 1.1828 \\ 1.0977 & 1.0977 & 1.1828 \\ 0.3596 & 0.3596 & 0.3874 \\ 0.3596 & 0.3596 & 0.3874 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ -1 \end{bmatrix} = \boldsymbol{0}.$$

The kernel of the associated observability matrix $\boldsymbol{\Gamma}$ is spanned by $\begin{bmatrix} 1 & -1 \end{bmatrix}$. As such, all solutions $\boldsymbol{x}$ to these singular, linear equations can thus be described

[11] It can be verified that this choice of eigentuple only generates additional equations that were already present in the lower-order problem.

as $\boldsymbol{x}^\mathsf{T} = \begin{bmatrix}1.0774 & 0\end{bmatrix} + c\begin{bmatrix}1 & -1\end{bmatrix}, c \in \mathbb{C}$, leading to a one-dimensional set of equivalent lower-order solutions for (4.8). ◇

The characterization (4.11), on the other hand, relies on describing the misfit as $\tilde{\boldsymbol{y}} = \boldsymbol{M}_{\boldsymbol{r}^2}^\mathsf{T}\boldsymbol{g}$. Here, the coefficient matrix may also have a nontrivial kernel, again causing the auxiliary variables $\boldsymbol{g}$ to be defined non-uniquely by this equation. Moreover, the rows of a Macaulay matrix, and, by extension, $\boldsymbol{M}_{\boldsymbol{r}^2}$, do generally not form a basis [12], regardless of $\boldsymbol{\Lambda}$. Consequently, for most problems, all solutions to (4.11) entail an infinite number of equivalent values of $\boldsymbol{g}$.

However, assuming distinct eigentuples, Lemma 4.3.1, implies that the rank of $\boldsymbol{M}_{\boldsymbol{r}^2} \in \mathbb{R}^{p\times N}$ is $N - n(m+1)$. As such, one could heuristically attempt to construct a basis by taking as many linearly independent row combinations, that is, by replacing it with $\boldsymbol{T}\boldsymbol{M}_{\boldsymbol{r}^2}$ in (4.11), where $\boldsymbol{T} \in \mathbb{R}^{N-n(m+1)\times q}$ is some matrix of full row-rank. For example, one may choose $\boldsymbol{T}$ to be a random matrix. For almost all $\boldsymbol{\Lambda}$, this eliminates the non-uniqueness of $\boldsymbol{g}$ for solutions with distinct eigentuples.

While this approach does eliminate a large number of equivalent models, lower-order solutions with $\boldsymbol{\lambda}_i = \boldsymbol{\lambda}_j, i \neq j$ again induce an infinite number of solutions. Indeed, such models cause $\boldsymbol{\lambda}_i$ to become a common root of multiplicity $m+1$ in the GCH equations $\boldsymbol{r}(\boldsymbol{\sigma})$ [186]. Using a similar analysis as in the proof of Theorem 4.3.1, its multiplicity as a root of $\boldsymbol{r}^2(\boldsymbol{\sigma})$ then rises to $\binom{m+3}{3}$, increasing the nullity of $(\boldsymbol{T})\boldsymbol{M}_{\boldsymbol{r}^2}$. This results in a rank drop of the coefficient matrix, which again leads to an infinite number of equivalent $\boldsymbol{g}$-values in (4.11).

4.5.2 Multiparameter eigenvalue problem

To leverage the fact that the auxiliary variables $\boldsymbol{x}$ and $\boldsymbol{g}$ only appear linearly in (4.8) and (4.11), respectively, these equations have been rephrased as MEPs [36, 104]. In this class of polynomial problems, one aims to find the variables $\boldsymbol{z}$ such that a polynomial matrix $\boldsymbol{L}(\boldsymbol{z}) \in \mathbb{C}^{r\times s}[\boldsymbol{z}]$ has a nontrivial null space, that is, one searches $\boldsymbol{z}$ such that there exists a vector $\boldsymbol{v}$ for which:

$$\boldsymbol{L}(\boldsymbol{z})\boldsymbol{v} = \boldsymbol{0}, \quad \|\boldsymbol{v}\|_2 = 1.$$

The corresponding value(s) for $\boldsymbol{v}$ then follow immediately from a linear equation once $\boldsymbol{z}$ is found. Exploiting this problem structure usually simplifies the problem computationally [188].

Similarly, it is straightforward to rephrase both (4.8) and (4.11) as MEPs, by replacing $\begin{bmatrix}\boldsymbol{x}^\mathsf{T} & -1\end{bmatrix}$ or $\begin{bmatrix}\boldsymbol{g}^\mathsf{T} & -1\end{bmatrix}$ by $\boldsymbol{v}^\mathsf{T}$, respectively. The associated choice of $\boldsymbol{L}(\boldsymbol{\Lambda})$ is then clear from the respective equations. This exploits the linearity

by which the auxiliary variables $\boldsymbol{x}$ and $\boldsymbol{g}$ occur, providing a computational advantage.

However, note that in both cases, the resulting MEP is not equivalent to the original polynomial system: All entries of $\boldsymbol{v}$ are variables and can thus become zero. This creates additional solutions for which $v_{n+1} = 0$ that satisfy the MEP formulation but not the original equations (4.8) or (4.11).

Furthermore, substituting $v_{n+1} = 0$ reveals that $\boldsymbol{\Gamma}$ or $(\boldsymbol{T}\boldsymbol{M}_{\boldsymbol{r}^2})$ having a nontrivial right null space suffices for a solution of this form to exist. By the analysis in the preceding subsection, we know that this condition is satisfied for any choice of parameters with $\boldsymbol{\lambda}_i = \boldsymbol{\lambda}_j, i \neq j$. Clearly, an infinite number of model parameters $\boldsymbol{\Lambda}$ meets this condition, leading to an infinite solution set. Currently, only algebraic methods exist for MEPs, which do not converge under these conditions [64, 81, 148, 149, 188].

Remark: This limitation forces the auxiliary variables to be treated as regular variables. Since (4.11) generally introduces many more auxiliary variables $\boldsymbol{g} \in \mathbb{C}^{N-n(m+1)}$ compared to $\boldsymbol{x} \in \mathbb{C}^n$ in (4.8), this leads to higher dimensional problems, adding to the computational complexity. However, in the one-dimensional case, the MEP reformulation makes the kernel-based characterization more computationally interesting because of this larger number of auxiliary variables [36, 104]. If algorithms for MEPs with infinite solution sets can be developed, this may also apply to the multidimensional kernel-based characterization (4.11).

4.6 Numerical experiments

In this section, some more elaborate examples are considered to further illustrate the efficacy of the proposed identification approach.

Example 4.6.1. Consider the problem of fitting a second-order ($n = 2$) model to the 2D data sequence $y_{\boldsymbol{\kappa}}$ given in Table 4.5. Solving (4.8) for this data sequence using *HomotopyContinuation.jl* [24], results in 69294 complex solutions, of which 75 are real models, of which 29 have real-valued eigentuples and 46 have complex-conjugate eigentuples. For the sake of brevity, only the ten best-fitting real models are presented in Table 4.6.

Note that the parameterization of $\hat{\boldsymbol{y}}$ and thus the equations are symmetric in $(\boldsymbol{\lambda}_1, x_1)$ and $(\boldsymbol{\lambda}_2, x_2)$, that is, exchanging the contributions as follows: the model $(\boldsymbol{\lambda}_1^{\text{new}}, x_1^{\text{new}}) = (\boldsymbol{\lambda}_2, x_2)$ and $(\boldsymbol{\lambda}_2^{\text{new}}, x_2^{\text{new}}) = (\boldsymbol{\lambda}_1, x_1)$ has the same $\hat{\boldsymbol{y}}$ and thus the same objective function. Consequently, only one of these permutations is provided in Table 4.6 for each pair of solutions. Under this symmetry, there

$\boldsymbol{\kappa}$	$y_{\boldsymbol{\kappa}}$	$\boldsymbol{\kappa}$	$y_{\boldsymbol{\kappa}}$	$\boldsymbol{\kappa}$	$y_{\boldsymbol{\kappa}}$
(0,0)	1.88	(1,0)	5.41	(0,1)	1.22
(2,0)	6.02	(1,1)	0.16	(0,2)	5.54
(3,0)	6.56	(2,1)	8.57	(1,2)	4.49
(0,3)	8.55	(4,0)	8.01	(3,1)	3.79
(2,2)	5.11	(1,3)	0.28	(0,4)	2.88
(5,0)	0.61	(4,1)	9.33	(3,2)	9.01
(2,3)	9.12	(1,4)	5.89	(0,5)	6.03
(6,0)	5.79	(5,1)	5.99	(4,2)	4.96
(3,3)	3.44	(2,4)	2.66	(1,5)	6.13
(0,6)	5.12	(7,0)	5.10	(6,1)	9.82
(5,2)	6.09	(4,3)	1.21	(3,4)	9.92
(2,5)	2.62	(1,6)	9.69	(0,7)	5.95

Table 4.5: Data $\boldsymbol{y}$ used in Example 4.6.1.

Table 4.6: Ten best-fitting real models that solve (4.8) for the problem presented in Example 4.6.1.

$\|\tilde{\boldsymbol{y}}\|^2$	$\boldsymbol{\lambda}_1$	$\boldsymbol{\lambda}_2$	x_1	x_2
232.14	$(1.062, 1.040)$	$(-0.841, -1.035)$	4.154	-1.219
241.90	$(1.072, 1.076)$	$(-6.557, 8.087)$	3.838	-1.324×10^{-6}
246.17	$(1.086, 1.064)$	$(-1.229, 0.565)$	3.794	5.673×10^{-1}
254.23	$(0.504, 0.721)$	$(0.977, 0.987)$	-5.300	6.850
256.34	$(1.838, -0.545)$	$(1.096, 1.052)$	-4.268×10^{-2}	3.897
261.33	$(-19.10, 6.251)$	$(1.087, 1.061)$	2.482×10^{-9}	3.841
262.55	$(1.069, 1.055)$	$(-0.436, -0.214)$	4.045	-1.799
265.12	$(1.075, 1.028)$	$(1.063, 1.810)$	3.997	3.049×10^{-2}
265.15	$(1.412, 2.672)$	$(1.073, 1.037)$	1.702×10^{-3}	4.036
265.47	$(1.855, 1.010)$	$(1.102, 1.059)$	-2.486×10^{-2}	3.855

may still be an uneven number of total solutions, as is the case here, since cases where $(\boldsymbol{\lambda}_1, x_1) = (\boldsymbol{\lambda}_2, x_2)$ cannot be paired with another solution.

While the corresponding kernel formulation (4.11) cannot be readily tackled by HomotopyContinuation.jl due to the large number of variables, as briefly discussed in Section 4.5, the claims of Theorem 4.3.2 can be verified by checking whether the solutions in Table 4.6 also solve this formulation. Similarly to the discussion in Section 4.3.2, once $\boldsymbol{\Lambda}$ is known, (4.11) boils down to solving a linear system for $\boldsymbol{g}$. Obtaining $\boldsymbol{g}$ in such a way for each of the models in Table 4.6, the corresponding residual norms of (4.11) can be obtained, the largest of

Table 4.7: Best-fitting real models that solve (4.8) as x_3 increases for the problems presented in Example 4.6.2.

x_3	$\|\tilde{\boldsymbol{y}}\|^2$	$\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2$	x_1, x_2
0.1	5.23×10^{-4}	$(0.985 \mp \mathrm{i}0.159, 0.947 \mp \mathrm{i}0.304)$	$1.047 \pm \mathrm{i}0.004$
1	3.36×10^{-2}	$(0.949 \pm \mathrm{i}0.150, 0.935 \pm \mathrm{i}0.260)$	$1.479 \mp \mathrm{i}0.011$
10	3.17×10^{-1}	$(0.868 \pm \mathrm{i}0.085, 0.940 \pm \mathrm{i}0.121)$	$5.969 \pm \mathrm{i}1.621$
100	4.74×10^{-1}	$(0.821, 0.922), (0.866, 0.982)$	$137.8, -35.86$

which is 2.388×10^{-10}.

Finally, consider the problem of fitting a first-order model to the same data. This results in a best-fitting real model of $\boldsymbol{\lambda}_1 = (1.075, 1.062), x_1 = 3.906$, with a fit of $\|\tilde{\boldsymbol{y}}\|^2 = 266.58$. It can be verified that this model also appears as a lower-order solution to the original second-order equations (4.8). Indeed, plugging in the second-order model with $\boldsymbol{\lambda}_2 = \boldsymbol{\lambda}_1$ and $x_2 = 0$ into said equations leads to a residual norm of 6.173×10^{-13}. Notice that increasing the model order causes the residual to strictly decrease. Indeed, as argued in the proof of Theorem 4.4.2, whenever $\tilde{\boldsymbol{y}} \neq 0$, there always exists an additional eigentuple that results in a better fit, causing the locally minimal real models to be models of precisely order n. ◇

Example 4.6.2. Consider data of the form $y_{\boldsymbol{\kappa}} = (0.99 + 0.16\mathrm{i}, 0.95 + 0.31\mathrm{i})^{\boldsymbol{\kappa}} + (0.99 - 0.16\mathrm{i}, 0.95 - 0.31\mathrm{i})^{\boldsymbol{\kappa}} + x_3(0.8, 0.9)^{\boldsymbol{\kappa}}$ for $\boldsymbol{\kappa} \in \mathbb{N}_0^2, \|\boldsymbol{\kappa}\|_1 \leq 7$ and for varying values of $x_1 \in \{0.1, 1, 10, 100\}$. We now fit a second-order $n = 2$ model to this data. The results of solving (4.8) and filtering out the best-fitting real models for each x_3 are presented in Table 4.7.

Remark that $\boldsymbol{y}$ is model-compliant with a third-order model so that for $x_3 = 0.1$, the best-fitting second-order model approximates the 'dominant' eigentuples $(0.99 \pm 0.16\mathrm{i}, 0.95 \pm +0.31\mathrm{i})$ quite well. As x_3 increases, the contributions of these eigentuples become less dominant, as reflected by the fitted 2D eigentuples straying further. For $x_3 = 100$ the third eigentuple $(0.8, 0.9)$ becomes dominant and is decently approximated by the fitted eigentuples. ◇

Example 4.6.3. Consider the 35-point data 3D data sequence $\forall \boldsymbol{\kappa} \in \mathbb{N}_0^m, \|\boldsymbol{\kappa}\|_1 \leq 4 : \boldsymbol{y} = 0.76(0.95, 0.12, 0.82,)^{\boldsymbol{\kappa}} + 0.92(0.07, 0.57, 0.69)^{\boldsymbol{\kappa}}$. Solving the corresponding equations (4.8) using $n = 1$ results in four real models, as shown in Table 4.8. As discussed in Section 4.4, these models include all real local minima, of which $\boldsymbol{\lambda}_1 = (0.681, 0.307, 0.763)$ achieves the best fit. ◇

Table 4.8: All real models that solve (4.8) for the problem in Example 4.6.3.

$\|\tilde{\boldsymbol{y}}\|^2$	$\boldsymbol{\lambda}_1$	x_1
0.495	$(0.681, 0.307, 0.763)$	1.561
10.39	$(-1.132, 0.117, -0.909)$	0.126
10.60	$(1.232, -0.979, -2.175)$	8.412×10^{-3}
10.68	$(-1.127, -4.442, 1.405)$	0.127×10^{-4}

4.7 Conclusion and future work

This work presents previously unknown properties of the local minima of the least-squares misfit identification problem for multidimensional, autonomous, scalar, linear, shift-invariant models, showing that they are also stationary points of a relaxed misfit problem and that their dynamics cannot exhibit a 'lower model order'. With these observations in mind, a novel polynomial characterization of the local minima is obtained through careful manipulation of the FONC. This leads to a multidimensional extension of the Walsh-like interpolation property previously observed for LTI systems [36, 157]. By duality between the kernel and image representation, this then leads to an alternative characterization based on the kernel representation, akin to [102, 104].

These polynomial characterizations allow the local minima to be found through off-the-shelf polynomial root-finding techniques, extending the approach for globally optimal identification pioneered in [36] to the multidimensional case. Moreover, we provide a rigorous theoretical footing for the assumptions made in the latter paper, such as the isolatedness of the solutions. Due to the large computational overhead entailed by this strife for global optimality, the contributions are of a predominantly theoretical nature, offering new insights into the problem's stationary points. This may, for example, be employed to study other, more heuristic algorithms for the problem at hand or to further tailor such methods to the problem, as was done for the LTI case in [206].

As this demonstrates that global optimality can be achieved for certain classes of models, we aim to further extend this approach to models including inputs and to models with multiple outputs in the future, as well as to nonlinear polynomial model classes. Additionally, it may be interesting to investigate alternative parameterizations for autonomous, scalar, linear, shift-invariant models, to eliminate the issues presented by *lower-order* solutions in the current approach.

Computationally, the structure of the polynomial equations may be leveraged to reduce the computational load, most notably, the symmetry of the equations

w.r.t. the eigentuples can most likely be exploited for this purpose. Applying algorithms that can tackle multiparameter eigenvalue problems with an infinite solution set may also result in a significant speedup, since the polynomial characterizations can be easily rephrased as such. As MEPs exploit the linearity of the problem w.r.t. the auxiliary variables, this may be especially interesting to reduce the dimensionality of the kernel-based equation (4.11). Alternatively, one could aim to reformulate these MEPs to eliminate the observed positive dimensionality.

Part II

Algorithms

Chapter 5

Exploiting the Block-Macaulay Matrix Structure

Chapter abstract

Novel algorithms are presented to compute null and row space bases for the block-Macaulay matrix, a class of matrices that arises in numerical root-finding approaches for polynomials and multiparameter eigenvalue problems. Moreover, computing such bases is often the bottleneck in these root-finding methods. The proposed approach leverages the shift-invariant structure of this class of matrices to reuse intermediately computed bases. For any block-row decomposition, these 'shifted' smaller bases can be joined using a structured intersection subproblem for the null space basis, and a direct sum problem for the row space basis. We propose a particular block-row decomposition that leads to recursive algorithms to compute such bases. Moreover, these algorithms avoid the explicit construction of the often large Macaulay matrix itself. Finally, using numerical experiments, the algorithm is compared to the state-of-the-art methods to compute the block-Macaulay matrix null space.

Remark: It has come to our attention that other authors have recently published a very similar approach that relies on many of the same principles in [195]. We emphasize that this chapter was independent work and we were not aware of said author's work at the time of research. To strengthen this claim, we mention that our work has been presented in a poster at the ERNSI 2024 conference before [187].

Big picture

The current chapter can be thought of as providing a new, structure-exploiting algorithm to obtain an image representation from a kernel representation of a model, as discussed in Chapter 2. This is also an important substep in root-finding techniques for MEPs and polynomial equations, and it is often the bottleneck in these solvers. As such, the presented algorithms can be used to tackle root-finding problems arising in globally optimal system identification problems for LTI models, e.g., see [9, 36, 104, 189]. Unfortunately, due to the infinite number of common roots, the algorithm is not applicable to the misfit problem presented in Chapter 4 for dimensions higher than one. Of course, other polynomial optimization problems can be tackled in similar fashion, where the presented algorithm may serve as a root-finding tool as well.

5.1 Introduction

Multivariate polynomials have garnered substantial attention throughout the years, as they are useful in many applications. The theoretical properties of this class of objects have also been studied extensively, most notably in the field of algebraic geometry [31, 32]. Algebraically manipulating the involved polynomials requires qualitative computational tools, either symbolic or numerical in nature, depending on the application. For symbolic polynomials, Gröbner bases are popular computational tools [32], most notably variations on the F4 and F5 algorithms [53, 54]. Here, applications include cryptography [94] and error correcting codes [31], among others.

In this work, we focus on polynomials with numerical coefficients, the algorithms for which require adequate adaptations to maintain their performance in the presence of numerical errors. To this end, border bases were developed, which are more lenient than Gröbner bases, allowing a numerically more appropriate choice of basis elements [92, 133, 137, 169, 181]. Techniques such as truncated normal forms [177, 178] and the Macaulay framework [12, 44] let well-established numerical linear algebra (NLA) methods pick the best basis elements. The latter framework relies on the singular value decomposition (SVD) to manipulate a large so-called *Macaulay matrix* and is the main focus of the current work.

The Macaulay framework has been extended to the block-Macaulay framework to deal with so-called (rectangular) *multiparameter eigenvalue problems* (MEPs) [188, 189, 191]. This class of polynomial problems generalizes multivariate

polynomials by allowing matrix rather than scalar coefficients. This allows one to take advantage of the linear structure of some of the variables, i.e., the eigenvector, which often speeds up computations. For the sake of generality, the results in this work are thus presented for the block-Macaulay framework. MEPS can be considered a generalization of the polynomial eigenvalue problem as well [35].

While MEPs can serve many purposes, many of their applications revolve around finding the common roots or *eigentuples*. For example: polynomial optimization problems, most notably globally optimal system identification and model order reduction [9, 36, 102, 103, 189] and spectral analysis for PDEs [5, 148, 150, 180], among others. As such, this work concentrates on such root-finding approaches. More precisely, those relying on the null space of a block-Macaulay matrix. Other MEP root-finding approaches, such as those based on the different, but related *square MEP problem* [80, 81], and homotopy continuation methods for polynomial root finding [24, 192] will not be considered here.

The central object of the block-Macaulay framework is the highly structured *block-Macaulay matrix*. Computing the null space of this potentially large matrix most often remains the bottleneck in root-finding algorithms, even though various methods have been developed to lighten this computational load: Methods allowing an incremental degree-wise increase in size of the null space basis, while leveraging the zero structure of the matrix [11, 191] as well as those leveraging the multilevel Toeplitz structure of the matrix [66]. Alternatively, the sparse nature of the monomial support of the polynomials themselves can be exploited to reduce the size of the Macaulay matrix itself [16]. In hopes of further mitigating this bottleneck, this work proposes a novel strategy to compute such a null space basis by exploiting the shift-invariant structure of this matrix. This results in a recursive approach that relies on taking intersections of previously computed null space bases. Moreover, this methodology is also adapted to compute a basis for the row space of such a matrix, where the intersection is replaced by a direct sum. The row space may also be used for root-finding purposes, and it is of interest for some applications other than root finding, such as elimination [10].

The remainder of this chapter is organized as follows: in Section 5.2, the block-Macaulay framework is briefly introduced and its shift-invariant structure is more closely examined. Next, a novel methodology for building its null space recursively based on intersections is presented in Section 5.3. The contributions here are threefold: Firstly, we present a methodology that extends the shift-invariance property to the null space and uses this property to compute the null space of a Macaulay-like matrix as an intersection of the null spaces associated with its Macaulay-like submatrices. Secondly, a specific row decomposition for this methodology is presented, minimizing duplicate rows. Thirdly, an approach

to exploit the structure in the resulting intersections is presented. This leads to an algorithm that builds the null space of a block-Macaulay matrix, while entirely avoiding the construction of the Macaulay matrix itself.

Since root finding using the columns of a Macaulay matrix is possible as well [190] and due to the close connection of both approaches, a similar method to build an orthogonal basis for the row space of a block-Macaulay matrix is outlined in Section 5.4. In Section 5.5, an experiment-based comparison to state-of-the-art methods is presented. Finally, a general conclusion and an outlook for future research paths are given in 5.6.

5.2 The block-Macaulay framework

As previously stated, the block-Macaulay framework provides numerical linear-algebra-based computational tools for MEPs and multivariate polynomials. In this work, we focus on the former type of problem, as multivariate polynomials can be viewed as special cases of an MEP [188]. In general, an MEP is the problem of finding the set of parameters $\boldsymbol{x}$ such that some polynomial matrix $\sum_{\boldsymbol{\alpha}\in\mathcal{S}_{\boldsymbol{L}}} \boldsymbol{L}_{\boldsymbol{\alpha}}\boldsymbol{x}^{\boldsymbol{\alpha}} = \boldsymbol{L}(\boldsymbol{x}) \in \mathbb{C}^{p\times q}[\boldsymbol{x}]$ becomes rank-deficient:

$$\text{find } \boldsymbol{x}, \boldsymbol{v} : \boldsymbol{L}(\boldsymbol{x})\boldsymbol{v} = \boldsymbol{0} \quad \text{where } \|\boldsymbol{v}\|_2 = 1, \boldsymbol{v} \in \mathbb{C}^q.$$

Notice that, due to the linearity of the eigenvector $\boldsymbol{v}$, once a suitable eigentuple $\boldsymbol{x}$ is found, $\boldsymbol{v}$ follows directly by solving a null space problem. The linear nature of the variables $\boldsymbol{v}$ can be exploited by considering them as auxiliary variables in a sense, reducing the problem complexity. At the heart of the block-Macaulay framework lies the highly structured Block-Macaulay matrix, which is first introduced in the next subsection, after which its use in root finding and its structure are considered more closely.

5.2.1 The block-Macaulay matrix

We start by introducing the block-Macaulay matrix by means of Example 5.2.1, before resorting to a formal treatment of the matter.

Example 5.2.1. Consider a first-order two-parameter eigenvalue problem, where the coefficient matrices are indexed by their corresponding multi-exponent:

$$\boldsymbol{L}(x, y)\boldsymbol{v} = (\boldsymbol{L}_{(0,0)} + \boldsymbol{L}_{(1,0)}x + \boldsymbol{L}_{(0,1)}y)\boldsymbol{v} = \boldsymbol{0}. \tag{5.1}$$

The associated block-Macaulay of degree two matrix is then:

$$\boldsymbol{M} = \begin{array}{r} \\ \boldsymbol{L}(x,y) \\ x\boldsymbol{L}(x,y) \\ y\boldsymbol{L}(x,y) \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & x & y & x^2 & xy & y^2 \end{array} \\ \begin{bmatrix} \boldsymbol{L}_{(0,0)} & \boldsymbol{L}_{(1,0)} & \boldsymbol{L}_{(0,1)} & & & \\ & \boldsymbol{L}_{(0,0)} & & \boldsymbol{L}_{(1,0)} & \boldsymbol{L}_{(0,1)} & \\ & & \boldsymbol{L}_{(0,0)} & & \boldsymbol{L}_{(1,0)} & \boldsymbol{L}_{(0,1)} \end{bmatrix} \end{array}.$$

Notice that each block row of $\boldsymbol{M}$ corresponds to a matrix-based representation of a 'shift', i.e., a monomial multiple of $\boldsymbol{L}(x,y)$.

Now define the vector: $\boldsymbol{z} = \begin{bmatrix} \boldsymbol{v}^\mathsf{T} & x\boldsymbol{v}^\mathsf{T} & y\boldsymbol{v}^\mathsf{T} & x^2\boldsymbol{v}^\mathsf{T} & xy\boldsymbol{v}^\mathsf{T} & y^2\boldsymbol{v}^\mathsf{T} \end{bmatrix}^\mathsf{T}$, a so-called multivariate Vandermonde vector. Notice that $\boldsymbol{z}$ essentially acts as an evaluation operation in the point $(x, y, \boldsymbol{v})$ w.r.t. the Macaulay matrix:

$$\boldsymbol{M}\boldsymbol{z} = \begin{bmatrix} \boldsymbol{L}(x,y)\boldsymbol{v} \\ x\boldsymbol{L}(x,y)\boldsymbol{v} \\ y\boldsymbol{L}(x,y)\boldsymbol{v} \end{bmatrix}.$$

As a consequence, $\boldsymbol{M}\boldsymbol{z} = \boldsymbol{0}$ whenever $(x, y, \boldsymbol{v})$ forms a solutions to the MEP (5.1), so that a corresponding multivariate Vandermonde vector appears the null space of $\boldsymbol{M}$ for each solution. Moreover, it has been shown that, given that all shifts $x^k y^l \boldsymbol{L}(x,y)$ up to a sufficiently high total degree $d = k + l$ are represented in the matrix, a so-called 'gap' appears in the null space, signaling that the matrix is large enough to extract the multiplication maps [12, 188].

The relation between the null space and the solutions then allows one to extract the eigentuples (x, y) by setting up a coupled eigenvalue problem through shift equations and subsequently solving for said eigenvalues [12, 44, 188]. The intricacies of this process will be explained later on in this section. ◇

Example 5.2.1 illustrates the intuitive idea behind block-Macaulay matrices: It contains vector-based representations of 'shifts' of some matrix polynomial $\boldsymbol{L}(\boldsymbol{x})$, i.e., $\boldsymbol{L}(\boldsymbol{x})$ multiplied by monomials in $\boldsymbol{x}$, resulting in a highly structured matrix. More precisely, its entries, representing the coefficients of $\boldsymbol{L}(\boldsymbol{x})$, are the same in each block row, but their placement within the block columns differs, fully defining the matrix. As will be illustrated in this section, the considered set of row shifts fully defines the nonzero columns of the Macaulay matrix. This work focuses on exploiting the structured nature of this matrix to compute its null space in an alternative way. First, however, some additional notation is introduced, and some more background is provided on the block-Macaulay matrix and its use.

In order to explicitly write down a block-Macaulay matrix, the monomials acting as column and row indices have to be ordered somehow. There exist many

ways to order monomials, though we consider the graded inverse lexicographic ordering [188].

For the purposes of this work, however, it is conceptually more convenient to use the monomials themselves, or rather, their multi-exponents, as indices, rather than first ordering them. Indeed, note that, any monomial $\boldsymbol{x}^{\boldsymbol{\alpha}} = \prod_{i=0}^{m} x_i^{\alpha_i}$ can be uniquely represented using their multi-exponent $\boldsymbol{\alpha} \in \mathbb{Z}^m$. Given a polynomial matrix $\boldsymbol{L}(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha} \in \mathcal{S}_{\boldsymbol{L}}} \boldsymbol{L}_{\boldsymbol{\alpha}} \boldsymbol{x}^{\boldsymbol{\alpha}} \in \mathbb{C}^{p \times q}[\boldsymbol{x}]$, the finite set[1] $\mathcal{S}_{\boldsymbol{L}} \subset \mathbb{N}_0^m$ is called the *support* of $\boldsymbol{L}(\boldsymbol{x})$ and the coefficient of $\boldsymbol{L}(\boldsymbol{x})$ associated with $\boldsymbol{x}^{\boldsymbol{\alpha}}$ will be denoted as $\boldsymbol{L}_{\boldsymbol{\alpha}}$. Note that the support $\mathcal{S}_{\boldsymbol{L}}$ is the set of multi-exponents of the monomials with which there is a nonzero coefficient associated in $\boldsymbol{L}(\boldsymbol{x})$, so that $\forall \boldsymbol{\alpha} \notin \mathcal{S}_{\boldsymbol{L}} : \boldsymbol{L}_{\boldsymbol{\alpha}} = \boldsymbol{0}$. Moreover, notice that the support of $\boldsymbol{x}^{\boldsymbol{\sigma}} \boldsymbol{L}(\boldsymbol{x})$ is $\{\boldsymbol{\sigma}\} \oplus \mathcal{S}_{\boldsymbol{L}}$, where $\oplus$ denotes the Minkowski sum[2]. Using this notation, we can now compactly define the Macaulay matrix as a map from two multi-indices to some matrix-valued coefficient.

Definition 5.2.1 (block-Macaulay matrix)**.** Given a polynomial matrix $\boldsymbol{L}(\boldsymbol{x}) \in \mathbb{C}^{p \times q}[\boldsymbol{x}]$ in the variables $\boldsymbol{x} \in \mathbb{C}^m$. Let $\mathcal{R}$ and $\mathcal{C}$ be finite subsets of $\mathbb{N}_0^m$, where $\mathcal{C} \subseteq \mathcal{R} \oplus \mathcal{S}_{\boldsymbol{L}}$. The associated block-Macaulay matrix with row multi-indices $\mathcal{R}$ and column multi-indices $\mathcal{C}$ is then the following map:

$$\boldsymbol{M}^{\mathcal{R},\mathcal{C}} : \mathcal{R} \times \mathcal{C} \Rightarrow \mathbb{C}^{p \times q}; (\boldsymbol{\rho}, \boldsymbol{\gamma}) \mapsto \boldsymbol{L}_{\boldsymbol{\gamma}-\boldsymbol{\rho}}.$$

To highlight that this is essentially a matrix, we use subscripts to index the elements, i.e., $\boldsymbol{M}^{\mathcal{R},\mathcal{C}}_{\boldsymbol{\rho},\boldsymbol{\gamma}} = \boldsymbol{M}^{\mathcal{R},\mathcal{C}}(\boldsymbol{\rho}, \boldsymbol{\gamma})$. △

Definition 5.2.1 is slightly broader than considered in the literature, where $\mathcal{R}$ is most often chosen to contain all multi-exponents up to some fixed degree [12, 44, 188]. This particular choice of $\mathcal{R}$ serves an important purpose when solving MEPs, as was highlighted earlier on in Example 5.2.1. Even though this chapter also revolves around the application to polynomial root finding, and, as such, block-Macaulay matrices with such a conventional choice of row indices, the presented notation allows us to easily talk about specific submatrices. For convenience, these submatrices are also referred to as block-Macaulay matrices. Since the current work deals with different such Macaulay submatrices, each all associated with the same matrix polynomial $\boldsymbol{L}(\boldsymbol{x})$, the dependence on $\boldsymbol{L}(\boldsymbol{x})$ will not be made explicit throughout to keep the notation more concise.

Ordering both finite sets $\mathcal{R}$ and $\mathcal{C}$ allows this map to be written out in a matrix format. Note that while the set of column indices $\mathcal{C}$ is required to specify the number of columns of the matrix, the only nonzero columns correspond to the

[1] $\mathbb{N}_0^m$ denotes the natural numbers including zero.

[2] The Minkowski sum of two sets corresponds to the set of all possible sums of set elements: $A \oplus B = \{a + b \mid a \in A, b \in B\}$.

column indices in $\mathcal{R} \oplus \mathcal{S}_{\boldsymbol{L}}$, as the support $\mathcal{S}_{\boldsymbol{L}}$ describes all monomials associated with nonzero coefficients. These concepts are now briefly illustrated in Example 5.2.2 below.

Example 5.2.2. Let $\boldsymbol{x} = (x, y)$ and consider the following polynomial matrix:

$$\begin{aligned} \boldsymbol{L}(x,y) &= \begin{bmatrix}1\\6\end{bmatrix} x + \begin{bmatrix}2\\7\end{bmatrix} y + \begin{bmatrix}3\\8\end{bmatrix} x^2 + \begin{bmatrix}4\\9\end{bmatrix} xy + \begin{bmatrix}5\\10\end{bmatrix} y^2 \\ &= \begin{bmatrix}1\\6\end{bmatrix} \boldsymbol{x}^{(1,0)} + \begin{bmatrix}2\\7\end{bmatrix} \boldsymbol{x}^{(0,1)} + \begin{bmatrix}3\\8\end{bmatrix} \boldsymbol{x}^{(2,0)} + \begin{bmatrix}4\\9\end{bmatrix} \boldsymbol{x}^{(1,1)} + \begin{bmatrix}5\\10\end{bmatrix} \boldsymbol{x}^{(0,2)}. \end{aligned} \tag{5.2}$$

Notice that the monomial support is characterized by the set of multi-exponents $\mathcal{S}_{\boldsymbol{L}} = \{(1,0), (0,1), (2,0), (1,1), (0,2)\}$. Multi-exponents associated with a zero coefficient are not part of the support, e.g., note that $\boldsymbol{L}_{(0,0)} = \boldsymbol{0}$. Now consider the sets of row multi-indices $\mathcal{R} = \{(0,0), (1,0), (0,1)\}$ and column multi-indices $\mathcal{C} = \{\boldsymbol{\gamma} \in \mathbb{N}_0^2 \mid \|\boldsymbol{\gamma}\|_1 \leq 3\}$. Definition 5.2.1 now defines a map from a pair of multi-indices onto some matrix-valued element, e.g., consider the indices $(1,0) \in \mathcal{R}$ and $(2,0) \in \mathcal{C}$, then $\boldsymbol{M}^{\mathcal{R},\mathcal{C}}_{(1,0),(2,0)} = \boldsymbol{L}_{(2,0)-(1,0)} = \begin{bmatrix}1 & 6\end{bmatrix}^{\mathsf{T}}$.

After ordering these index sets, this map can be summarized in the following block-matrix:

$$\boldsymbol{M}^{\mathcal{R},\mathcal{C}} = \overbrace{\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & & & & \\ 6 & 7 & 8 & 9 & 10 & & & & \\ & & 1 & 2 & & 3 & 4 & 5 & \\ & & 6 & 7 & & 8 & 9 & 10 & \\ & & & 1 & 2 & & 3 & 4 & 5 \\ & & & 6 & 7 & & 8 & 9 & 10 \end{bmatrix}}^{\mathcal{C}} \begin{matrix} (0,0) \\ \\ (1,0) \\ \\ (0,1) \\ \\ \end{matrix} \Bigg\} \mathcal{R}.$$

$$\begin{matrix} (0,0) & (1,0) & (0,1) & (2,0) & (1,1) & (0,2) & (3,0) & (2,1) & (1,2) & (0,3) \end{matrix}$$

Of course, permuting the rows or columns does not affect the underlying map.

In essence, this block-Macaulay matrix is a matrix-based representation of the set of polynomials $\mathcal{R} \cdot \boldsymbol{L}(\boldsymbol{x}) = \{\boldsymbol{x}^{\boldsymbol{\rho}} \boldsymbol{L}(\boldsymbol{x}) \mid \boldsymbol{\rho} \in \mathcal{R}\} = \{\boldsymbol{L}(\boldsymbol{x}), x\boldsymbol{L}(\boldsymbol{x}), y\boldsymbol{L}(\boldsymbol{x})\}$. Using the notation of Definition 5.2.1, we index the block row associated with $\boldsymbol{x}^{\boldsymbol{\rho}} \boldsymbol{L}(\boldsymbol{x})$ using its multi-exponent $\boldsymbol{\rho}$, and, similarly, the block column associated with the monomial $\boldsymbol{x}^{\boldsymbol{\gamma}}$ is indexed using the multi-exponents $\boldsymbol{\gamma}$. Remark that the monomial support associated with $\boldsymbol{x}^{\boldsymbol{\rho}} \boldsymbol{L}(\boldsymbol{x})$ is, $\{\boldsymbol{\rho}\} \oplus \mathcal{S}_{\boldsymbol{L}}$, such that the full support of the set of matrix polynomials $\mathcal{R} \cdot \boldsymbol{L}(\boldsymbol{x})$ is $\mathcal{R} \oplus \mathcal{S}_{\boldsymbol{L}} = \mathcal{C} \backslash \{(0,0)\}$, as is evident from the nonzero part of $\boldsymbol{M}^{\mathcal{R},\mathcal{C}}$. As such, there is a zero column associated with the column index $(0,0)$, since $\forall \boldsymbol{\rho} \in \mathcal{R} : \boldsymbol{L}_{(0,0)} = \boldsymbol{0}$. ◇

5.2.2 Null space and polynomial root finding

We now look into the null space of the block-Macaulay matrix and its use in root-finding. To further avoid dealing with the specifics of ordering the monomials, it is natural to extend multi-indexing to the null space of block-Macaulay matrices. Indeed, a basis for the null space can be represented through a multi-indexed array, though with a natural number as column index, i.e., it is a map: $\boldsymbol{N} : \mathcal{C} \times \{1, \ldots, n\} \to \mathbb{C}^1; (\boldsymbol{\gamma}, i) \mapsto \boldsymbol{N}(\boldsymbol{\gamma}, i); n \in \mathbb{N}$. A product between two such 'matrices' is then readily defined element-wise as $\left(\boldsymbol{M}^{\mathcal{R},\mathcal{C}}\boldsymbol{N}\right)_{\boldsymbol{\rho},i} = \sum_{\boldsymbol{\gamma}\in\mathcal{C}} \boldsymbol{M}^{\mathcal{R},\mathcal{C}}_{\boldsymbol{\rho},\boldsymbol{\gamma}} \boldsymbol{N}_{\boldsymbol{\gamma},i}$. Of course, when $\boldsymbol{N}$ constitutes a basis for $\operatorname{null} \boldsymbol{M}^{\mathcal{R},\mathcal{C}}$ this product is zero everywhere, as well as $n = \operatorname{nullity} \boldsymbol{M}^{\mathcal{R},\mathcal{C}}$ and $\nexists \boldsymbol{z} \in \mathbb{C}^n : \boldsymbol{N}\boldsymbol{z} = \boldsymbol{0}$. A basis for the null space of $\boldsymbol{M}^{\mathcal{R}}$ will be denoted as $\boldsymbol{N}^{\mathcal{R}}$, and, similarly, a basis for the null space of $\boldsymbol{M}^{\mathcal{R},\mathcal{C}}$ is denoted with $\boldsymbol{N}^{\mathcal{R},\mathcal{C}}$ from here on.

As already highlighted in Example 5.2.1, the null space of a block-Macaulay matrix is of major importance to its main application: solving MEPs. Indeed, if the associated MEP has a finite number of solutions (including those at infinity), it has been observed that the nullity of a block-Macaulay matrix stabilizes to said number of solutions, including multiplicities, as the total degree of the matrix is increased [12, 188]. For equations with an infinite number of solutions at infinity, the situation is slightly different, though the appearance of a 'gap' in the null space allows the solutions at infinity to be filtered out using linear algebra techniques. The result is then utilized to construct a set of coupled eigenvalue problems that reveal the eigentuples of the MEP.

To not stray too far from the main message, we only briefly sketch one such procedure and the importance of the null space through Example 5.2.3. Approaches that determine the solutions at infinity also exist [16], as well as slight variations on the illustrated approach, e.g., where the coupled eigenvalue problem is tackled through a tensor decomposition [184, 185, 196], as well as a row-space based approach [43, 190]. For a more complete treatment of the matter, we refer the reader to [169, 176, 188].

Example 5.2.3. Reconsider $\boldsymbol{L}(\boldsymbol{x}), \mathcal{R}, \mathcal{C}$ and $\boldsymbol{M}^{\mathcal{R},\mathcal{C}}$ from Example 5.2.2. A basis for the corresponding null space of $\boldsymbol{M}^{\mathcal{R},\mathcal{C}}$ is then the following matrix:

$$\boldsymbol{N}^{\mathcal{R},\mathcal{C}} = \left[\begin{array}{rrrr|l} 1 & 0 & 0 & 0 & (0,0) \\ 0 & 1 & 0 & 0 & (1,0) \\ 0 & 0 & 1 & 0 & (0,1) \\ 0 & 0 & 0 & 1 & (2,0) \\ 0 & -4 & -3 & -2 & (1,1) \\ 0 & 3 & 2 & 1 & (0,2) \\ \hline 0 & 3 & 2 & -\frac{2}{3} & (3,0) \\ 0 & 6 & 5 & \frac{10}{3} & (2,2) \\ 0 & -5 & -4 & -\frac{5}{3} & (1,2) \\ 0 & 0 & 0 & -\frac{2}{3} & (0,3) \end{array}\right].$$

Notice that the first four rows are linearly independent. Consequently, rows corresponding to the indices third-degree exponents $\{(3,0),(2,1),(1,2),(0,3)\}$ are linearly dependent w.r.t. the preceding rows. Whenever all the (block) rows of a whole degree block are linearly dependent on their lower-degree predecessors, this is called a 'gap' in the null space. This signals that we may proceed to the next steps of the algorithm. As a side note, when linearly independent rows occur after the gap, one can filter out solutions at infinity by first using a column compression [44, 188].

Since this is not the case, we set up the shift equations using the rows of $\boldsymbol{N}^{\mathcal{R},\mathcal{C}}$. Remember from Example 5.2.1 that any multivariate Vandermonde vector associated with a solution (x, y) to (5.2) lies in range $\boldsymbol{N}^{\mathcal{R},\mathcal{C}}$ ($\boldsymbol{v} = 1$ in this case). As such, an alternative null space basis exists, consisting of such Vandermonde vectors, which we call $\boldsymbol{Z}^{\mathcal{R},\mathcal{C}}$. For an appropriate selection of the rows of this basis $\boldsymbol{Z}^{\mathcal{R},\mathcal{C}}$, a so-called multiplication map $\boldsymbol{D}_1$ exists such that, by abuse of the multi-index notation, the following 'shift equations' holds[3] (a

[3] ':' denotes 'all indices'.

similar construction can be made to define $\boldsymbol{D}_2$):

$$\begin{aligned}\boldsymbol{Z}^{\mathcal{R},\mathcal{C}}_{\|\boldsymbol{\gamma}\|_1\leq 2,:}\boldsymbol{D}_1 &= \begin{bmatrix} 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ x_1^2 & x_2^2 & x_3^2 & x_4^2 \\ x_1y_1 & x_2y_2 & x_3y_3 & x_4y_4 \\ y_1^2 & y_2^2 & y_3^2 & y_4^2 \end{bmatrix} \begin{bmatrix} x_1 & & & \\ & x_2 & & \\ & & x_3 & \\ & & & x_4 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 1 & 1 & 1 \\ x_1^2 & x_2^2 & x_3^2 & x_4^2 \\ x_1y_1 & x_2^2y_2 & x_3y_3 & x_4y_4 \\ x_1^3 & x_2^3 & x_3^3 & x_4^3 \\ x_1^2y_1 & x_2^2y_2 & x_3^2y_3 & x_4^2y_4 \\ x_1y_1^2 & x_2y_2^2 & x_3y_3^2 & x_4y_4^2 \end{bmatrix} = \boldsymbol{Z}^{\mathcal{R},\mathcal{C}}_{\gamma_1\geq 1,:}.\end{aligned} \tag{5.3}$$

Now, since range $\boldsymbol{Z}^{\mathcal{R},\mathcal{C}} = \text{range}\,\boldsymbol{N}^{\mathcal{R},\mathcal{C}}$, it holds that there exists an invertible $\boldsymbol{P}$ so that range $\boldsymbol{Z}^{\mathcal{R},\mathcal{C}} = \text{range}\,\boldsymbol{N}^{\mathcal{R},\mathcal{C}}\boldsymbol{P}$. As such, we can rewrite the shift equation (5.3) to the form below and introduce the multiplication maps $\boldsymbol{A}_1 = \boldsymbol{P}\boldsymbol{D}_1\boldsymbol{P}^{-1}$ and, similarly, $\boldsymbol{A}_2 = \boldsymbol{P}\boldsymbol{D}_2\boldsymbol{P}^{-1}$:

$$\begin{aligned}\boldsymbol{N}^{\mathcal{R},\mathcal{C}}_{\|\boldsymbol{\gamma}\|_1\leq 2,:}\boldsymbol{A}_1 &= \begin{matrix}(0,0)\\(1,0)\\(0,1)\\(2,0)\\(1,1)\\(0,2)\end{matrix}\!\!\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -4 & -3 & -2 \\ 0 & 3 & 2 & 1 \end{bmatrix}\boldsymbol{A}_1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -4 & -3 & -2 \\ 0 & 3 & 2 & -\frac{2}{3} \\ 0 & 6 & 5 & \frac{10}{3} \\ 0 & -5 & -4 & -\frac{5}{3} \end{bmatrix}\!\!\begin{matrix}(1,0)\\(2,0)\\(1,1)\\(3,0)\\(2,1)\\(1,2)\end{matrix} \\ &= \boldsymbol{N}^{\mathcal{R},\mathcal{C}}_{\gamma_1\geq 1,:},\end{aligned}$$

$$\begin{aligned}\boldsymbol{N}^{\mathcal{R},\mathcal{C}}_{\|\boldsymbol{\gamma}\|_1\leq 2,:}\boldsymbol{A}_2 &= \begin{matrix}(0,0)\\(1,0)\\(0,1)\\(2,0)\\(1,1)\\(0,2)\end{matrix}\!\!\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -4 & -3 & -2 \\ 0 & 3 & 2 & 1 \end{bmatrix}\boldsymbol{A}_2 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & -4 & -3 & -2 \\ 0 & 3 & 2 & 1 \\ 0 & 6 & 5 & \frac{10}{3} \\ 0 & -5 & -4 & -\frac{5}{3} \\ 0 & 0 & 0 & -\frac{5}{3} \end{bmatrix}\!\!\begin{matrix}(0,1)\\(1,1)\\(0,2)\\(2,1)\\(1,2)\\(0,3)\end{matrix} \\ &= \boldsymbol{N}^{\mathcal{R},\mathcal{C}}_{\gamma_2\geq 1,:}.\end{aligned}$$

In both equations, the coefficient matrix is of full column rank, a feature guaranteed by the existence of the previously discussed 'gap'. Consequently, these equations can be set up and subsequently solved uniquely for the multiplication maps $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ using the known basis $\boldsymbol{N}^{\mathcal{R},\mathcal{C}}$. Notice that

these multiplication maps, like $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$, also perform multiplication with x and y, respectively, though in another polynomial basis [169]. This is also evident from their definition: they map the rows of $\boldsymbol{N}^{\mathcal{R},\mathcal{C}}$ corresponding to a monomial $\boldsymbol{x}^{\boldsymbol{\gamma}}$ onto the rows associated with $x(x,y)^{\boldsymbol{\gamma}} = \boldsymbol{x}^{\boldsymbol{\gamma}+(1,0)}$ and $y(x,y)^{\boldsymbol{\gamma}} = \boldsymbol{x}^{\boldsymbol{\gamma}+(0,1)}$, respectively.

Now, since $xy = yx$, and these maps express this multiplication, they must commute, i.e., $\boldsymbol{A}_1\boldsymbol{A}_2 = \boldsymbol{A}_2\boldsymbol{A}_1$. Indeed, solving the shift equations using a method of choice results in the following commuting matrices:

$$\boldsymbol{A}_1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -4 & -3 & -2 \\ 0 & 3 & 2 & -\frac{2}{3} \end{bmatrix}, \quad \boldsymbol{A}_2 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & -4 & -3 & -2 \\ 0 & 3 & 2 & 1 \\ 0 & 6 & 5 & \frac{10}{3} \end{bmatrix}.$$

Due to this property, they can be brought into a joint upper triangular format using the same orthogonal similarity transformation $\boldsymbol{Q}$ [29]. Applying this transformation then reveals the eigentuples (x, y) on the diagonals of the resulting matrices [169]. This leads to the following eigentuples $(0, 0), (-1.959 \pm 0.383i, 0.841 \pm 1.101i)$ and $(0.251, -0.348)$. Since $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ are similar to $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$, they share the same eigentuples and which thus solve (5.2). By the Bézout theorem [32], there are at most 4 solutions to this problem, so that these are all of them. ◇

Through previous efforts to avoid the explicit construction of the (block-) Macaulay matrix [11, 66, 191], the bottleneck in the approach outlined in Example 5.2.3 is most often computing the null space of the block-Macaulay matrix. This matrix tends to become very large due to the combinatorial growth of the number of rows and columns with the total degree of the matrix.

Due to this bottleneck in the outlined root-finding approach, it is of interest to further leverage the intricate structure present in the block-Macaulay matrix to compute a null space basis. Before the novel methods are presented, the structure of these matrices is further explored in the next subsection.

5.2.3 Shifting the row- and null space

It was already illustrated in the preceding subsections that the considered row multi-indices $\mathcal{R}$ almost fully define the structure of a block-Macaulay matrix. As such, together with $\boldsymbol{L}(\boldsymbol{x})$, $\mathcal{R}$ suffices to fully construct the nonzero part of any associated block-Macaulay matrix. Because of this, we will denote $\boldsymbol{M}^{\mathcal{R}}$ as $\boldsymbol{M}^{\mathcal{R},\mathcal{R}\oplus\mathcal{S}_L}$ for a more compact notation.

An important consequence of this intricate matrix structure is that there is a direct correspondence between different block-Macaulay matrices defined by a set of row multi-indices and 'shifts' thereof: By Definition 5.2.1, it holds that:

$$\forall \boldsymbol{\rho} \in \mathcal{R}, \boldsymbol{\gamma} \in \mathcal{R} \oplus \mathcal{S}_{\boldsymbol{L}} : \boldsymbol{M}^{\boldsymbol{\sigma}+\mathcal{R}}_{\boldsymbol{\sigma}+\boldsymbol{\rho},\boldsymbol{\sigma}+\boldsymbol{\gamma}} = \boldsymbol{M}^{\mathcal{R}}_{\boldsymbol{\rho},\boldsymbol{\gamma}},$$

where $\boldsymbol{\sigma} \in \mathbb{N}_0^m$ and we denote $\boldsymbol{\sigma} + \mathcal{R} = \{\boldsymbol{\sigma}\} \oplus \mathcal{R}$ for notational convenience.

Consequently, aside from the precise row and column indices, the matrix itself remains essentially unchanged by shifting the row indices $\mathcal{R}$. More precisely, the elements of the matrix $\boldsymbol{M}^{\mathcal{R}}_{\boldsymbol{\rho},\boldsymbol{\gamma}}$ are simply re-indexed by adding a constant offset $\boldsymbol{\sigma}$, without any further change to the structure of the matrix itself, as illustrated in Example 5.2.4 below. This 'shift-invariance' property of Macaulay matrices and the associated property of their null spaces forms the cornerstones of the proposed algorithm.

Example 5.2.4. Reconsider the matrix polynomial $\boldsymbol{L}(\boldsymbol{x})$ and the sets of multi-indices $\mathcal{R}$ and $\mathcal{C}$ from Example 5.2.2. $\boldsymbol{M}^{\mathcal{R}}$ then describes the nonzero columns of the associated block-Macaulay matrix, i.e., the columns with indices in $\mathcal{C}\backslash\{(0,0)\}$. Next, consider the Macaulay matrix associated with the row indices $(2,0)+\mathcal{R}$, summarized in the matrix below.

$$\boldsymbol{M}^{(2,0)+\mathcal{R}} = \overbrace{\begin{matrix} (3,0) & (2,1) & (4,0) & (3,1) & (2,2) & (5,0) & (4,1) & (3,2) & (2,3) \end{matrix}}^{(2,0)+\mathcal{R}\oplus\mathcal{S}_{\boldsymbol{L}}} \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & & & & \\ 6 & 7 & 8 & 9 & 10 & & & & \\ & & 1 & 2 & & 3 & 4 & 5 & \\ & & 6 & 7 & & 8 & 9 & 10 & \\ & & & 1 & 2 & & 3 & 4 & 5 \\ & & & 6 & 7 & & 8 & 9 & 10 \end{bmatrix} \begin{matrix} (2,0) \\ (3,0) \\ (2,1) \end{matrix} \Bigg\} (2,0)+\mathcal{R}.$$

Notice that, while the set of row and column indices changed, the array itself remains the same due to the shift-invariant structure of Macaulay matrices. Remember that any additional columns would be empty, since the full support of $((2,0)+\mathcal{R}) \cdot \boldsymbol{L}(\boldsymbol{x})$ is captured by the current set of columns. ◇

As previously explained, any columns of $\boldsymbol{M}^{\mathcal{R},\mathcal{C}}$ associated with indices $\boldsymbol{\gamma} \notin \mathcal{R} \oplus \mathcal{S}_{\boldsymbol{L}}$ are empty. Consequently, it is always possible to partition the set of columns $\mathcal{C}$ into $\mathcal{C}_1 = \mathcal{R} \oplus \mathcal{S}_{\boldsymbol{L}}$ and $\mathcal{C}_2 = \mathcal{C}\backslash(\mathcal{R} \oplus \mathcal{S}_{\boldsymbol{L}})$. This partitioning of the column indices allows us to trivially extend the null space basis of the nonzero part, i.e., $\boldsymbol{N}^{\mathcal{R}}$, to a basis for the whole null space by adding identity rows:

$$\boldsymbol{M}^{\mathcal{R},\mathcal{C}}\boldsymbol{N}^{\mathcal{R},\mathcal{C}} = \begin{matrix} \mathcal{R} \end{matrix} \begin{matrix} \begin{bmatrix} \mathcal{C}_1 & \mathcal{C}_2 \\ \boldsymbol{M}^{\mathcal{R}} & \boldsymbol{0} \end{bmatrix} \end{matrix} \begin{bmatrix} \boldsymbol{N}^{\mathcal{R}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{matrix} \mathcal{C}_1 \\ \mathcal{C}_2 \end{matrix} = \boldsymbol{0}. \tag{5.4}$$

As will be explained in Section 5.3, the trivial part of the null space corresponding to the row indices in $\mathcal{C}_2$ can essentially be ignored for the purpose of computations, since all the information is contained in $\boldsymbol{N}^{\mathcal{R}}$.

Finally, the shift-invariance property 5.2.3 naturally extends to any null space basis as well. More precisely, for any null space basis $\boldsymbol{N}^{\mathcal{R}}$, there exists a basis $\boldsymbol{N}^{\boldsymbol{\sigma}+\mathcal{R}}$ such that: $\forall \boldsymbol{\gamma} \in (\boldsymbol{\sigma}+\mathcal{R}) \oplus \mathcal{S}_{\boldsymbol{L}} : \boldsymbol{N}^{\boldsymbol{\sigma}+\mathcal{R}}_{\boldsymbol{\gamma},i} = \boldsymbol{N}^{\mathcal{R}}_{\boldsymbol{\gamma}-\boldsymbol{\sigma},i}$. This can be elegantly captured by a re-indexing operation $\phi_{\boldsymbol{\sigma}} : (\boldsymbol{\sigma}+\mathcal{R}) \times \{1,\dots,n\} \to \mathcal{R} : (\boldsymbol{\gamma}, i) \mapsto (\boldsymbol{\gamma}-\boldsymbol{\sigma}, i)$, such that this relation can be written as a concatenation of maps:

$$\boldsymbol{N}^{\boldsymbol{\sigma}+\mathcal{R}} = \boldsymbol{N}^{\mathcal{R}} \circ \phi_{\boldsymbol{\sigma}}, \text{ that is, } (\boldsymbol{N}^{\mathcal{R}} \circ \phi_{\boldsymbol{\sigma}})_{\boldsymbol{\gamma},i} = \boldsymbol{N}^{\mathcal{R}}_{\boldsymbol{\gamma}-\boldsymbol{\sigma},i}, \tag{5.5}$$

where $\boldsymbol{\gamma} \in \mathcal{R} \oplus \mathcal{S}_{\boldsymbol{L}}$. Importantly, this implies that for any $\boldsymbol{\sigma} \in \mathbb{N}_0^m$, when a basis $\boldsymbol{N}^{\mathcal{R}}$ is known, $\boldsymbol{N}^{\boldsymbol{\sigma}+\mathcal{R}}$ follows directly by simply re-indexing the rows using the constant shift $\boldsymbol{\sigma}$. Moreover, by (5.4) such a basis can be trivially embedded in a larger set of column indices.

5.3 Intersection-based null space computation

Block-Macaulay matrices are often much larger than their null space bases, as both the number of columns and rows of block-Macaulay matrices grow combinatorially with the total degree of the considered matrix. Moreover, in practice, the nullity often grows much more slowly than the rank of Macaulay matrices. To avoid this combinatorial growth of rows, it is of interest to not explicitly construct (parts of) the block-Macaulay matrix, but rather, to compute the null space directly instead. Note that the number of rows of a null space basis does, however, still grow combinatorially with the degree.

This section explores this idea: Subsection 5.3.1 introduces an intersection-based approach to compute the null space of a larger Macaulay matrix starting from the null spaces of two smaller ones. Subsection 5.3.2 then introduces a decomposition into block rows that allows for a recursive computation of the null space. Finally, Subsection 5.3.3 presents an algorithm to efficiently compute these intersections by exploiting the structure of this subproblem.

5.3.1 Avoiding construction of the Macaulay matrix

In the previous section, the importance of the row-index set $\mathcal{R}$ as well as the associated shift-invariance property was highlighted. In this Section, it is shown how this shift-invariance can be exploited to construct null space bases related to

larger row-index sets, without explicit construction of the associated Macaulay matrix. This concept is illustrated using a simple example where the null space of a banded block-Toeplitz matrix is computed.

Given two sets of row indices $\mathcal{R}_1$ and $\mathcal{R}_2$, two Macaulay matrices associated with the same matrix polynomial $\boldsymbol{L}(\boldsymbol{x})$ can be constructed, i.e., $\boldsymbol{M}^{\mathcal{R}_1}$ and $\boldsymbol{M}^{\mathcal{R}_2}$. Now define the sets of column indices $\mathcal{C}_1 = \mathcal{R}_1 \oplus \mathcal{S}_{\boldsymbol{L}}$ and $\mathcal{C}_2 = \mathcal{R}_2 \oplus \mathcal{S}_{\boldsymbol{L}}$. Trivially expanding both their column index sets to $\mathcal{C}_1 \cup \mathcal{C}_2$, and assuming that $\mathcal{R}_1 \cap \mathcal{R}_2 = \emptyset$, these matrices can be interpreted as block rows associated with the row index sets $\mathcal{R}_1$ and $\mathcal{R}_2$ of the larger block-Macaulay matrix $\boldsymbol{M}^{\mathcal{R}_1 \cup \mathcal{R}_2}$. This matrix can then always be partitioned as:

$$\boldsymbol{M}^{\mathcal{R}_1 \cup \mathcal{R}_2} = \begin{array}{c} \begin{array}{ccc} \mathcal{C}_1 \backslash \mathcal{C}_2 & \mathcal{C}_1 \cap \mathcal{C}_2 & \mathcal{C}_2 \backslash \mathcal{C}_1 \end{array} \\ \left[\begin{array}{c|c|c} \boldsymbol{M}^{\mathcal{R}_1}_{:,\mathcal{C}_1 \backslash \mathcal{C}_2} & \boldsymbol{M}^{\mathcal{R}_1}_{\mathcal{C}_1 \cap \mathcal{C}_2} & \boldsymbol{0} \\ \hline \boldsymbol{0} & \boldsymbol{M}^{\mathcal{R}_2}_{:,\mathcal{C}_1 \cap \mathcal{C}_2} & \boldsymbol{M}^{\mathcal{R}_2}_{:,\mathcal{C}_2 \backslash \mathcal{C}_1} \end{array}\right] \begin{array}{c} \mathcal{R}_1 \\ \mathcal{R}_2 \end{array} \end{array}. \tag{5.6}$$

Remark that when the assumption $\mathcal{R}_1 \cap \mathcal{R}_2 = \emptyset$ is violated, a similar construction can still be performed, resulting in a matrix with the same row space as $\boldsymbol{M}^{\mathcal{R}_1 \cup \mathcal{R}_2}$. Indeed, duplicate row multi-indices only result in duplicate rows, which, of course, do not alter the underlying row space.

Any vector orthogonal to both block rows of (5.6) then lies in the intersection of their respective null spaces. As such, given that the null spaces $\boldsymbol{N}^{\mathcal{R}_1}$ and $\boldsymbol{N}^{\mathcal{R}_2}$ are known, the null space of this larger Macaulay matrix $\boldsymbol{N}^{\mathcal{R}_1 \cup \mathcal{R}_2}$ then follows by trivially extending the null spaces as in (5.4) and considering this intersection:

$$\text{null } \boldsymbol{M}^{\mathcal{R}_1 \cup \mathcal{R}_2} = \text{range} \left[\begin{array}{c|c} \boldsymbol{N}^{\mathcal{R}_1}_{\mathcal{C}_1 \backslash \mathcal{C}_2,:} & \boldsymbol{0} \\ \boldsymbol{N}^{\mathcal{R}_1}_{\mathcal{C}_1 \cap \mathcal{C}_2,:} & \boldsymbol{0} \\ \hline \boldsymbol{0} & \boldsymbol{I} \end{array}\right] \cap \text{range} \left[\begin{array}{c|c} \boldsymbol{0} & \boldsymbol{I} \\ \hline \boldsymbol{N}^{\mathcal{R}_2}_{\mathcal{C}_1 \cap \mathcal{C}_2,:} & \boldsymbol{0} \\ \boldsymbol{N}^{\mathcal{R}_2}_{\mathcal{C}_2 \backslash \mathcal{C}_1,:} & \boldsymbol{0} \end{array}\right] \begin{array}{l} \mathcal{C}_1 \backslash \mathcal{C}_2 \\ \mathcal{C}_1 \cap \mathcal{C}_2 \\ \mathcal{C}_2 \backslash \mathcal{C}_1 \end{array}. \tag{5.7}$$

In Section 5.3.3, an approach to efficiently compute a basis for the result of this intersection will be presented.

Now, by appropriately decomposing the desired set of row indices $\mathcal{R}$ into subsets, repeated application of (5.7) allows us to compute the associated null space $\boldsymbol{N}^{\mathcal{R}}$ as a series of intersections of the null spaces associated with these subsets of row indices. Moreover, the shift-invariance property (5.5) of the null space bases, and the associated relabeling operation, allows us to straightforwardly obtain null space bases associated with shifts of previous sets of row indices. This enables the reuse of previously computed 'smaller' null space bases, and, by extension, a recursive computation of $\boldsymbol{N}^{\mathcal{R}}$. Computing the desired bases through intersections then avoids the need to construct the full Macaulay matrix $\boldsymbol{M}^{\mathcal{R}}$. We first illustrate this approach for the onedimensional case. Next, in

Subsection 5.3.3, a decomposition will be presented that allows the recursive computation of null space bases for general Macaulay matrices.

Example 5.3.1. For polynomial eigenvalue problems, the block-Macaulay matrix boils down to a block-Toeplitz matrix. Consider the following polynomial eigenvalue problem:

$$\boldsymbol{L}(\boldsymbol{x})\boldsymbol{v} = (\boldsymbol{L}_1 + \boldsymbol{L}_2 x + \boldsymbol{L}_3 x^2 + \boldsymbol{L}_4 x^3)\boldsymbol{v} = \boldsymbol{0},\ \|\boldsymbol{v}\|_1 \neq 0$$

The block-Macaulay matrix associated with the row indices $\mathcal{R}_1 = \{0, 1, 2\}$ is then:

$$\boldsymbol{M}^{\mathcal{R}_1} = \begin{bmatrix} \boldsymbol{L}_1 & \boldsymbol{L}_2 & \boldsymbol{L}_3 & \boldsymbol{L}_4 & & \\ & \boldsymbol{L}_1 & \boldsymbol{L}_2 & \boldsymbol{L}_3 & \boldsymbol{L}_4 & \\ & & \boldsymbol{L}_1 & \boldsymbol{L}_2 & \boldsymbol{L}_3 & \boldsymbol{L}_4 \end{bmatrix}$$

Now, $\boldsymbol{M}^{\mathcal{R}_1}$ essentially acts as a block row to the larger block-Toeplitz matrix with row indices $\mathcal{R}_1 \cup (3 + \mathcal{R}_1) = \{0, ..., 5\} = \mathcal{R}_2$, that is, with a shorthand notation, like in Equation (5.6):

$$\boldsymbol{M}^{\mathcal{R}_2} = \boldsymbol{M}^{\mathcal{R}_1 \cup (3+\mathcal{R}_1)} = \begin{array}{c} \begin{matrix} \{1,2,3\} & \{4,5,6\} & \{7,8,9\} \end{matrix} \\ \left[\begin{array}{cc|c} \multicolumn{2}{c|}{\boldsymbol{M}^{\mathcal{R}_1}} & \boldsymbol{0} \\ \hline \boldsymbol{0} & \multicolumn{2}{c}{\boldsymbol{M}^{(3+\mathcal{R}_1)}} \end{array}\right] \begin{matrix} \{1,2,3\} \\ \{4,5,6\} \end{matrix} \end{array}$$

$$= \left[\begin{array}{cccccc|cc} \boldsymbol{L}_1 & \boldsymbol{L}_2 & \boldsymbol{L}_3 & \boldsymbol{L}_4 & & & & \\ & \boldsymbol{L}_1 & \boldsymbol{L}_2 & \boldsymbol{L}_3 & \boldsymbol{L}_4 & & & \\ & & \boldsymbol{L}_1 & \boldsymbol{L}_2 & \boldsymbol{L}_3 & \boldsymbol{L}_4 & & \\ \hline & & & \boldsymbol{L}_1 & \boldsymbol{L}_2 & \boldsymbol{L}_3 & \boldsymbol{L}_4 & \\ & & & & \boldsymbol{L}_1 & \boldsymbol{L}_2 & \boldsymbol{L}_3 & \boldsymbol{L}_4 \end{array}\right].$$

Due to the similarity of $\boldsymbol{M}^{\mathcal{R}_1}$ and $\boldsymbol{M}^{(3+\mathcal{R}_1)}$, this larger matrix does not need to be constructed to compute its null space, however.

Indeed, a null space basis of the larger matrix $\boldsymbol{M}^{\mathcal{R}_2}$ can be obtained by intersecting the null spaces of the two individual block rows $\boldsymbol{M}^{\mathcal{R}_1}$ and $\boldsymbol{M}^{3+\mathcal{R}_1}$. Let $\boldsymbol{N}^{\mathcal{R}_1}$ be a basis for the null space of $\boldsymbol{M}^{\mathcal{R}}$. Due to the shift-invariance property (5.5), a basis $\boldsymbol{N}^{\mathcal{R}_2} = \boldsymbol{N}^{3+\mathcal{R}_1} = \boldsymbol{N}^{\mathcal{R}_1} \circ \phi_3$ is also known. By (5.7), the null space associated with the larger block-Macaulay matrix $\boldsymbol{M}^{\mathcal{R}_2}$ can then be computed through an intersection:

$$\operatorname{null} \boldsymbol{M}^{\{1,\ldots,5\}} = \operatorname{range} \left[\begin{array}{c|c} \boldsymbol{N}^{\{1,\ldots,2\}} & \boldsymbol{0} \\ \hline \boldsymbol{0} & \boldsymbol{I} \end{array}\right] \cap \operatorname{range} \left[\begin{array}{c|c} \boldsymbol{0} & \boldsymbol{I} \\ \hline \boldsymbol{N}^{\{3,\ldots,5\}} & \boldsymbol{0} \end{array}\right] \begin{matrix} \{0,1,2\} \\ \{3,4,5\}. \\ \{6,7,8\} \end{matrix}$$

Notice that the block-Macaulay matrix with the shifts $\mathcal{R}_2$ was never constructed. Using the resulting null space $\boldsymbol{N}^{\mathcal{R}_2}$, this procedure can be repeated to compute a null space associated with row indices $\mathcal{R}_3 = \mathcal{R}_2 \cup (6 + \mathcal{R}_2) = \{0, ..., 11\}$. Likewise, the recursive decomposition $\mathcal{R}_{i+1} = \mathcal{R}_i \cup ((\max(\mathcal{R}_i) + 1) + \mathcal{R}_i)$ then leads to a computation of null spaces associated with row-index sets of increasing size. ◇

5.3.2 Decomposing the row indices for recursive computations

As hinted at in Subsection 5.3.1, through repeated application of (5.7) any decomposition of $\mathcal{R}$ entails a way to compute $\boldsymbol{N}^{\mathcal{R}}$ directly from the null space bases associated with each of the subsets of the decomposition. For the purpose of solving MEPs, a common choice of $\mathcal{R}$ is the set of all shifts up to some degree d, the multi-exponents of which are described by the set $\mathcal{X}_d = \{\boldsymbol{\rho} \in \mathbb{N}_0^m \mid \|\boldsymbol{\rho}\|_1 \leq d\}$. While for polynomials, other choices of $\mathcal{R}$ have been explored to exploit a certain structure to the support of $\boldsymbol{L}(\boldsymbol{x})$ [16], this is case-specific and as such will not be touched upon.

In this subsection, a recursive decomposition of $\mathcal{X}_d$ is presented, which entails a recursive approach to compute the null space of the associated Macaulay matrix by exploiting the shift property (5.5). We first illustrate this decomposition in 2D, after which it is generalized to general dimensions m.

Example 5.3.2. Consider the set of all row indices up to degree $d = 3$, $\mathcal{X}_d$, where $\mathcal{X}_d = \{\boldsymbol{\rho} \in \mathbb{N}_0^2 \mid \|\boldsymbol{\rho}\|_d \leq d\}$. Define $\mathcal{G}_d = \{\boldsymbol{\rho} \in \mathbb{N}_0^2 \mid \|\boldsymbol{\rho}\|_\infty \leq d\}$, and note that the following decomposition holds $\mathcal{X}_3 = \mathcal{G}_1 \cup ((2,0) + \mathcal{X}_1) \cup ((0,2) + \mathcal{X}_1)$. Additionally, $\mathcal{G}_3 = \{\boldsymbol{\rho} \in \mathbb{N}_0^2 \mid \|\boldsymbol{\rho}\|_\infty \leq 3\}$ can be decomposed as well: first compute $\mathcal{B} = \mathcal{G}_1 \cup ((2,0) + \mathcal{G}_3)$, such that $\mathcal{G}_3 = \mathcal{B} \cup ((0,2) + \mathcal{B})$. Both decompositions are visualized in Figure 5.1. A similar decomposition can be

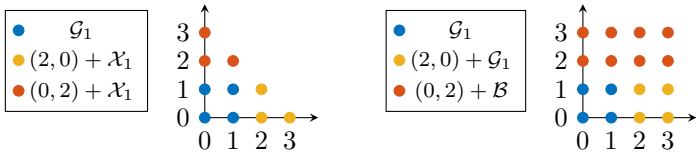


Figure 5.1: Decompositions of the multi-indices up to degree three, i.e., $\mathcal{X}_3$, and the multi-indices up to infinity norm three, that is, $\mathcal{G}_3$ in 2D. These decompositions naturally imply a block-row decomposition of the associated Macaulay matrix.

repeated for $\mathcal{X}_7$ and $\mathcal{G}_7$, by using shifts of the form $4\boldsymbol{e}_i$ instead of $2\boldsymbol{e}_i$, where $\boldsymbol{e}_i$

denotes the ith column of the 2×2 identity matrix. As such, these sets can be grown recursively.

Now, by (5.7), an intersection of two null spaces can be associated with each $\cup$-operation, and (5.5) allows us to obtain a null space associated with a shifted row set. Let us consider constructing $\boldsymbol{N}^{\mathcal{X}_3}$: Starting from $\boldsymbol{N}^{\mathcal{X}_1}$ and $\boldsymbol{N}^{\mathcal{G}_1}$, we first relabel its rows by applying 5.5 to obtain $\boldsymbol{N}^{(2,0)+\mathcal{X}_1,\mathcal{C}} = \boldsymbol{N}^{\mathcal{X}_1} \circ \phi_{(2,0)}$ and $\boldsymbol{N}^{(2,0)+\mathcal{X}_1,\mathcal{C}} = \boldsymbol{N}^{\mathcal{X}_1} \circ \phi_{(0,2)}$. Next, let $\mathcal{C} = \mathcal{X}_3 \oplus \mathcal{S}_{\boldsymbol{L}}$ and note that $\boldsymbol{N}^{\mathcal{G}_1,\mathcal{C}}, \boldsymbol{N}^{(2,0)+\mathcal{X}_1,\mathcal{C}}$ and $\boldsymbol{N}^{(0,2)+\mathcal{X}_1,\mathcal{C}}$ can be readily obtained by trivially extending[4] $\boldsymbol{N}^{\mathcal{G}_1}, \boldsymbol{N}^{(2,0)+\mathcal{X}_1}$ and $\boldsymbol{N}^{(0,2)+\mathcal{X}_1}$, respectively, as in (5.4). Then, by associating an intersection with each union needed to construct the set of row indices $\mathcal{X}_3$, the desired basis can be constructed:

$$\begin{aligned}\operatorname{range} \boldsymbol{N}^{\mathcal{X}_3} &= \operatorname{range} \boldsymbol{N}^{\mathcal{G}_1 \cup (2,0)+\mathcal{X}_1 \cup (0,2)+\mathcal{X}_1} \\ &= \operatorname{range} \boldsymbol{N}^{\mathcal{G}_1,\mathcal{C}} \cap \operatorname{range} \boldsymbol{N}^{(2,0)+\mathcal{X}_1,\mathcal{C}} \cap \operatorname{range} \boldsymbol{N}^{(0,2)+\mathcal{X}_1,\mathcal{C}}.\end{aligned}$$

Finally, a similar procedure can be applied to compute $\boldsymbol{N}^{\mathcal{G}_3}$. First, let $\mathcal{C}_1 = \mathcal{B} \oplus \mathcal{S}_{\boldsymbol{L}}$, so that $\operatorname{range} \boldsymbol{N}^{\mathcal{B}} = \operatorname{range} \boldsymbol{N}^{\mathcal{G}_3,\mathcal{C}_1} \cap \operatorname{range} \boldsymbol{N}^{(2,0)+\mathcal{G}_3,\mathcal{C}_1}$, where both arguments of this operation again follow from re-indexing and extending $\boldsymbol{N}^{\mathcal{G}_1}$ using (5.5) and (5.4), respectively. Then, let $\mathcal{C}_1 = \mathcal{G}_3 \oplus \mathcal{S}_{\boldsymbol{L}}$, and compute a basis for $\operatorname{range} \boldsymbol{N}^{\mathcal{G}_3} = \boldsymbol{N}^{\mathcal{B},\mathcal{C}_2} \cap \operatorname{range} \boldsymbol{N}^{(0,2)+\mathcal{B},\mathcal{C}_2}$ to obtain the desired result. This process can then be repeated to obtain $\boldsymbol{N}^{\mathcal{X}_7}$ and $\boldsymbol{N}^{\mathcal{G}_7}$, and so on. ◇

We now present a generalization of this decomposition to arbitrary dimensions. First, note that an m-dimensional cubic subset of $\mathbb{N}_0^m$ can be described, similarly to the 2D case, as $\mathcal{G}_d = \{\boldsymbol{\rho} \in \mathbb{N}_0^m \mid \|\boldsymbol{\rho}\|_\infty \leq d\}$. Next, consider the family of sets described by $\mathcal{B}_d^k = \{\boldsymbol{\rho} \in \mathbb{N}_0^m \mid \|\boldsymbol{\rho}\|_\infty \leq d, \|\boldsymbol{\rho}\|_1 \leq k(d+1) - 1\}$. Due to the following relation between the 1- and $\infty-$norm: $\forall \boldsymbol{\rho} \in \mathbb{N}_0^m : m\|\boldsymbol{\rho}\|_\infty \geq \|\boldsymbol{\rho}\|_1$, it then holds that $\mathcal{X}_d = \mathcal{B}_d^1$ and $\mathcal{G}_d = \mathcal{B}_d^k$ for $k \geq m$. Lastly, note that $\mathcal{B}_d^0$ is the empty set. We now decompose $\mathcal{X}_d$ for general dimension by showing that any set $\mathcal{B}_{2d+1}^k$ can be decomposed into shifts of the smaller sets $\mathcal{B}_d^k$.

First, subdivide each of the axes into two parts: one smaller than or equal to d and one larger than d. To formalize this, consider the following subparts of the grid: $\mathcal{P}_{\mathcal{I}} = \{\boldsymbol{\rho} \in \mathbb{N}_0^m \mid \forall i \notin \mathcal{I} : \rho_i \leq d, \forall i \in \mathcal{I} : \rho_i > d\}$, where $\mathcal{I} \subseteq \{1, \ldots, m\}$ indicates precisely for which coordinates $\mathcal{P}_{\mathcal{I}}$ lies in the second part of the corresponding axis. For all parts $\mathcal{P}_{\mathcal{I}}$ of the grid, each characterized by some $\mathcal{I} \subseteq \{1, \ldots, m\}$, we then look at the subset of $\mathcal{B}_{2d+1}^k$ located there. Let $\boldsymbol{e}_i$ denote the ith column of the $m \times m$ identity matrix, the subpart of $\mathcal{B}_{2d+1}^k$

[4] As will be explained in Subsection 5.3.3, this extension need not be performed explicitly. Instead, the identity rows may essentially be ignored during the intersection.

located in $\mathcal{P}_\mathcal{I}$ is then:

$$\begin{aligned}
&\mathcal{B}^k_{2d+1} \cap \mathcal{P}_\mathcal{I} \\
&= \{\boldsymbol{\rho} \mid \forall i \in \mathcal{I} : d+1 \leq \rho_i \leq 2d+1, \forall i \notin \mathcal{I} : \rho_i \leq d,\ \|\boldsymbol{\rho}\|_1 \leq k(2d+2)-1\} \\
&= \{\boldsymbol{\rho} \mid \forall i : \rho_i \leq d,\ \|\boldsymbol{\rho}\|_1 \leq (2k-\#\mathcal{I})(d+1)-1\} + \sum_{i\in\mathcal{I}} (d+1)\boldsymbol{e}_i \\
&= \{\boldsymbol{\rho} \mid\ \|\boldsymbol{\rho}\|_\infty \leq d,\ \|\boldsymbol{\rho}\|_1 \leq (2k-\#\mathcal{I})(d+1)-1\} + \sum_{i\in\mathcal{I}} (d+1)\boldsymbol{e}_i \\
&= \mathcal{B}^{2k-\#\mathcal{I}}_d + \sum_{i\in\mathcal{I}} (d+1)\boldsymbol{e}_{i_l}.
\end{aligned} \tag{5.8}$$

The first equality holds because the operation $+(d+1)\boldsymbol{e}_i$ is simply a shift over $d+1$ grid points in the ith coordinate and $\forall \boldsymbol{\rho} \in \mathbb{N}_0^m : \|\boldsymbol{\rho}\|_1 = \sum_{i=1}^m \rho_i$.

Since (5.8) holds for any $\mathcal{I} \subseteq \{1,\ldots,m\}$, each $\mathcal{B}^k_{2d+1}$ can be decomposed into the smaller, shifted sets $\mathcal{B}^{2k-\#\mathcal{I}}_d$, where $\#\mathcal{I}$ indicates the number of size-$(d+1)$ shifts required for the placement of $\mathcal{B}^{2k-\#\mathcal{I}}_d$ within the larger set $\mathcal{B}^k_{2d+1}$. Notice that the required number of shifts $\#\mathcal{I}$ is the only factor determining the precise 'type' k of subpart is placed there. Moreover, since all larger parts $\mathcal{B}^k_{2d+1}$ may be built from their smaller-sized counterparts, many of the intermediate results re-occur in the construction of parts of a different 'type'. To keep the number of union operations low, it is of interest to reuse these similar building blocks as best as possible.

Before we arrive at an updating scheme, let us introduce some definitions to more easily talk about sets that respect the structure observed in (5.8). For the purposes of this brief exposition, we call any set $\mathcal{D}$ for which $\forall I \subseteq \{1,\ldots,r\}$ and $\forall j \in \{1,\ldots,r\}\backslash\mathcal{I} : \mathcal{D} \cap \mathcal{P}_\mathcal{I} = \mathcal{B}^k_d$ and $\mathcal{D} \cap \mathcal{P}_{\mathcal{I}\cup\{j\}} = \mathcal{B}^{k-1}_d$, *well-structured w.r.t. the indices* $\{1,\ldots,r\}$. Fixing the 'type' k of $\mathcal{D}\cap\mathcal{P}_{\{\}}$ then fully defines a well-structured $\mathcal{D}$ over all parts $\mathcal{P}_\mathcal{I}, \mathcal{I} \subseteq \{1,\ldots,r\}$. Indeed, remark that by (5.8), all $\mathcal{B}^k_{2d+1}$ are well-structured w.r.t. $\{1,\ldots,m\}$ with $\mathcal{D}\cap\mathcal{P}_{\{\}} = \mathcal{B}^{2k}_d$.

Now, given two sets $\mathcal{D}_1$ and $\mathcal{D}_2$, both well-structured w.r.t. $\{1,\ldots,r\}$, then $\mathcal{D}_2$ is considered *adjacent to* $\mathcal{D}_1$ whenever $\mathcal{D}_1 \cap \mathcal{P}_{\{\}} = \mathcal{B}^k_d$ and $\mathcal{D}_2 \cap \mathcal{P}_{\{\}} = \mathcal{B}^{k-1}_d$. The following properties are then readily verified: For two adjacent sets it holds that $\mathcal{D}_1 \cup ((d+1)\boldsymbol{e}_{r+1} + \mathcal{D}_2)$ is well-structured w.r.t. $\{1,\ldots,r+1\}$. Moreover, given a third set $\mathcal{D}_3$ adjacent to $\mathcal{D}_2$, then $\mathcal{D}_2 \cup ((d+1)\boldsymbol{e}_{r+1} + \mathcal{D}_3)$ is adjacent to $\mathcal{D}_1 \cup ((d+1)\boldsymbol{e}_{r+1} + \mathcal{D}_2)$.

The latter properties lead to a straightforward updating scheme of all $\mathcal{B}^k_d$, $k = 1\ldots,m$. Notice that $\mathcal{B}^m_d, \ldots, \mathcal{B}^1_d, \mathcal{B}^0_d$ are all adjacent to their direct predecessors

w.r.t. to the indices $\{\}$. As such, defining the sets $\mathcal{D}_1^j = \mathcal{B}_d^j \cup ((d+1)\boldsymbol{e}_1 + \mathcal{B}_d^{j-1})$ leads to a new series of well-structured sets w.r.t. the first index, all adjacent to their predecessors. Remember that $\mathcal{B}_d^0 = \emptyset$ and $\mathcal{B}_d^{k \geq m} = \mathcal{G}_d$, so that the above definition leads to $\mathcal{D}_1^{k \leq 0} = \emptyset$ or $\mathcal{D}_1^{k \geq m+1} = \mathcal{D}_1^{m+1}$.

Consequently $\mathcal{D}_1^{m+1}$ acts as its own predecessor and $\mathcal{D}_1^{k \leq 0}$ does not have to be determined. This leads to $m+1$ different nonempty sets with which this process can now be repeated: $\mathcal{D}_{r+1}^j = \mathcal{D}_r^j \cup ((d+1)\boldsymbol{e}_{r+1} + \mathcal{D}_r^{j-1})$. Noting that $\mathcal{D}_r^{k \leq 0} = \emptyset$ and $\mathcal{D}_r^{k \geq m+r} = \mathcal{D}_r^{m+r}$ due to similar reasons as before, this then forms another series of well structured, adjacent sets w.r.t. $\{1, \ldots, r+1\}$. This leads to $2m-1$ nonempty, different sets after $m-1$ iterations, though only m sets are required by the analysis of (5.8). Notice that, by their construction, $\forall r : \mathcal{D}_r^j \cap \mathcal{P}_{\{\}} = \mathcal{D}_1^j \cap \mathcal{P}_{\{\}} = \mathcal{B}_d^j$, so that, by (5.8) it holds that $\mathcal{B}_{2d+1}^k = \mathcal{D}_m^{2k}$.

As illustrated in Example 5.3.2 through (5.7), each of these unions can be associated with an intersection operation, and through the shift-invariance property (5.5), bases for $\boldsymbol{N}^{\boldsymbol{\sigma}+\mathcal{B}_d^k}$ are readily obtained from $\boldsymbol{N}^{\mathcal{B}_d^k}$. Algorithm 1 then summarizes how these properties can be employed to compute $\boldsymbol{N}^{\mathcal{X}_{2^l-1}}$. Notice that the innermost loop may be performed in parallel. Lastly, remark that the algorithm reflects that, to obtain $\mathcal{X}_{2^l-1}$, not all $\mathcal{B}_d^k$ need to be updated to the full size $d = 2^l - 1$ and that unions with the empty set, of course, need not be explicitly performed.

Remarks: The presented decomposition is one possible way to decompose $\mathcal{X}_d$. As stated in Subsection 5.3, any decomposition can likewise be associated with a series of intersections, leading to $\boldsymbol{N}^{\mathcal{X}_d}$. It stems from the heuristic idea that, to book as much progress as possible with each intersection, all the subsets of the decomposition should be disjoint. By no means do the authors imply that this is the best possible option, something that is most likely highly problem-dependent.

To the best of the author's knowledge, there seem to be no theoretical results available on the (approximate) dimensions of range $\boldsymbol{N}^{\mathcal{B}_d^k}$ for various k, a number that greatly influences the size of the considered subproblems. This lack of knowledge complicates an asymptotic complexity analysis.

Another drawback may be that range $\boldsymbol{N}^{\mathcal{B}_d^k}$ has to be stored for all values of k, while they all represent null spaces of very similar matrices and thus should, intuitively, contain a lot of the same information. However, the core idea, that is, using the shift property (5.5) is useful for the reuse of intermediate results, which has great potential to reduce the overall computational load.

Algorithm 1 Recursive construction of $\boldsymbol{N}^{\mathcal{X}_{2^l-1}}$ using the decomposition of $\mathcal{X}_{2^l-1}$ as presented in Section 5.3.2.

function BUILD NULL SPACE BASIS($\boldsymbol{N}^{\{\mathbf{0}\}}, l$)
  $\forall k \in \{1, \ldots, m\} : \boldsymbol{N}^{\mathcal{D}_k} \leftarrow \boldsymbol{N}^{\{\mathbf{0}\}}$
  $d \leftarrow 0$
  for $i = 1, .., l$ **do**
    last $\leftarrow 2^{l-i}$
    for $j = 1, .., m-1$ **do**
      $\boldsymbol{\sigma} = (d+1)\boldsymbol{e_j}$
      for $k = \min(m+j+1, 2 \cdot \text{last}), .., 2$ **do**
        q $\leftarrow \min(k, m+j)$
        $\boldsymbol{N}^{\mathcal{D}_k} \leftarrow$ structured intersection$(\boldsymbol{N}^{\mathcal{D}_q}, \boldsymbol{N}^{\mathcal{D}_{k-1}} \circ \phi_{\boldsymbol{\sigma}})$ ▷ Alg. 2.
      end for
    end for
    $\boldsymbol{\sigma} = (d+1)\boldsymbol{e_m}$
    for $k = 1, , .., \min(\text{last}, m)$ **do**
      $\boldsymbol{N}^{\mathcal{D}_k} \leftarrow$ structured intersection$(\boldsymbol{N}^{\mathcal{D}_{2k}}, \boldsymbol{N}^{\mathcal{D}_{2k-1}} \circ \phi_{\boldsymbol{\sigma}})$ ▷ Alg. 2.
    end for
    $d \leftarrow 2d+1$
  end for
return $\boldsymbol{N}^{\mathcal{D}_1}$
end function

5.3.3 Computing bases for the intersections

Many algorithms can be used to compute a basis for the subspace resulting from each of the intersection operations (5.7) required [65, 207]. In this section, we present one such approach that exploits the structure of this subproblem. Indeed, it is shown that one can essentially ignore the identity rows involved in this problem, solving a smaller null space problem derived from the rows associated with $\mathcal{C}_1 \cap \mathcal{C}_2$ instead. The latter null space problem can be tackled through a singular value decomposition (SVD), among other approaches.

The null spaces in (5.7) have a particular structure due to the identity rows resulting from extending the set of column indices. For the sake of generality and notational brevity, we rename, $\boldsymbol{B}_1 = \boldsymbol{N}^{\mathcal{R}_1}_{\mathcal{C}_1 \cap \mathcal{C}_2,:}, \boldsymbol{C}_1 = \boldsymbol{N}^{\mathcal{R}_2}_{\mathcal{C}_1 \cap \mathcal{C}_2,:}$ and $\boldsymbol{B}_2 = \boldsymbol{N}^{\mathcal{R}_1}_{\mathcal{C}_1 \setminus \mathcal{C}_2,:}, \boldsymbol{C}_2 = \boldsymbol{N}^{\mathcal{R}_2}_{\mathcal{C}_2 \setminus \mathcal{C}_1,:}$, where $\mathcal{C}_i = \mathcal{R}_i \oplus \mathcal{S}_{\boldsymbol{L}}$, so that $\boldsymbol{N}^{\mathcal{R}_1}$ and $\boldsymbol{N}^{\mathcal{R}_2}$ can

respectively be rewritten as

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{B}_2 & \boldsymbol{0} \\ \boldsymbol{B}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \text{ and } \boldsymbol{C} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \\ \boldsymbol{C}_1 & \boldsymbol{0} \\ \boldsymbol{C}_2 & \boldsymbol{0} \end{bmatrix}.$$

Next, note that it holds that $\forall \boldsymbol{p} \in \operatorname{range} \boldsymbol{B} \cap \operatorname{range} \boldsymbol{C} : \exists \boldsymbol{y}, \boldsymbol{z}$:

$$\boldsymbol{p} = \boldsymbol{B}\boldsymbol{y} = \begin{bmatrix} \boldsymbol{B}_2 & \boldsymbol{0} \\ \boldsymbol{B}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \\ \boldsymbol{C}_1 & \boldsymbol{0} \\ \boldsymbol{C}_2 & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{z}_1 \\ \boldsymbol{z}_2 \end{bmatrix} = \boldsymbol{C}\boldsymbol{z} \Leftrightarrow \begin{cases} \boldsymbol{B}_1\boldsymbol{y}_1 = \boldsymbol{C}_1\boldsymbol{z}_1 \\ \boldsymbol{z}_2 = \boldsymbol{B}_2\boldsymbol{y}_1 \\ \boldsymbol{y}_2 = \boldsymbol{C}_2\boldsymbol{z}_1 \end{cases} \quad (5.9)$$

Remark that $\boldsymbol{B}_1\boldsymbol{y}_1 = \boldsymbol{C}_1\boldsymbol{z}_1 \Leftrightarrow \begin{bmatrix} \boldsymbol{y}_1{}^\mathsf{T} & \boldsymbol{z}_1{}^\mathsf{T} \end{bmatrix}^\mathsf{T} \in \operatorname{null} \begin{bmatrix} \boldsymbol{B}_1 & -\boldsymbol{C}_1 \end{bmatrix}$. The elements in the intersection are thus in one-to-one correspondence with the solutions of (5.9), which is solved by determining a basis $\begin{bmatrix} \boldsymbol{Y}_1{}^\mathsf{T} & \boldsymbol{Z}_1{}^\mathsf{T} \end{bmatrix}^\mathsf{T}$ for $\operatorname{null} \begin{bmatrix} \boldsymbol{B}_1 & -\boldsymbol{C}_1 \end{bmatrix}$. In conclusion, a basis $\boldsymbol{P}$ for $\operatorname{range} \boldsymbol{B} \cap \operatorname{range} \boldsymbol{C}$ is then obtained as:

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{B}_2 & \boldsymbol{0} \\ \boldsymbol{B}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{C}_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{Y}_1 \\ \boldsymbol{Z}_1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{B}_2\boldsymbol{Y}_1 \\ \boldsymbol{B}_1\boldsymbol{Y}_1 \\ \boldsymbol{C}_2\boldsymbol{Z}_1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{B}_2\boldsymbol{Y}_1 \\ \boldsymbol{C}_1\boldsymbol{Z}_1 \\ \boldsymbol{C}_2\boldsymbol{Z}_1 \end{bmatrix}.$$

Finally, note that $\boldsymbol{P}$ is generally not orthonormal, even when $\boldsymbol{B}$ and $\boldsymbol{C}$ are. To restore the orthonormality, an additional, invertible transformation $\boldsymbol{V}$ is therefore applied to the columns $\boldsymbol{P}$ such that the Gramian $\boldsymbol{G} = \boldsymbol{V}^\mathsf{T}\boldsymbol{P}^\mathsf{T}\boldsymbol{P}\boldsymbol{V}$ equals the identity matrix: $\boldsymbol{G} = \boldsymbol{I}$. Based on the assumption that, by construction, $\begin{bmatrix} \boldsymbol{B}_2 \\ \boldsymbol{B}_1 \end{bmatrix}$, $\begin{bmatrix} \boldsymbol{C}_1 \\ \boldsymbol{C}_2 \end{bmatrix}$ and $\begin{bmatrix} \boldsymbol{Y}_1 \\ \boldsymbol{Z}_1 \end{bmatrix}$ are orthonormal, bases we obtain:

$$\begin{aligned} \boldsymbol{P}^\mathsf{T}\boldsymbol{P} &= \boldsymbol{Y}_1^\mathsf{T}\boldsymbol{B}_2^\mathsf{T}\boldsymbol{B}_2\boldsymbol{Y}_1 + \boldsymbol{Y}_1^\mathsf{T}\boldsymbol{B}_1^\mathsf{T}\boldsymbol{B}_1\boldsymbol{Y}_1 + \boldsymbol{Z}_1^\mathsf{T}\boldsymbol{C}_2^\mathsf{T}\boldsymbol{C}_2\boldsymbol{Z}_1 \\ &= (\boldsymbol{Y}_1^\mathsf{T}\boldsymbol{Y}_1 + \boldsymbol{Z}_1^\mathsf{T}\boldsymbol{Z}_1 - \boldsymbol{Z}_1{}^\mathsf{T}\boldsymbol{C}_1{}^\mathsf{T}\boldsymbol{C}_1\boldsymbol{Z}_1) \\ &= (\boldsymbol{I} - \boldsymbol{Z}_1{}^\mathsf{T}\boldsymbol{C}_1{}^\mathsf{T}\boldsymbol{C}_1\boldsymbol{Z}_1). \end{aligned}$$

Now consider the full SVD $\boldsymbol{C}_1\boldsymbol{Z}_1 = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\mathsf{T}$, from which the eigendecomposition $\boldsymbol{P}^\mathsf{T}\boldsymbol{P} = \boldsymbol{V}(\boldsymbol{I} - \boldsymbol{S}^2)\boldsymbol{V}^\mathsf{T}$ follows. Note that, since $\boldsymbol{P}$ forms a basis, none of the eigenvalues are zero. $\boldsymbol{P}\boldsymbol{V}(\boldsymbol{I} - \boldsymbol{S}^2)^{\frac{-1}{2}}$ then forms an orthonormal basis for $\operatorname{range} \boldsymbol{P}$.

Algorithm 2 then summarizes how this procedure is to be employed to compute an orthonormal $\boldsymbol{N}^{\mathcal{R}_1 \cup \mathcal{R}_2}$ from $\boldsymbol{N}^{\mathcal{R}_1}, \boldsymbol{N}^{\mathcal{R}_2}$, as in (5.7). Note that the orthonormalization step may be omitted if it is not required.

Algorithm 2 Computation of the intersection (5.7), starting from $\boldsymbol{N}^{\mathcal{R}_1}$ and $\boldsymbol{N}^{\mathcal{R}_2}$ and resulting in $\boldsymbol{N}^{\mathcal{R}_1 \cup \mathcal{R}_2}$.

function STRUCTURED INTERSECTION($\boldsymbol{N}^{\mathcal{R}_1}, \boldsymbol{N}^{\mathcal{R}_2}$)
  $\mathcal{O} \leftarrow \mathcal{C}_1 \cap \mathcal{C}_2$
  $\mathcal{E} \leftarrow \mathcal{C}_2 \backslash \mathcal{C}_1$
  $\begin{bmatrix} \boldsymbol{Y}^\mathsf{T} & \boldsymbol{Z}^\mathsf{T} \end{bmatrix}^\mathsf{T} \leftarrow$ orthonormal basis for null $\begin{bmatrix} \boldsymbol{N}^{\mathcal{R}_1}_{\mathcal{O},:} & -\boldsymbol{N}^{\mathcal{R}_2}_{\mathcal{O},:} \end{bmatrix}$
  $\boldsymbol{P}^\mathsf{T} \leftarrow \begin{bmatrix} \boldsymbol{Y}^\mathsf{T} \left(\boldsymbol{N}^{\mathcal{R}_1}\right)^\mathsf{T} & \boldsymbol{Z}^\mathsf{T} \left(\boldsymbol{N}^{\mathcal{R}_1}_{\mathcal{E},:}\right)^\mathsf{T} \end{bmatrix}$
  $(\boldsymbol{U},)\boldsymbol{S}, \boldsymbol{V} \leftarrow$ full SVD of $\boldsymbol{N}^{\mathcal{R}_2}_{\mathcal{O},:}\boldsymbol{Z}$ ▷ Optional orthonormalization step
  $\boldsymbol{P} \leftarrow \boldsymbol{P}\boldsymbol{V}(\boldsymbol{I} - \boldsymbol{S}^2)^{-\frac{1}{2}}$
return $\boldsymbol{P}$
end function

5.4 Row space computation

Some applications, such as elimination [10, 203], require the use of the row space of a (block-)Macaulay matrix. Due to its duality with the null space, an approach similar to Algorithm 1 to compute an orthogonal basis for the row space is possible, and many of the previously mentioned properties can be reused. Indeed, it was already noted in Example 5.2.4 that shifting the set of row indices $\mathcal{R}$ only changes the column multi-indices, while the overall matrix remains the same. Consequently, in similar fashion to properties (5.4) and (5.5) of the null space, once a basis $\boldsymbol{Q}^{\mathcal{R}}$ for row $\boldsymbol{M}^{\mathcal{R}}$ is known, a basis for any space row $\boldsymbol{M}^{\boldsymbol{\sigma}+\mathcal{R},\mathcal{C}}$ follows immediately by relabeling the nonzero the columns and appropriately padding them with zero columns. Redefining the re-indexing operation to work on the column indices instead, $\varphi_{\boldsymbol{\sigma}} : (\boldsymbol{\sigma}+\mathcal{R}) \times \{1, \ldots, n\} \to \mathcal{R} : (i, \boldsymbol{\gamma}) \mapsto (i, \boldsymbol{\gamma}-\boldsymbol{\sigma})$, a basis can then be obtained as follows:

$$\boldsymbol{Q}^{\boldsymbol{\sigma}+\mathcal{R}} = \boldsymbol{Q}^{\mathcal{R}} \circ \varphi_{\boldsymbol{\sigma}}.$$

Now, the only ingredient required to build a row space basis using a similar shift-based approach is 'gluing' different bases together into a new basis. Given two bases $\boldsymbol{Q}^{\mathcal{R}_1}$ and $\boldsymbol{Q}^{\mathcal{R}_2}$ for $\operatorname{row} \boldsymbol{M}^{\mathcal{R}_1}$ and $\operatorname{row} \boldsymbol{M}^{\mathcal{R}_2}$, respectively, it then follows from (5.6) that:

$$\operatorname{row} \boldsymbol{M}^{\mathcal{R}_1 \cup \mathcal{R}_2} = \operatorname{row} \left[\begin{array}{c|c|c} \begin{matrix} \mathcal{C}_1 \backslash \mathcal{C}_2 \\ \boldsymbol{Q}^{\mathcal{R}_1}_{\mathcal{C}_1 \backslash \mathcal{C}_2} \end{matrix} & \begin{matrix} \mathcal{C}_1 \cap \mathcal{C}_2 \\ \boldsymbol{Q}^{\mathcal{R}_1}_{\mathcal{C}_1 \cap \mathcal{C}_2} \end{matrix} & \begin{matrix} \mathcal{C}_2 \backslash \mathcal{C}_1 \\ \boldsymbol{0} \end{matrix} \\ \hline \boldsymbol{0} & \boldsymbol{Q}^{\mathcal{R}_2}_{\mathcal{C}_1 \cap \mathcal{C}_2} & \boldsymbol{Q}^{\mathcal{R}_2}_{\mathcal{C}_2 \backslash \mathcal{C}_1} \end{array}\right].$$

Notice that the right-hand-side matrix is not necessarily a basis. As such, we now derive an algorithm to obtain an orthonormal basis for the row space of

this matrix, assuming $\boldsymbol{Q}^{\mathcal{R}_1}$ and $\boldsymbol{Q}^{\mathcal{R}_2}$ are orthonormal to begin with. Again, there are many ways to do this, though preferably, the problem structure is taken into account.

Take linear combinations of the rows by left-multiplying them with some matrix $\boldsymbol{T}$:

$$\boldsymbol{Q}^{\mathcal{R}_1 \cup \mathcal{R}_2} = \boldsymbol{T} \left[\begin{array}{ccc} \mathcal{C}_1 \backslash \mathcal{C}_2 & \mathcal{C}_1 \cap \mathcal{C}_2 & \mathcal{C}_2 \backslash \mathcal{C}_1 \\ \hline \boldsymbol{Q}^{\mathcal{R}_1}_{\mathcal{C}_1 \backslash \mathcal{C}_2} & \boldsymbol{Q}^{\mathcal{R}_1}_{\mathcal{C}_1 \cap \mathcal{C}_2} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Q}^{\mathcal{R}_2}_{\mathcal{C}_1 \cap \mathcal{C}_2} & \boldsymbol{Q}^{\mathcal{R}_2}_{\mathcal{C}_2 \backslash \mathcal{C}_1} \end{array}\right].$$

For $\boldsymbol{Q}^{\mathcal{R}_1 \cup \mathcal{R}_2}$ to be an orthonormal basis, its Gramian should be the identity and of the same rank as row $\boldsymbol{M}^{\mathcal{R}_1 \cup \mathcal{R}_2}$. Assuming $\boldsymbol{Q}^{\mathcal{R}_1}$ and $\boldsymbol{Q}^{\mathcal{R}_2}$ are orthonormal, the Gramian can be determined to be:

$$\boldsymbol{G} = \boldsymbol{T} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{Q}^{\mathcal{R}_1}_{:,\mathcal{C}_1 \cap \mathcal{C}_2} (\boldsymbol{Q}^{\mathcal{R}_2}_{:,\mathcal{C}_1 \cap \mathcal{C}_2})^{\mathsf{T}} \\ \boldsymbol{Q}^{\mathcal{R}_2}_{:,\mathcal{C}_1 \cap \mathcal{C}_2} (\boldsymbol{Q}^{\mathcal{R}_1}_{:,\mathcal{C}_1 \cap \mathcal{C}_2})^{\mathsf{T}} & \boldsymbol{I} \end{bmatrix} \boldsymbol{T}^{\mathsf{T}}.$$

Now, consider the full SVD of $\boldsymbol{Q}^{\mathcal{R}_1}_{:,\mathcal{C}_1 \cap \mathcal{C}_2} \boldsymbol{Q}^{\mathcal{R}_2}_{:,\mathcal{C}_1 \cap \mathcal{C}_2}{}^{\mathsf{T}} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^{\mathsf{T}}$ and $\boldsymbol{T} = \begin{bmatrix} \boldsymbol{I} & \\ -\boldsymbol{S} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{U}^{\mathsf{T}} & \\ & \boldsymbol{V}^{\mathsf{T}} \end{bmatrix}$, leading to:

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} - \boldsymbol{S}^2 \end{bmatrix}.$$

Notice that $\boldsymbol{T}$ is an invertible matrix, such that the rank of $\boldsymbol{G}$ equals that of $\boldsymbol{M}^{\mathcal{R}_1 \cup \mathcal{R}_2}$. The rank thus equals the sum of the row dimensions of $\boldsymbol{Q}^{\mathcal{R}_1}$ and $\boldsymbol{Q}^{\mathcal{R}_2}$ minus the number of singular values equal to one. Adjusting $\boldsymbol{T}$ to obtain the identity matrix as the Gramian is then straightforward, as is done in Algorithm 3, which outlines the whole procedure to obtain the basis $\boldsymbol{Q}^{\mathcal{R}_1 \cup \mathcal{R}_2}$.

The full algorithm to compute a row space basis for $\boldsymbol{M}^{\mathcal{X}_{2^l-1}}$ is then very similar to 1. One only needs to replace the 'structured intersection' step with Algorithm 3 and use $\varphi_{\boldsymbol{\sigma}}$ to re-index the columns instead of the rows. Starting the iteration with an orthonormal basis $\boldsymbol{Q}^{\{0\}}$ then results in the desired basis for the larger block-Macaulay matrix.

5.5 Numerical experiments

In this section, various numerical experiments[5] are performed to compare Algorithm 1 to the sparse, iterative implementation to compute a null space

[5]All experiments were run on a Xeon Gold 6240 CPUs@2.6 GHz core with 128 GB of RAM.

Algorithm 3 Computes basis $\boldsymbol{Q}^{\mathcal{R}_1 \cup \mathcal{R}_2}$ starting from the orthonormal bases $\boldsymbol{Q}^{\mathcal{R}_1}$ and $\boldsymbol{Q}^{\mathcal{R}_2}$.

function STRUCTURED ROW BASIS($\boldsymbol{Q}^{\mathcal{R}_1}$, $\boldsymbol{Q}^{\mathcal{R}_2}$)
$\boldsymbol{U}, \boldsymbol{S}, \boldsymbol{V} \leftarrow$ full svd of $(\boldsymbol{Q}^{\mathcal{R}_1}_{:,\mathcal{C}_1 \cap \mathcal{C}_2} (\boldsymbol{Q}^{\mathcal{R}_2}_{:,\mathcal{C}_1 \cap \mathcal{C}_2})^{\mathsf{T}})$
indices $\leftarrow \{i \in \{1 \dots \dim\} \mid \boldsymbol{S}_{i,i} \neq 1\}$
$\boldsymbol{V} \leftarrow \boldsymbol{V}(:, \text{indices})$
$\boldsymbol{R} \leftarrow \boldsymbol{S}(\text{indices}, \text{indices})$
$\boldsymbol{S} \leftarrow \boldsymbol{S}(:, \text{indices})$
$\boldsymbol{Q} \leftarrow \begin{bmatrix} \boldsymbol{Q}^{\mathcal{R}_1}_{:,\mathcal{C}_1 \setminus \mathcal{C}_2} & \boldsymbol{Q}^{\mathcal{R}_1}_{:,\mathcal{C}_1 \cap \mathcal{C}_2} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Q}^{\mathcal{R}_2}_{:,\mathcal{C}_1 \setminus \mathcal{C}_2} & \boldsymbol{Q}^{\mathcal{R}_2}_{:,\mathcal{C}_2 \setminus \mathcal{C}_1} \end{bmatrix}$
$\boldsymbol{T} \leftarrow \begin{bmatrix} \boldsymbol{I} & \\ & (\boldsymbol{I} - \boldsymbol{R}^2)^{-\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & \\ -\boldsymbol{S} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{U}^{\mathsf{T}} & \\ & \boldsymbol{V}^{\mathsf{T}} \end{bmatrix}$
return $\boldsymbol{T}\boldsymbol{Q}$
end function

basis in MacaulayLab, which is outlined in [191]. For each experiment, both the run time and the Frobenius norm of the residual error are reported. The residual error is defined as the product between the computed null space basis and the associated block-Macaulay matrix, that is $\left\| \boldsymbol{M}^{\mathcal{X}} \boldsymbol{N}^{\mathcal{X}} \right\|_{\mathrm{F}}$.

First, some experiments involving randomly generated MEPs are presented, after which we switch to randomly generated problems from an application in globally optimal system identification. For the rank decisions involved, a relative tolerance of 10^{-9} is used. Note that, since the complexity of these problems, most notably the number of null space rows, often increases combinatorially in the degree and dimension of the problem, it is not always possible to push these parameters very far. Moreover, we are mostly interested in investigating the scaling of the approach w.r.t. these parameters, so we opt for smaller problems, which can be tackled over a wider variety of one of these parameters. We focus on the null space construction, as this is most often used for root-finding [188].

5.5.1 General multiparameter eigenvalue problems

Note that the number of rows of the MEPs is always set to $m + q - 1$ for the following experiments.

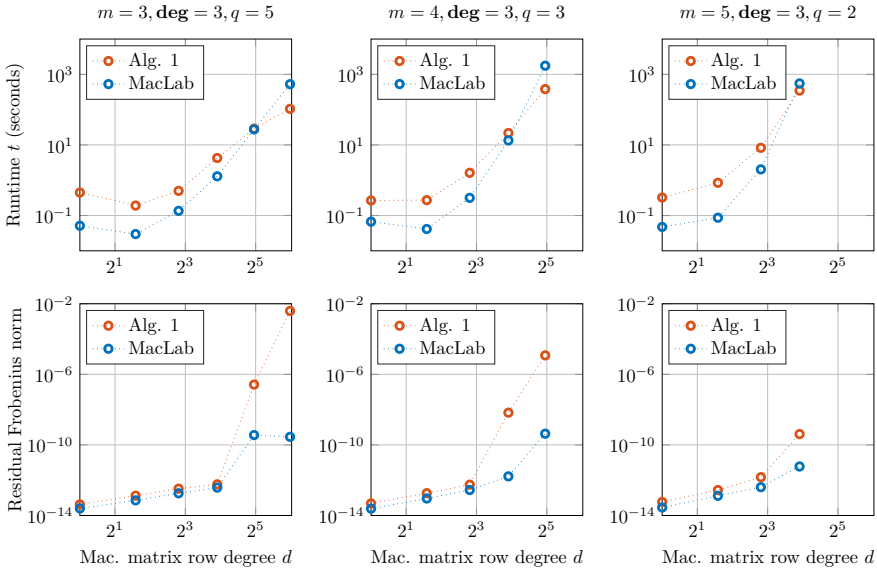


Figure 5.2: Comparison of the runtime and residuals of Algorithm 1 to MacaulayLab's sparse null space implementation for Macaulay matrices $\boldsymbol{M}^{\mathcal{X}_d}$ of increasing degree d, associated with random MEPs.

Increasing Macaulay matrix degree

This set of experiments computes the null space of block-Macaulay matrices associated with MEPs with a variety of dimensions m, number of columns q, and MEP degrees. The null spaces $\boldsymbol{N}^{\mathcal{X}_d}$ of the corresponding block-Macaulay matrices are computed for increasing total degrees d, using both Algorithm 1 and the MacaulayLab implementation. The results concerning the runtime and residuals are reported in Figure 5.2.

For low degrees, Algorithm 1 performs worse in terms of runtime compared to the other implementation, while the accuracy is comparable. Indeed, for all three sets of experiments, the computational load of the novel approach seems to scale better with the Macaulay matrix row degree d. Moreover, the crossover point slightly decreases as m increases.

For the residuals, it may be noted that, while the accuracy of both algorithms deteriorates for increasing d, at some point, that of Algorithm 1 drops significantly faster with d. The resulting bases may not be considered accurate enough, especially in the $m = 3$ and $m = 4$ cases. However, inspecting the data reveals that the MacaulayLab implementation returns the wrong dimension of

null space: Macaulaylab implementation returns a nullity of 1215 for $d = 15$ and of 983 for $d = 31$, whereas the nullity of 1 stabilizes to 1215, as one would expect. We must emphasize that there is not enough data to draw any strong conclusions on the trends in computational load. Especially d should be pushed further to see if there are no anomalies for larger sizes. It can, however, be concluded that there exist cases of null space problems for which the presented approach is of interest, albeit with the use of higher-precision arithmetic or other ways to increase the accuracy.

This setup is not very indicative of a practical root-finding scenario, however. Here, one only needs to compute the null space up to a limited degree, i.e., until the gap is found.

Increasing MEP degree

In this set of experiments, random MEPs were generated for a selection of dimensions m and number of columns q, while varying the degree of the problems themselves. To simulate the use of the null space for the purpose of root finding, we first determine the degree d^* for which the null space basis $\boldsymbol{N}^{\mathcal{X}_d}$ exhibits a gap [44, 188]. Then, we use Algorithm 1 and the MacaulayLab implementation to determine a null space basis $\boldsymbol{N}^{\mathcal{X}_d}$ of total degree d of at least the previously determined gap degree, i.e., $d \geq d^*$ [191]. Only the latter results are timed, and Figure 5.3 reports the runtime and residuals.

Overall, there does not seem to be a scaling advantage for Algorithm 1 for an increasing MEP degree. While both lines inch slightly closer, in the $m = 5$ cases, and very subtly in the $m = 3$ case, this does not persist in the $m = 4$ case. Moreover, the occasionally larger 'jumps' in the computation time of the novel algorithm negate this effect, which may also be expected for larger degrees in the $m = 5$ case.

A contributing factor to these jumps is most likely that the novel algorithm computes the null spaces associated with the row multi-indices $\mathcal{X}_{2^l-1} = \{\boldsymbol{\rho} \in \mathbb{N}_0^m | \|\boldsymbol{\rho}\| \leq 2^l - 1\}$, where l is the total number of iterations. That is, the row degree is roughly doubled in each iteration, which may result in a significant overshoot of the desired d^*. For example, in the $m = 4$, degree 9 case, there is a gap at the Macaulay matrix row degree 20, while Algorithm 1 computes a null associated with a row degree of 31. To further illustrate this 'overshoot', the number of rows of the computed null space bases is summarized in 5.1 for the larger degrees. This table indicates that the jump in timings at degree 7 for $m = 4$ coincides with a large jump in the number of rows as well, as 5 instead of 4 iterations l of Algorithm 1 are required to achieve the required degree. A similar jump from 3 to 4 iterations l can be observed for the $m = 3$ case at the

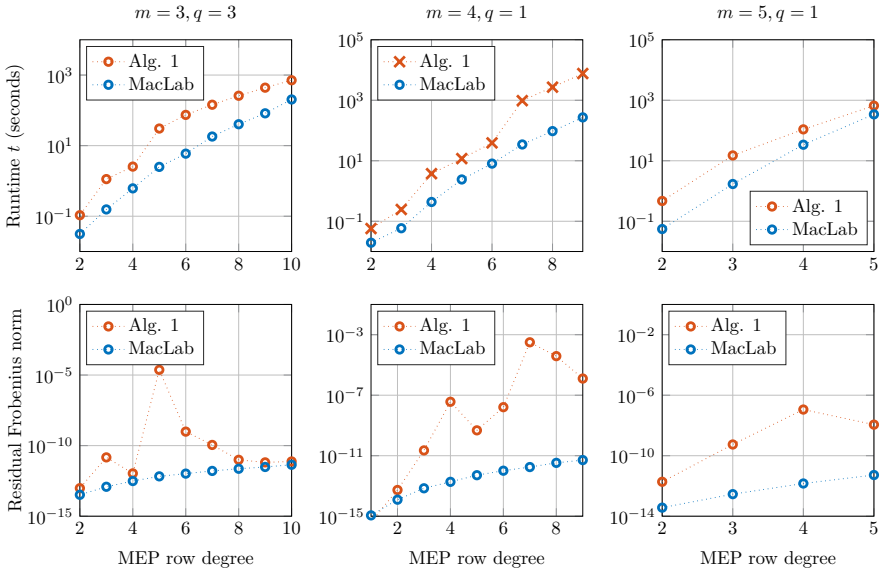


Figure 5.3: Comparison of the runtime and residual of Algorithm 1 to MacaulayLab's sparse null space implementation, for random MEPs of varying degrees.

transition from degree 4 to 5. In the $m = 5$, degree 5 case, there is no overshoot, so that the computation times are much closer. We expect a similar 'jump' for higher degrees[6], though.

The accuracy of the obtained null space also seems generally worse overall, rather erratic, and at times even insufficient. Here, the overshoot may also be a contributing factor, as we observed in the previous set of experiments that for high row degrees, both algorithms had a significant drop in accuracy. Again, to really study the asymptotic effects and draw stronger conclusions, this parameter value should ideally be pushed further.

Increasing dimension

This set of experiments considers an almost identical setup as the preceding experiment, though now we consider multivariate polynomials of a varying number of independent variables m. The remainder of the experiment setup remains unaltered. The runtimes and residuals are presented in Figure 5.4.

[6]Such cases could not be computed due to memory constraints.

	Alg. 1			MacLab		
MEP deg	$m = 3$	$m = 4$	$m = 5$	$m = 3$	$m = 4$	$m = 5$
4	4620	8855	42504	3990	2380	20349
5	27417	10626	53130	7800	5985	53130
6	29640	12650		10962	10626	
7	31980	111930		16368	17550	
8	34440	123410		21420	27405	
9	37023	135751		27417	40920	
10	39732			37023		

Table 5.1: Number of rows of the null space bases obtained through Algorithm 1 and the MacaulayLab implementation for varying MEP degrees. The associated row degrees may be uniquely derived from this table and are thus not reported.

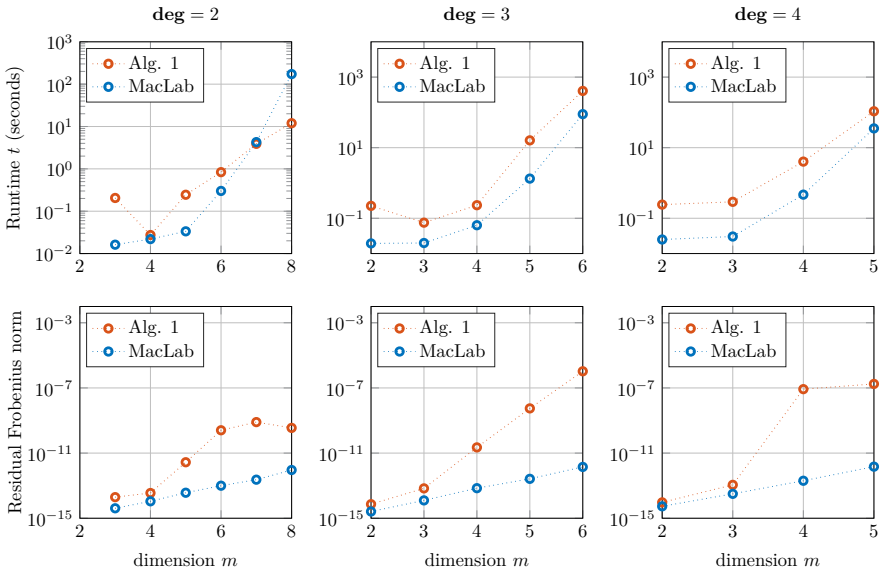


Figure 5.4: Comparison of the runtime and residual errors of Algorithm 1 to MacaulayLab's sparse null space implementation, for random polynomials of varying dimension.

	Alg. 1			MacLab		
m	deg= 2	deg= 3	deg= 4	deg= 2	deg= 3	deg= 4
3	56	286	8855	35	120	2380
4	126	1001	42504	126	715	20349
5	2002	33649		462	4368	
6	5005	134596		1716	27132	
7	11440			6435		
8	24310			24310		

Table 5.2: Number of rows of the null space bases obtained through Algorithm 1 and the MacaulayLab implementation for a varying number of independent variables. The associated row degrees can be derived from this table and are thus not reported.

As in the previous set of experiments, there does not seem to be a general scaling advantage w.r.t. the problem dimension m for either approach, and the novel approach generally performs worse. For the degree 2 case, the novel approach does seem to scale better, for the limited data available. As such, we can conclude that cases do exist where the presented approach is advantageous in terms of computational load. Looking at the number of rows of the computed matrices in Table 5.2, one observes that the overshoot for degree 2 is limited, and even absent in the last iteration, explaining the positive results. The results for the degree 4 case also move closer in the last iteration, where the ratio between the number of columns decreases to roughly 2.

Again, the numerical accuracy seems to decline more strongly with the problem dimension for the novel approach.

5.5.2 Application: globally optimal identification of autonomous LTI models

MEPs arise in the globally optimal identification of linear, time-invariant (LTI) models. In this section, the presented algorithm is applied to the system identification problem described in [103, 104]. It concerns the globally optimal least-squares misfit identification of autonomous LTI models, the output signal $\hat{y}_k$ of which adheres to a difference equation: $\sum_{i=0}^{n} a_i \hat{y}_{k+i} = 0$. Here, one tries to find the difference equation with the output that most closely describes the

given data y_k, where $k \in \{0, \ldots, N-n\}$:

$$\begin{aligned} &\min_{a_i, \hat{y}_k} \sum_{k=0}^{N} (y_k - \hat{y}_k)^2, \\ &\text{subject to } \sum_{i=0}^{n} a_i \hat{y}_{k+i} = 0, \forall k \in \{0, \ldots, N-n\}. \end{aligned} \tag{5.10}$$

The local minima of this problem can be shown to be solutions of the following equations:

$$\mathbf{0} = \begin{bmatrix} \boldsymbol{T}^{\boldsymbol{a}}_{N-n}\boldsymbol{y} & \boldsymbol{T}^{\boldsymbol{a}}_{N-n}(\boldsymbol{T}^{\boldsymbol{a}}_{N-n})^{\mathsf{T}}(\boldsymbol{T}^{\boldsymbol{a}}_{N-2n})^{\mathsf{T}} \end{bmatrix} \begin{bmatrix} -1 \\ \boldsymbol{g} \end{bmatrix}, \tag{5.11}$$

where $\boldsymbol{T}^{\boldsymbol{a}}_{q}$ is a $q \times N$ Toeplitz matrix formed with the difference equation coefficients $\boldsymbol{a} = \begin{bmatrix} a_0 & \ldots & a_n \end{bmatrix}$ as its rows, $\boldsymbol{y} = \begin{bmatrix} y_1 & \ldots & y_N \end{bmatrix}$ and $\boldsymbol{g} \in \mathbb{C}^{N-2n}$. To obtain an MEP, consider $\boldsymbol{v}^{\mathsf{T}} = \begin{bmatrix} -1 & \boldsymbol{g}^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}}$ as the eigenvector. Solving (5.11) thus provides a deterministic way to obtain the model that globally minimizes the objective function.

We now solve (5.10) for various parameter configurations and randomly generated data instances $\boldsymbol{y}$. More precisely, we first investigate the effect of increasing the number of data N. Next, an increasing model order n is considered. Both sets of experiments are organized in the same way as the preceding two.

Increasing data

In this first set of experiments, problems of increasing data size N are considered for various configurations. The results are presented in Figure 5.5.

For the order $n = 1$ and $n = 2$ cases, the novel method performs generally worse than MacaulayLab's implementation in terms of computational load, and seems comparable in terms of numerical accuracy. The scaling with N of both approaches is very similar. Inspecting the 'gap degrees' for these problems, we see that this is rather stable and hovers between 24 and 28 for the second-order case and between 4 and 5 for the first-order case. By the initial experiment 5.2, we can mainly expect a scaling advantage for increasing row degrees, which is not the case.

For the third order case, the 'gap degree' does vary as follows: $8, 36, 48$, and 63, so that Algorithm 1 resorts to a total degree of 66 for the last three problems, partly explaining why it seems to scale better for increasing N. However,

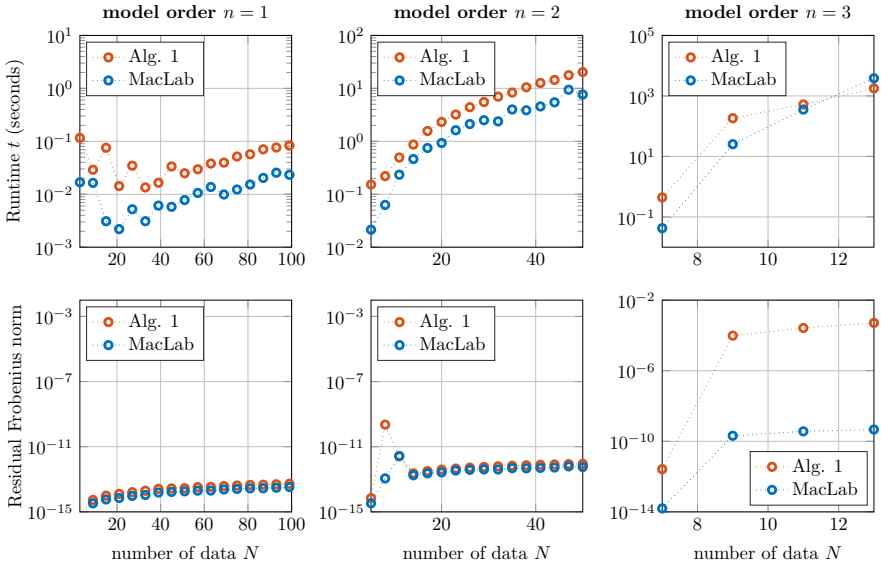


Figure 5.5: Comparison of the runtime and residual errors of Algorithm 1 to MacaulayLab's sparse null space implementation, for random polynomials of varying dimension.

due to the aforementioned overshoot, we may expect large jumps as in Figure 5.3 as additional iterations are required when N rises further. Again, this does demonstrate the existence of cases where the approach can be beneficial regarding computational complexity. Despite this limited runtime improvement, the numerical accuracy in this case is lackluster for practical use.

Increasing model order

In this first set of experiments, problems of increasing model order n are considered for various configurations.

This experiment is closely related to that of the increasing dimension, presented in Figure 5.4, that is, the number of independent variables increases. The results also seem similar, as neither algorithms demonstrates a clear scaling advantage over this limited range of m, while the accuracy drops much more strongly for the novel approach. In the last iterations, Algorithm 1 moves closer to the MacaulayLab implementation in all cases, performing slightly better in the second case. This may again be (partly) explained by a more favorable overshoot: In the second case, for example, the ratio between the

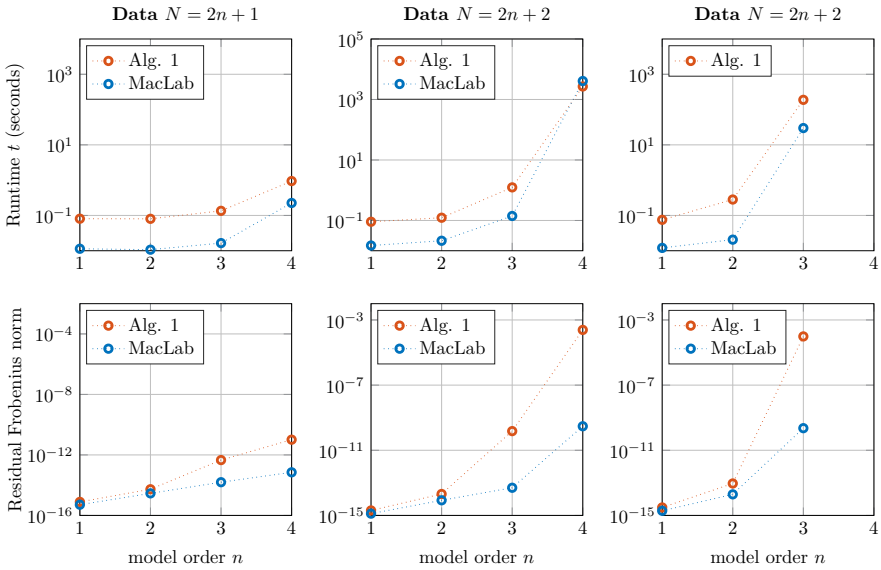


Figure 5.6: Comparison of the runtime and residual errors of Algorithm 1 to MacaulayLab's sparse null space implementation, for random polynomials of varying dimension.

first dimensions of the bases is 3.6538 in the second-to-last problem, while it is 1.4098 in the last problem.

For this particular problem, there are solutions at infinity, which cause the nullity to keep increasing for row higher degrees, rather than stabilize. This imposes an additional penalty in the case of overshoot. For example, for the $n = 4, N = 2n + 2$ problem, the bases have 15456 and 11892 columns for the novel algorithm and the MacaulayLab implementation, respectively.

5.5.3 General discussion

The method generally seems to have an overhead compared to the state-of-the-art for less complex problems. A possible reason for this initial increase in computational load is that multiple null spaces are being computed in the presented algorithm. Intuitively, the main advantage is that this set of null spaces can be reused to determine similar null spaces of roughly double the size. So, the larger these 'building blocks' get, the larger the advantage of their reuse, as reflected in the first set of experiments in Figure 5.2. As such, one

needs problems with a high enough 'gap degree' to benefit from Algorithm 1. However, this also entails a greater risk of overshooting the required degree, as each successive iteration increases the total row degree more and more.

This overshooting problem is hard to avoid in practical root finding, as, to the best of the author's knowledge, no accurate estimates exist for this degree for general MEPs. For well-known problems, such an estimate may be available, so that one can more efficiently target the desired degree. A strategy to perform this targeting could be a hybrid approach, where during (earlier) iterations, block rows are added incrementally, as done in the MacaulayLab implementation [188], to increase the size of the current 'building blocks' of the decomposition. This allows one to reach block sizes different from $2^l - 1$.

Remark that there may be other factors than the overshooting at play. However, due to the combinatorial scaling in computational complexity, it is hard to push the parameter values far enough to problems to see the scaling behavior. We can state that there seem to be cases where the novel approach is slightly beneficial in terms of run time, mostly for higher complexity problems. Moreover, the accuracy often drops strongly as the problem complexity grows.

5.6 Conclusion and future work

We presented a novel approach to compute a null space basis of a block-Macaulay matrix, avoiding the explicit construction of said matrix itself. This approach relies on leveraging the shift-invariant structure of this class of matrices to reuse previously computed bases by joining them using intersection problems. An algorithm to exploit the structure of this intersection problem is presented, as well as disjoint block-row decomposition, which allows for a recursive computation. The approach is also extended to the row space of the block-Macaulay matrix by replacing the intersection operation with a direct sum.

We investigated the efficacy of the null space computation algorithm through numerical experiments. From the limited data presented, the algorithm does seem to scale better w.r.t. an increasing Macaulay matrix degree. However, for practical root-finding, one is only interested in bases for a limited, though potentially large matrix degree. As such, there does not seem to be an inherent scaling advantage in terms of the dimension, degree or size of the solved MEPs. There are parameter configurations where the novel approach does provide an advantage in terms of computational load, though these seem scarce. It is also observed that the accuracy of the null space often becomes lackluster for larger problem instances. A contributing factor for both these problems is that the

recursive nature of the algorithm tends to overshoot the required degree, leading to larger matrices than required. The accuracy issues may be alleviated through higher-precision arithmetic, at the cost of additional computational overhead. A stability analysis may also be beneficial.

Nonetheless, while the current applicability to numerical root finding remains limited, a novel framework that allows recursive computations for matrices exhibiting shift-invariant structure was presented. This algorithm may be of use for other applications, such as polynomial elimination [10], or other classes of matrices with a similar structure. Moreover, alternative row decompositions may prove better for root-finding applications. If an accurate estimate for the gap degree is available, the decomposition may even be tailored to said degree.

To draw more meaningful conclusions on the asymptotic complexity, a theoretical analysis may be beneficial. A challenge herein is obtaining good estimates of the nullities of the Macaulay-like matrices associated with the various row multi-index sets. The practical use of an asymptotic analysis may be debated, as the combinatorial scaling makes it hard to tackle such asymptotic cases.

A further way to reduce the computational load might be to exploit the similarity in structure between the different null space bases that are computed. These objects are currently treated as unrelated bases, though they are null spaces of very similar Macaulay-like matrices. Finally, the possibility of parallelism can be explored to compute the different null space bases in each iteration.

Chapter 6

Direct Computation of Multiplication Maps

Chapter abstract

This work presents a novel algorithm to compute multiplication maps for numerical multivariate polynomial root finding. The algorithm relies on insights from border and Gröbner basis methods to reduce the size of intermediate matrices compared to other numerical approaches. Contrary to the former methods, the presented method is agnostic about the underlying polynomials that are being represented. Instead, it only relies on coefficient matrices that express these unknown polynomials as linear combinations of some unknown underlying polynomial basis. To achieve this basis-agnostic approach, a rank-based check involving only the coefficient matrices is introduced to decide upon the number of underlying basis elements to add in each iteration, leading to a purely numerical linear-algebra-based approach. While a full-blown proof of convergence remains an open research question, brief numerical experiments are provided to showcase the method.

We would like to thank Prof. Benoît Legat for his useful insights and his collaboration on this topic.

Big picture

The current chapter provides a novel algorithm to compute the homogeneous multiplication maps, given a set of multivariate polynomial equations. This may also be viewed as a more direct transition from a set of difference equations to the state-transition matrices, and thus, a state-space representation for the model class presented in Chapter 2. As it accounts for the roots at infinity of the polynomials, it can handle descriptor models as well. The algorithm relies on a shift-equation similar to the one encountered in the realization algorithms discussed in Chapter 2. In this chapter, we focus on the root-finding perspective, and note that the method is applicable to the globally optimal system identification approaches for LTI models found in [9, 36, 104, 189]. Like any algebraic root-finding algorithm, it can be applied to other polynomial optimization problems as well, as long as the solution set is finite.

Remark that, for a convenient exposition, there are some slight changes in the notation. As the objects underlying the rows and columns of the coefficient matrices in this chapter are now general polynomials, rather than only monomials, we label them as such, instead of only indexing them with the associated multi-exponent of a monomial (multi-indices).

6.1 Introduction

The current work proposes a novel algorithm for numerical multivariate polynomial root-finding. For this class of problems, there is a broad variety of techniques that can roughly be subdivided in two classes: homotopy-based methods [24, 192] and algebraic methods. This work focuses on the latter class of approaches, which most often rely on a set of joint eigenvalue problems to recover the common roots using linear algebra-techniques [169]. Within this class of techniques, there still exists a multitude of different approaches, each with its own strengths and weaknesses, a brief overview of which is presented in the following section [176].

The current work presents a novel algebraic root-finding approach for polynomial problems with a finite number of projective roots. The proposed method aims at combining insights from both border basis-style algorithms [92, 133, 136, 137, 169, 181] and resultant-based methods [12, 44, 47, 63, 123, 177, 178]. However, contrary to these methods, the proposed approach relies on a rank decision to decide on the number of basis polynomials to add in each iteration.

This property essentially eliminates any bookkeeping about the underlying polynomials, resulting in a flexible algorithm that solely acts on the coefficient matrices of these implicit polynomials through the well-established techniques of numerical linear algebra.

First, a brief overview of numerical, algebraic polynomial root-finding techniques is presented in Section 6.2, after which the main iteration and the initialization of the novel method is discussed in Sections 6.3 and 6.4. Some parallels with border basis methods are drawn in the latter section as well. Section 6.5 then presents numerical experiments to investigate the performance of the novel algorithm. Finally, an overall conclusion is drawn, and future challenges are identified in Section 6.6.

6.2 Eigenvalue-based root finding

This section elaborates on the basics underpinning eigenvalue-based polynomial root finding, that is, multiplication matrices and reduction by a set of polynomials. We first focus on how to construct the multiplication maps from the perspective of border basis techniques and then briefly discuss the variations on this approach that exist in the literature.

6.2.1 Multiplication maps

Let us start with an example to sketch the border basis approach and illustrate the general idea of eigenvalue-based root finding.

Example 6.2.1. Consider the following set of polynomial equations $f(x, y) = 1 + x + y = 0$; $g(x, y) = 2 - 3x + x^2 = 0$. New equations can then be derived through polynomial combinations of these seed equations, e.g., $0 = g(x, y) - xf(x, y) = 2 - 4x - xy$. These equations define linear relations between the monomials $\{1, x, y, x^2, xy\}$ evaluated in the common roots of f and g. Moving the 1- and x-terms to the other side of these three equations then allows them to be written compactly in the matrix format below, where the trivial equations $1 = 1$ and $x = x$ are also added:

$$\begin{bmatrix} 1 \\ x \\ y \\ x^2 \\ xy \end{bmatrix} = \begin{bmatrix} & 1 & x \\ 1 & 1 & 0 \\ x & 0 & 1 \\ y & -1 & -1 \\ x^2 & -2 & 3 \\ xy & 2 & -4 \end{bmatrix} \begin{bmatrix} 1 \\ x \end{bmatrix}. \tag{6.1}$$

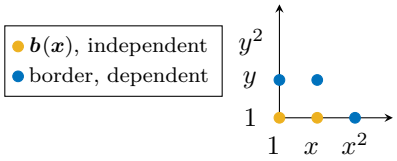


Figure 6.1: Visual representation of the linearly independent and dependent monomials in Example 6.2.1. The border elements can be written as linear combinations of the independent monomials $\boldsymbol{b}(\boldsymbol{x})$. By recursively applying these relations, any monomial multiple of the border can thus be represented in terms of these independent monomials.

This format essentially expresses all the involved monomials as linear combinations of the basis monomials $\boldsymbol{b}(\boldsymbol{x})^\mathsf{T} = \begin{bmatrix}1 & x\end{bmatrix}$, which are thus linearly independent. In the terminology of border basis techniques, the set of linearly dependent monomials $\{y, x^2, xy\}$ is called the *border*, as visualized in Figure 6.1 [169].

Since the border fully comprises both $x\boldsymbol{b}(\boldsymbol{x})$ and $y\boldsymbol{b}(\boldsymbol{x})$, the relations captured in (6.1) can be applied recursively, so that any monomial can be expressed as a linear combination of the monomials in $\boldsymbol{b}(\boldsymbol{x})$. For example: $y^2 = y(y) = y(-1-x) = -y - xy = -(-1-x) - 2 = -1 + x$. Such a 'reduction' onto the border can be straightforwardly extended to any monomial, leading to a representation in terms of the basis elements $\boldsymbol{b}(\boldsymbol{x})$, at least, when evaluated in the common roots. By linearity, this property naturally extends to any polynomial as well.

Since these relations allow all multiples of $\boldsymbol{b}(\boldsymbol{x})$ to be expressed as a linear combination of the basis elements $\boldsymbol{b}(\boldsymbol{x})$, multiplication by x and y becomes a linear map from the elements of $\boldsymbol{b}(\boldsymbol{x})$ onto itself. This linear map can be captured in matrix format, leading to so-called multiplication matrices $\boldsymbol{A}_1, \boldsymbol{A}_2 \in \mathbb{R}^{2\times 2}$:

$$\begin{bmatrix} x \\ x^2 \end{bmatrix} = \begin{matrix} 1 & x \end{matrix} \\ \begin{bmatrix} 0 & 1 \\ -2 & 3 \end{bmatrix} \begin{bmatrix} 1 \\ x \end{bmatrix} = \boldsymbol{A}_1 \begin{bmatrix} 1 \\ x \end{bmatrix},$$

$$\begin{bmatrix} y \\ xy \end{bmatrix} = \begin{matrix} 1 & x \end{matrix} \\ \begin{bmatrix} -1 & -1 \\ 2 & -4 \end{bmatrix} \begin{bmatrix} 1 \\ x \end{bmatrix} = \boldsymbol{A}_2 \begin{bmatrix} 1 \\ x \end{bmatrix}.$$

Notice that the rows of these matrices simply correspond to the coefficients of $x\boldsymbol{b}(\boldsymbol{x})$ and $y\boldsymbol{b}(\boldsymbol{x})$ w.r.t. $\boldsymbol{b}(\boldsymbol{x})$, respectively. In this way, these matrices essentially substitute the elements of $\boldsymbol{b}(\boldsymbol{x})$ by linear expressions of the appropriate multiples w.r.t. $\boldsymbol{b}(\boldsymbol{x})$, thereby expressing multiplication through acting on

the coefficient vectors:

$$x(1+2x) = \begin{bmatrix}1 & 2\end{bmatrix}\begin{bmatrix}x\\x^2\end{bmatrix} = \begin{bmatrix}1 & x\end{bmatrix}\begin{bmatrix}1 & x\\0 & 1\\-2 & 3\end{bmatrix}\begin{bmatrix}1\\x\end{bmatrix} = \begin{bmatrix}1 & x\end{bmatrix}\begin{bmatrix}1 & x\\-4 & 7\end{bmatrix}\begin{bmatrix}1\\x\end{bmatrix}.$$

Finally, since $xy = yx$, it necessarily holds that $\boldsymbol{A}_1\boldsymbol{A}_2 = \boldsymbol{A}_2\boldsymbol{A}_1$, i.e., such multiplication matrices must commute. Moreover, it can be verified that the multiplication maps can be diagonalized using the same similarity transform. i.e., there exists a $\boldsymbol{V} : \boldsymbol{V}\boldsymbol{A}_1\boldsymbol{V}^{-1} = \mathrm{diag}(1,2)$ and $\boldsymbol{V}\boldsymbol{A}_2\boldsymbol{V}^{-1} = \mathrm{diag}(-2,-3)$. This unveils the solutions of $f(x,y)$ and $g(x,y)$: $(1,-2)$ and $(2,-3)$ [29]. ◇

Example 6.2.1 briefly illustrates the general principle of eigenvalue-based root-finding methods: Given a set of $m-$variate polynomials $f_1(\boldsymbol{x}), \ldots, f_s(\boldsymbol{x})$, in the variables $\boldsymbol{x} = (x_1, \ldots, x_m) \in \mathbb{C}^m$, new polynomials in the ideal $I = \{\sum_{i=1}^{s} h_i f_i \mid h_i(\boldsymbol{x}) \in \mathbb{C}[\boldsymbol{x}]\}$ are first derived by considering 'shifts' of the current set of equations, as well as linear combinations thereof. This process is repeated until a suitable representation of I is obtained. From this representation, a basis $\boldsymbol{b}(\boldsymbol{x})$ for the so-called coordinate ring $\mathbb{C}[\boldsymbol{x}]/I$ can be constructed. Note that whenever the common roots of $f_1, \ldots, f_s$, counted with multiplicity, are finite in number, $\mathbb{C}[\boldsymbol{x}]/I$ forms a vector space of dimension equal to this number [32, 169]. This was observed in Example 6.2.1 as well, where there are two common roots.

Once such a basis $\boldsymbol{b}(\boldsymbol{x})$ is obtained, the multiplication maps can be constructed by considering the coefficients of the elements of $x_i\boldsymbol{b}(\boldsymbol{x})$ w.r.t. $\boldsymbol{b}(\boldsymbol{x})$: $x_i\boldsymbol{b}(\boldsymbol{x}) = \boldsymbol{A}_i\boldsymbol{b}(\boldsymbol{x})$. At the core of this approach lies the fact that the eigenvalues of these multiplication maps then coincide with specific coordinates of the common roots. We briefly elaborate on this: Let $\boldsymbol{b}(\boldsymbol{x}) \in \mathbb{C}^q[\boldsymbol{x}]$ contain the basis polynomials in any order. By construction, it then holds that $x_i\boldsymbol{b}(\boldsymbol{x}) = \boldsymbol{A}_i\boldsymbol{b}(\boldsymbol{x})$ for any common root $\boldsymbol{x}$, such that x_i is clearly an eigenvalue of $\boldsymbol{A}_i$, with an eigenvector $\boldsymbol{b}(\boldsymbol{x})$. Conversely, for any eigenvalue λ_j of $\boldsymbol{A}_i$ and a corresponding left eigenvector $\boldsymbol{v}_j^\mathsf{T}$, it holds that $x_i\boldsymbol{v}_j^\mathsf{T}\boldsymbol{b}(\boldsymbol{x}) = \boldsymbol{v}_j^\mathsf{T}\boldsymbol{A}_i\boldsymbol{b}(\boldsymbol{x}) = \lambda_j\boldsymbol{v}_j^\mathsf{T}\boldsymbol{b}(\boldsymbol{x})$ so that $(x_i - \lambda_j)\boldsymbol{v}_j^\mathsf{T}\boldsymbol{b}(\boldsymbol{x}) = \boldsymbol{0}$. Since $\boldsymbol{v}_j^\mathsf{T}\boldsymbol{b}(\boldsymbol{x})$ is a nonzero polynomial in $\mathbb{C}[\boldsymbol{x}]/I$, it does not lie in I, and thus does not vanish in all common roots $\boldsymbol{x}$ at once. Hence, there must exist at least one common root $\boldsymbol{x}$ for which $x_i = \lambda_j$ [169]. In conclusion, the eigenvalues of the multiplication map $\boldsymbol{A}_i$ coincide precisely with the ith coordinates of the common roots.

Finally, since the multiplication maps commute by construction, they share common eigenvectors. More precisely, for each eigenvalue of $\boldsymbol{A}_i$, there is at least one common eigenvector among the other maps $\boldsymbol{A}_j$ [143]. In general, a joint Schur decomposition exploits this structure and allows the eigenvalues of the multiplication maps to be paired into $m-$tuples, which correspond to the common roots of the considered equations [29].

6.2.2 Variations on algebraic root finding

The way in which new equations are obtained distinguishes different algebraic root-finding techniques: For example, the current equations may be shifted and subsequently reduced to a suitable representation w.r.t. $\boldsymbol{b}(\boldsymbol{x})$ through a reduced row echelon form of the corresponding coefficient matrix, leading to a new set of generating equations for I. Such approaches are prevalent in Gröbner and border basis techniques [92, 133, 136, 137, 169, 181]. In resultant-based methods, the considered equations are shifted using predetermined shifts or until some stopping criterion is satisfied, without ever reducing the obtained equations or updating the set of generators for I [12, 44, 47, 63, 177, 178]. A drawback of the latter class of approaches is that it often leads to very large coefficient matrices, which entail a significant computational load.

On the other hand, the latter class of methods often do not construct the polynomial basis for $\mathbb{C}[\boldsymbol{x}]/I$ explicitly, focusing only on the coefficient matrices instead. This allows for flexibility in the choice basis and allows one to (implicitly) delegate its construction to standard numerical linear algebra techniques, which often benefits numerical stability [177]. Indeed, to construct the ith multiplication map, it suffices to know a full column-rank coefficient matrix of a set of polynomials $\boldsymbol{p}(\boldsymbol{x})$ w.r.t. this unknown basis, as well as the coefficient matrix of $x_i\boldsymbol{p}(\boldsymbol{x})$ w.r.t. the same basis. However, one does not need to bookkeep what said basis is precisely. Instead, an overdetermined linear system can be formulated to find $\boldsymbol{A}_i$, which can subsequently be tackled with standard numerical linear algebra techniques. [44]. Example 6.2.2 briefly illustrates this approach, common in the Macaulay and truncated normal form frameworks [12, 44, 177, 178].

Example 6.2.2. Reconsider the equations from Example 6.2.1. It was already derived that the relations $y^2 = -1 + x$ and $xy = 2 - 4x$ also hold in the common roots, such that we can set up the following equations. We focus on the coefficient matrices, abstracting some of the underlying polynomials through the row labels:

$$\begin{array}{c} \\ x \\ x^2 \\ xy \end{array}\!\!\begin{array}{c} \begin{array}{cc} 1 & x \end{array} \\ \begin{bmatrix} 0 & 1 \\ -2 & 3 \\ 2 & -4 \end{bmatrix} \end{array}\!\!\begin{bmatrix} 1 \\ x \end{bmatrix} = \begin{array}{c} \\ 1 \\ x \\ y \end{array}\!\!\begin{array}{c} \begin{array}{cc} 1 & x \end{array} \\ \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & -1 \end{bmatrix} \end{array}\!\!\boldsymbol{A}_1 \begin{bmatrix} 1 \\ x \end{bmatrix},$$

$$\begin{array}{c} \\ y \\ xy \\ y^2 \end{array}\!\!\begin{array}{c} \begin{array}{cc} 1 & x \end{array} \\ \begin{bmatrix} -1 & -1 \\ 2 & -4 \\ -1 & 1 \end{bmatrix} \end{array}\!\!\begin{bmatrix} 1 \\ x \end{bmatrix} = \begin{array}{c} \\ 1 \\ x \\ y \end{array}\!\!\begin{array}{c} \begin{array}{cc} 1 & x \end{array} \\ \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & -1 \end{bmatrix} \end{array}\!\!\boldsymbol{A}_2 \begin{bmatrix} 1 \\ x \end{bmatrix}.$$

Evaluating the equation above in the common roots $(1,-2)$ and $(2,-3)$ results in the matrix equations:

$$\begin{array}{c} \\ x \\ x^2 \\ xy \end{array}\!\!\begin{array}{c} \begin{array}{cc} 1 & x \end{array} \\ \begin{bmatrix} 0 & 1 \\ -2 & 3 \\ 2 & -4 \end{bmatrix} \end{array}\!\!\begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}\!\!\begin{array}{c} 1 \\ x \end{array} = \begin{array}{c} \\ 1 \\ x \\ y \end{array}\!\!\begin{array}{c} \begin{array}{cc} 1 & x \end{array} \\ \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & -1 \end{bmatrix} \end{array}\!\!\boldsymbol{A}_1\!\!\begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}\!\!\begin{array}{c} 1 \\ x \end{array}$$

$$\begin{array}{c} \\ y \\ xy \\ y^2 \end{array}\!\!\begin{array}{c} \begin{array}{cc} 1 & x \end{array} \\ \begin{bmatrix} -1 & -1 \\ 2 & -4 \\ -1 & 1 \end{bmatrix} \end{array}\!\!\underbrace{\begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}}_{\boldsymbol{B}}\!\!\begin{array}{c} 1 \\ x \end{array} = \begin{array}{c} \\ 1 \\ x \\ y \end{array}\!\!\begin{array}{c} \begin{array}{cc} 1 & x \end{array} \\ \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & -1 \end{bmatrix} \end{array}\!\!\boldsymbol{A}_2\!\!\underbrace{\begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}}_{\boldsymbol{B}}\!\!\begin{array}{c} 1 \\ x \end{array}$$

Here, $\boldsymbol{B}$ contains the evaluations of the basis elements $\{1, x\}$ as its rows. As this is an invertible matrix, right multiplying with $\boldsymbol{B}^{-1}$ then eliminates $\boldsymbol{B}$ from these equations. This leads to a set of linear equations that can be used to determine $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$. Since only $\boldsymbol{B}$ depends on the roots and can always be eliminated from the equations, the common roots do not need to be known beforehand. This core idea holds for any set of polynomials with a zero-dimensional solutions set [169].

Let us now look at the effects of choosing a different underlying polynomial basis. It was already illustrated in Example 6.2.1 that the rows of the multiplication maps correspond to the coefficients of $x\boldsymbol{b}(\boldsymbol{x})$ and $y\boldsymbol{b}(\boldsymbol{x})$ w.r.t. the chosen basis $\boldsymbol{b}(\boldsymbol{x})$. As such, changing this basis also changes the multiplication maps. They can, however, still be determined in similar fashion, given that the necessary coefficient matrices w.r.t. the new basis are known.

For example, let $\boldsymbol{P} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$, then consider the basis $\begin{bmatrix} 1+x & 1-x \end{bmatrix} = \begin{bmatrix} 1 & x \end{bmatrix}\boldsymbol{P}^{\mathsf{T}}$ and revisit the equation defining $\boldsymbol{A}_1$:

$$\begin{array}{c} \\ x \\ x^2 \\ xy \end{array}\!\!\begin{array}{c} \begin{array}{cc} 1 & x \end{array} \\ \begin{bmatrix} 0 & 1 \\ -2 & 3 \\ 2 & -4 \end{bmatrix} \end{array}\!\!\boldsymbol{P}^{-1}\!\!\begin{bmatrix} 1+x \\ 1-x \end{bmatrix} = \begin{array}{c} \\ 1 \\ x \\ y \end{array}\!\!\begin{array}{c} \begin{array}{cc} 1 & x \end{array} \\ \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & -1 \end{bmatrix} \end{array}\!\!\boldsymbol{P}^{-1}\boldsymbol{P}\boldsymbol{A}_1\boldsymbol{P}^{-1}\!\!\begin{bmatrix} 1+x \\ 1-x \end{bmatrix}$$

$$\Leftrightarrow \begin{array}{c} \\ x \\ x^2 \\ xy \end{array}\!\!\begin{array}{c} \begin{array}{cc} 1+x & 1-x \end{array} \\ \begin{bmatrix} 0.5 & -0.5 \\ 0.5 & -2.5 \\ -1 & 3 \end{bmatrix} \end{array}\!\!\begin{bmatrix} 1+x \\ 1-x \end{bmatrix} = \begin{array}{c} \\ 1 \\ x \\ y \end{array}\!\!\begin{array}{c} \begin{array}{cc} 1+x & 1-x \end{array} \\ \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & -0.5 \\ -1 & 0 \end{bmatrix} \end{array}\!\!\boldsymbol{P}\boldsymbol{A}_1\boldsymbol{P}^{-1}\!\!\begin{bmatrix} 1+x \\ 1-x \end{bmatrix}.$$

Clearly, $\boldsymbol{P}\boldsymbol{A}_1\boldsymbol{P}^{-1}$ acts like a multiplication map w.r.t. this new basis. As this is a similarity transform of $\boldsymbol{A}_1$, it has the same eigenvalues, and thus carries the same information about the common roots. ◇

Example 6.2.2 illustrates that, to construct $\boldsymbol{A}_i$, it suffices to have vector representations of a satisfactory set of polynomials $\boldsymbol{p}(\boldsymbol{x})$, i.e., $\begin{bmatrix}1 & x & y\end{bmatrix}^\mathsf{T}$ in the example, as well as their images $x_i\boldsymbol{p}(\boldsymbol{x})$ w.r.t. the same basis. Indeed, while the multiplication maps depend on the specific basis used, their eigenvalues do not, such that we can essentially abstract the column labels to some unknown underlying basis elements, instead of considering the basis polynomials explicitly. Furthermore, since left multiplication with an invertible matrix does not affect the solutions of the linear equations for $\boldsymbol{A}_i$, the specific row labels do also not necessarily matter to obtain the multiplication maps.

In this work, these ideas are explored, aiming to further bridge the gap between the different algebraic techniques. To this end, we introduce a purely numerical linear algebra (NLA) based, basis-agnostic approach to compute the multiplication maps. Contrary to common NLA-based approaches, the presented algorithm reduces the equations onto the current basis in each iteration. Moreover, the presented approach further simplifies the 'bookkeeping' of underlying polynomials, relying solely on coefficient matrices, without explicit knowledge of the polynomials being represented. This results in a fairly straightforward algorithm.

6.3 Basis-agnostic approach

We now outline the computational steps in the main iteration of the proposed approach. While the derivations concern polynomials, all the computational steps only act upon the coefficient matrices and do not depend on the underlying polynomials. As such, none of these polynomials need to be constructed explicitly, allowing flexibility in the choice of bases. Furthermore, none of the steps rely on explicit gradings of the polynomials or tracking a suitable set of monomials, as is the case for border basis algorithms [134]. Instead, it relies on a rank decision to determine which new dimensions to add. Remark that, while the underlying polynomials are considered explicitly in this section, they can be abstracted since all the computational steps solely rely on the coefficient matrices.

Consider a set of basis polynomials $\boldsymbol{b}(\boldsymbol{x})$ and some equations $\boldsymbol{p}_1(\boldsymbol{x}) = \boldsymbol{N}_0\boldsymbol{b}(\boldsymbol{x})$. As before, these relations can be interpreted as $\boldsymbol{p}_1(\boldsymbol{x})$ being represented in terms of the basis elements $\boldsymbol{b}(\boldsymbol{x})$. Assume a similar representation for the images $x_i\boldsymbol{p}_1(\boldsymbol{x}) = \boldsymbol{N}_i\boldsymbol{b}(\boldsymbol{x})$ for all i is given. Since expressions for $x_i\boldsymbol{p}_1(\boldsymbol{x})$ w.r.t. the current basis $\boldsymbol{b}(\boldsymbol{x})$ are known, row $\boldsymbol{N}_0$ can be thought of as the $\boldsymbol{b}(\boldsymbol{x})$-coefficient space representing polynomials whose images under multiplication with x_i can again be represented in terms of the current 'working basis' $\boldsymbol{b}(\boldsymbol{x})$. Its orthogonal

complement is then the coefficient space of polynomials with multiples that cannot yet be expressed in terms of $\boldsymbol{b}(\boldsymbol{x})$.

We now aim to derive a 'richer' set of equations that includes expressions for all first-degree multiples of $\boldsymbol{b}(\boldsymbol{x})$. By the given relations, the multiples of a polynomial $\boldsymbol{v}^\mathsf{T}\boldsymbol{b}(\boldsymbol{x}) = (\boldsymbol{v}_1 + \boldsymbol{v}_2)^\mathsf{T}\boldsymbol{b}(\boldsymbol{x})$, where $\boldsymbol{v}_1 \in \operatorname{row}\boldsymbol{N}_0$ and $\boldsymbol{v}_2 \in \operatorname{null}\boldsymbol{N}_0$, are of the form $x_i\boldsymbol{v}^\mathsf{T}\boldsymbol{b}(\boldsymbol{x}) = \boldsymbol{v}_3^\mathsf{T}\boldsymbol{b}(\boldsymbol{x}) + \boldsymbol{v}_2^\mathsf{T}x_i\boldsymbol{b}(\boldsymbol{x})$, with $\boldsymbol{v}_3 \in \operatorname{row}\boldsymbol{N}_i$. Since the current equations do not provide an expression for $\boldsymbol{v}_2 x_i\boldsymbol{b}(\boldsymbol{x})$ in terms of $\boldsymbol{b}(\boldsymbol{x})$, we extend the working basis to $\begin{bmatrix}\boldsymbol{b}(\boldsymbol{x})^\mathsf{T} & x_i\boldsymbol{p}_2(\boldsymbol{x})^\mathsf{T}\end{bmatrix}$, where $\boldsymbol{p}_2(\boldsymbol{x}) = \boldsymbol{Q}\boldsymbol{b}(\boldsymbol{x})$ and $\boldsymbol{Q}^\mathsf{T}$ is a basis for $\operatorname{null}\boldsymbol{N}_0$. Note that the number of basis elements added thus follows from $\operatorname{rank}\boldsymbol{N}_0$.

One can then define the maps $\boldsymbol{A}_i$ as follows:

$$\begin{aligned} x_i\begin{bmatrix}\boldsymbol{N}_0\\ \boldsymbol{Q}\end{bmatrix}\boldsymbol{b}(\boldsymbol{x}) = x_i\begin{bmatrix}\boldsymbol{p}_1(\boldsymbol{x})\\ \boldsymbol{p}_2(\boldsymbol{x})\end{bmatrix} &= \begin{bmatrix}\boldsymbol{N}_i & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{I}\end{bmatrix}\begin{bmatrix}\boldsymbol{b}(\boldsymbol{x})\\ x_i\boldsymbol{p}_2(\boldsymbol{x})\end{bmatrix} \\ \Rightarrow x_i\boldsymbol{b}(\boldsymbol{x}) = \begin{bmatrix}\boldsymbol{N}_0{}^\dagger & \boldsymbol{Q}^\dagger\end{bmatrix}\begin{bmatrix}\boldsymbol{N}_i & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{I}\end{bmatrix}\begin{bmatrix}\boldsymbol{b}(\boldsymbol{x})\\ x_i\boldsymbol{p}_2(\boldsymbol{x})\end{bmatrix} &= \boldsymbol{A}_i\begin{bmatrix}\boldsymbol{b}(\boldsymbol{x})\\ x_i\boldsymbol{p}_2(\boldsymbol{x})\end{bmatrix}. \end{aligned} \tag{6.2}$$

This results in new equations that express the coefficients of $x_i\boldsymbol{b}(\boldsymbol{x})$ w.r.t. the larger working basis $\begin{bmatrix}\boldsymbol{b}(\boldsymbol{x})^\mathsf{T} & x_i\boldsymbol{p}_2(\boldsymbol{x})^\mathsf{T}\end{bmatrix}$, allowing us to express multiples of any polynomial spanned by the elements of $\boldsymbol{b}(\boldsymbol{x})$. Indeed, the maps $\boldsymbol{A}_i$ express multiplication w.r.t. $\boldsymbol{b}(\boldsymbol{x})$-coefficients in the following sense:

$$x_i\boldsymbol{v}^\mathsf{T}\boldsymbol{b}(\boldsymbol{x}) = \boldsymbol{v}^\mathsf{T}x_i\boldsymbol{b}(\boldsymbol{x}) = \boldsymbol{v}^\mathsf{T}\boldsymbol{A}_i\begin{bmatrix}\boldsymbol{b}(\boldsymbol{x})\\ x_i\boldsymbol{p}_2(\boldsymbol{x})\end{bmatrix} = \begin{bmatrix}(\boldsymbol{A}_i)_1 & (\boldsymbol{A}_i)_2\end{bmatrix}\begin{bmatrix}\boldsymbol{b}(\boldsymbol{x})\\ x_i\boldsymbol{p}_2(\boldsymbol{x})\end{bmatrix}.$$

Also remark that $\boldsymbol{A}_i$ may map the coefficients $\boldsymbol{v}^\mathsf{T}$ to a higher-dimensional vector space, to account for the new basis polynomials that were added to express the unknown images of $\boldsymbol{p}_2(\boldsymbol{x})$.

Now, the multiplication property of the matrices $\boldsymbol{A}_i$ can be used to construct new linear relations between the basis elements $\boldsymbol{b}(\boldsymbol{x})$ and $x_i\boldsymbol{p}_2(\boldsymbol{x})$ as follows:

$$\begin{aligned}\mathbf{0} =& x_j\left(x_i\boldsymbol{b}(\boldsymbol{x}) - \boldsymbol{A}_i\begin{bmatrix}\boldsymbol{b}(\boldsymbol{x})\\ x_i\boldsymbol{p}_2(\boldsymbol{x})\end{bmatrix}\right) = \boldsymbol{A}_j\begin{bmatrix}x_i\boldsymbol{b}(\boldsymbol{x})\\ x_ix_j\boldsymbol{p}_2(\boldsymbol{x})\end{bmatrix} - \boldsymbol{A}_i\begin{bmatrix}x_j\boldsymbol{b}(\boldsymbol{x})\\ x_ix_j\boldsymbol{p}_2(\boldsymbol{x})\end{bmatrix}\\ =&(\boldsymbol{A}_j)_1x_i\boldsymbol{b}(\boldsymbol{x}) + (\boldsymbol{A}_j)_2x_jx_i\boldsymbol{p}_2(\boldsymbol{x}) - (\boldsymbol{A}_i)_1x_j\boldsymbol{b}(\boldsymbol{x}) - (\boldsymbol{A}_i)_2x_jx_i\boldsymbol{p}_2(\boldsymbol{x})\\ =&((\boldsymbol{A}_j)_1\begin{bmatrix}(\boldsymbol{A}_i)_1 & (\boldsymbol{A}_i)_2 & \mathbf{0}\end{bmatrix} - (\boldsymbol{A}_i)_1\begin{bmatrix}(\boldsymbol{A}_j)_1 & \mathbf{0} & (\boldsymbol{A}_j)_2\end{bmatrix})\begin{bmatrix}\boldsymbol{b}(\boldsymbol{x})\\ x_i\boldsymbol{p}_2(\boldsymbol{x})\\ x_j\boldsymbol{p}_2(\boldsymbol{x})\end{bmatrix}\\ =&\begin{bmatrix}(\boldsymbol{C}_{i,j})_0 & (\boldsymbol{C}_{i,j})_1 & (\boldsymbol{C}_{i,j})_2\end{bmatrix}\begin{bmatrix}\boldsymbol{b}(\boldsymbol{x})\\ x_i\boldsymbol{p}_2(\boldsymbol{x})\\ x_j\boldsymbol{p}_2(\boldsymbol{x})\end{bmatrix} = \boldsymbol{C}_{i,j}\begin{bmatrix}\boldsymbol{b}(\boldsymbol{x})\\ x_i\boldsymbol{p}_2(\boldsymbol{x})\\ x_j\boldsymbol{p}_2(\boldsymbol{x})\end{bmatrix},\end{aligned} \tag{6.3}$$

where the fourth equality holds as $(\boldsymbol{A}_i)_2 = (\boldsymbol{A}_j)_2, \forall i,j \in \{1,\dots,m\}$ by their construction in (6.2). The elements of $\begin{bmatrix}\boldsymbol{b}(\boldsymbol{x})^\mathsf{T} & x_i\boldsymbol{p}_2(\boldsymbol{x})^\mathsf{T} & x_j\boldsymbol{p}_2(\boldsymbol{x})^\mathsf{T}\end{bmatrix}$ thus satisfy any of the linear relations in $\operatorname{row}\boldsymbol{C}_{i,j}$. By these new relations between the basis elements, they can be written as linear combinations of a smaller number of polynomials whenever $\boldsymbol{C}_{i,j}$ is nonzero.

To construct such a new basis, consider $\boldsymbol{C}$ defined below, which captures the relations over generated by all (i,j)-pairs. To this end, define the set of pairs $\mathcal{A} = \{(i,j) \in \{1\dots,m\}^2 \mid j > i\}$, and then construct the block-matrix $\boldsymbol{C}$ using row multi-indices $\boldsymbol{\alpha} \in \mathcal{A}$:

$$\begin{gathered}\forall\boldsymbol{\alpha} \in \mathcal{A} : \boldsymbol{C}_{\boldsymbol{\alpha},1} = (\boldsymbol{C}_{\boldsymbol{\alpha}})_0;\ \boldsymbol{C}_{\boldsymbol{\alpha},\alpha_1} = (\boldsymbol{C}_{\boldsymbol{\alpha}})_1;\ \boldsymbol{C}_{\boldsymbol{\alpha},\alpha_2} = (\boldsymbol{C}_{\boldsymbol{\alpha}})_2,;\ \boldsymbol{C}_{\boldsymbol{\alpha},\{1,\dots,m\}\setminus\alpha} = \mathbf{0},\\ \boldsymbol{T} = \operatorname{basis}(\operatorname{null}\boldsymbol{C}).\end{gathered} \tag{6.4}$$

Equation (6.3) then implies that the current basis polynomials can be represented by vectors in range $\boldsymbol{T}$. Indeed, let the rows of $\boldsymbol{R}$ form a basis for $\operatorname{row}\boldsymbol{C}$ and

define the following polynomials:

$$\begin{bmatrix} \boldsymbol{R} \\ \boldsymbol{T}^{\dagger} \end{bmatrix} \begin{bmatrix} \boldsymbol{b}(\boldsymbol{x}) \\ x_1 \boldsymbol{p}_2(\boldsymbol{x}) \\ \vdots \\ x_m \boldsymbol{p}_2(\boldsymbol{x}) \end{bmatrix} \overset{(6.3)}{=} \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{b}_{\text{new}}(\boldsymbol{x}) \end{bmatrix}$$
$$\Leftrightarrow \begin{bmatrix} \boldsymbol{R}^{\dagger} & \boldsymbol{T} \end{bmatrix} \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{b}_{\text{new}}(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{T}_0 \\ \boldsymbol{T}_1 \\ \vdots \\ \boldsymbol{T}_m \end{bmatrix} \boldsymbol{b}_{\text{new}}(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{b}(\boldsymbol{x}) \\ x_1 \boldsymbol{p}_2(\boldsymbol{x}) \\ \vdots \\ x_m \boldsymbol{p}_2(\boldsymbol{x}) \end{bmatrix}. \tag{6.5}$$

More generally, the left inverse of $\boldsymbol{T}$ is only defined up the left kernel of $\boldsymbol{T}$, so that in theory, the basis $\boldsymbol{b}_{\text{new}}(\boldsymbol{x})$ can be chosen modulo the relations in $\operatorname{row} \boldsymbol{C}$.

Finally, by (6.2), it then holds that:

$$x_i \boldsymbol{b}(\boldsymbol{x}) = \boldsymbol{A}_i \begin{bmatrix} \boldsymbol{b}(\boldsymbol{x}) \\ x_i \boldsymbol{p}_2(\boldsymbol{x}) \end{bmatrix} = \boldsymbol{A}_i \begin{bmatrix} \boldsymbol{T}_0 \\ \boldsymbol{T}_i \end{bmatrix} \boldsymbol{b}_{\text{new}}(\boldsymbol{x}) = \boldsymbol{N}_i \boldsymbol{b}_{\text{new}}(\boldsymbol{x}), \tag{6.6}$$

leaving us with matrix representations for both $x_i \boldsymbol{b}(\boldsymbol{x})$, $\boldsymbol{b}(\boldsymbol{x})$ and all $x_i \boldsymbol{p}_2(\boldsymbol{x})$ w.r.t. the new working basis $\boldsymbol{b}_{\text{new}}(\boldsymbol{x})$. This allows us to construct maps again through (6.2) by choosing, $\boldsymbol{p}_1(\boldsymbol{x}) = \boldsymbol{b}(\boldsymbol{x})$, $\boldsymbol{N}_0 = \boldsymbol{T}_0$ and $\boldsymbol{N}_i$ as defined in (6.6), so that the whole procedure can be repeated. The computational steps are summarized in Algorithm 4 in Section 6.4, where the convergence condition of the algorithm is discussed.

6.4 Initialization and convergence criterion

In this section, the behavior of the procedure presented in Section 6.3 is demonstrated using a running example. Note that we operate under the assumption that the initial set of polynomials has a finite number of projective solutions. First, an adequate initialization of the algorithm is illustrated, which is of major importance to ensure convergence. We emphasize that a convergence proof remains an active research question. Nonetheless, valuable insights are provided for the case where monomial bases are chosen. Intuitively, the general case can then be achieved through a change of basis.

6.4.1 Initialization

By the second part of (6.5), it holds that whenever $\boldsymbol{b}(\boldsymbol{x}), \boldsymbol{p}_2(\boldsymbol{x})$ consist of monomials, $\boldsymbol{b}_{\text{new}}(\boldsymbol{x})$ does as well if $\boldsymbol{T}^\mathsf{T}$ is chosen to be in reduced row echelon form: The identity rows of $\boldsymbol{T}$ select some set of monomials out of those in $\boldsymbol{b}(\boldsymbol{x}), x_1\boldsymbol{p}_2(\boldsymbol{x}), \ldots, x_m\boldsymbol{p}_2(\boldsymbol{x})$. The use of this particular null space basis to obtain a monomial basis is briefly sketched in Example 6.4.1, which illustrates the initialization of the algorithm as well.

Example 6.4.1. Consider the following set of bivariate polynomial equations:

$$f(x,y) = 1 + 3x + y + 2y^2 = 0,$$

$$g(x,y) = 1 + 2x + y + 3x^2 + xy + 3y^2 + xy^2 + 3y^3 = 0.$$

To enforce the relations implied by $f(x,y)$ on all third-degree monomials, the shifts $xf(x,y)$ and $yf(x,y)$ are considered as well. In border basis terms, this allows the polynomials to be fully 'reduced' among themselves first, avoiding ambiguity on how to reduce the linearly dependent monomials, and ensuring that their reductions are well defined [134].

These equations can be summarized in the following coefficient matrix, a so-called Macaulay matrix [12]. Remark that from here on, we adopt a notation that better highlights the role of the coefficient matrices and the implicit nature of the underlying polynomial objects, which are only indicated using row and column labels, as in Examples 6.2.1 and 6.2.2:

$$\boldsymbol{M} = \begin{array}{r} \\ f(x,y) \\ xf(x,y) \\ yf(x,y) \\ g(x,y) \end{array} \begin{array}{c} \begin{array}{cccccccccc} 1 & x & y & x^2 & xy & y^2 & x^3 & x^2y & xy^2 & y^3 \end{array} \\ \begin{bmatrix} 1 & 3 & 1 & & & 2 & & & & \\ & 1 & & 3 & 1 & & & & 2 & \\ & & 1 & & 3 & 1 & & & & 2 \\ 1 & 2 & 1 & 3 & 1 & 3 & & & 2 & 1 \end{bmatrix} \end{array}. \tag{6.7}$$

Taking a basis $\boldsymbol{N}$ for the null space of $\boldsymbol{M}$ is a straightforward way to obtain the linear relations between the different basis monomials, as implied by the equations described in the Macaulay matrix (6.7). By a similar argument as in (6.5), the dimension of this null space indicates the number of basis elements needed to describe all the currently considered polynomials $\boldsymbol{p}(\boldsymbol{x})$, i.e., monomials up to degree three. Here, $\boldsymbol{N}^\mathsf{T}$ is chosen in reduced row-echelon form to guarantee that the underlying basis $\boldsymbol{b}(\boldsymbol{x})$, i.e., the set of column labels of $\boldsymbol{N}$, consists of

monomials:

$$\boldsymbol{N} = \begin{array}{c} \\ 1 \\ x \\ y \\ x^2 \\ xy \\ y^2 \\ x^3 \\ x^2y \\ xy^2 \\ y^3 \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & x & y & x^2 & x^3 & x^2y \end{array} \\ \begin{bmatrix} 1 & & & & & \\ & 1 & & & & \\ & & 1 & & & \\ & & & 1 & & \\ -\frac{1}{6} & -\frac{11}{6} & -\frac{1}{2} & & & \\ -\frac{1}{2} & -\frac{3}{2} & -\frac{1}{2} & & & \\ & & & & 1 & \\ & & & & & 1 \\ \frac{1}{12} & \frac{5}{12} & \frac{1}{4} & -\frac{3}{2} & & \\ \frac{1}{2} & \frac{7}{2} & \frac{1}{2} & & & \end{bmatrix} \end{array}.$$

In general, the choice of null space basis only affects the (implicit) underlying polynomial basis $\boldsymbol{b}(\boldsymbol{x})$, as already pointed out in Example 6.2.2.

Among these monomials we can now select $\boldsymbol{p}_1(\boldsymbol{x})^\mathsf{T} = \begin{bmatrix} 1 & x & y & x^2 & xy & y^2 \end{bmatrix}$, as all their multiples have known vector representations. Notice that the images of the third-degree monomials are not immediately known. However, since xy^2 and y^3 can be expressed in terms of $\boldsymbol{p}_1(\boldsymbol{x})$, their images under multiplication are linear combinations of those of $\boldsymbol{p}_1(\boldsymbol{x})$, which are known.

The multiples of the monomials x^3 and x^2y are not known, however, as is revealed using any basis $\boldsymbol{Q}$ for the null space of[1] $\boldsymbol{N}_{\mathcal{I},:}$, with $\mathcal{I}_0 = \{1, x, y, x^2, xy, y^2\}$, that is, the rows of $\boldsymbol{N}$ corresponding to the elements of $\boldsymbol{p}_1(\boldsymbol{x})$, e.g.:

$$\boldsymbol{Q} = \begin{array}{c} \\ x^3 \\ y^3 \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & x & y & x^2 & x^3 & x^2y \end{array} \\ \begin{bmatrix} & & & & 1 & \\ & & & & & 1 \end{bmatrix} \end{array}.$$

Note the similarity with choosing taking the kernel of $\boldsymbol{T}_0$ to choose $\boldsymbol{p}_2(\boldsymbol{x})$ in (6.2). Consequently, we choose $\boldsymbol{p}_2(\boldsymbol{x})^\mathsf{T} = \boldsymbol{b}(\boldsymbol{x})^\mathsf{T}\boldsymbol{Q}^\mathsf{T}\begin{bmatrix} x^3 & x^2y \end{bmatrix}$. Now, the x-images $\{x^4, x^3y\}$ and y-images $\{x^3y, x^2y^2\}$ of these independent basis elements are introduced to define maps as in (6.2), where we choose $\boldsymbol{N}_0^\mathsf{T} = \boldsymbol{N}_{\mathcal{I}_0,:}$ and $\boldsymbol{N}_1 = \boldsymbol{N}_{\mathcal{I}_1,:}$, $\boldsymbol{N}_2 = \boldsymbol{N}_{\mathcal{I}_2,:}$, with $\mathcal{I}_1 = x \cdot \mathcal{I}_0 = \{xi \mid i \in \mathcal{I}_0\}$ and $\mathcal{I}_2 = y \cdot \mathcal{I}_0$:

[1] ':' indicates 'all indices' and for convenience, we index the rows of $\boldsymbol{N}$ using the set of the corresponding monomials, as the precise ordering of the rows does not matter in general.

$$\boldsymbol{A}_1 = \begin{array}{c} \\ x \\ x^2 \\ xy \\ x^3 \\ x^4 \\ xy^3 \end{array} \begin{array}{c} \begin{array}{cccccccc} 1 & x & y & x^2 & x^3 & y^3 & x^4 & x^3y \end{array} \\ \left[\begin{array}{cccccccc} -0.0097 & 0.9129 & -0.0290 & 0.2130 & & 0.1162 & & \\ 0.0238 & 0.2139 & 0.0713 & 0.4771 & & -0.2852 & & \\ -0.1631 & -1.8016 & -0.4894 & -0.0775 & & -0.0423 & & \\ & & & & 1 & & & \\ & & & & & & 1 & \\ & & & & & & & 1 \end{array}\right] \end{array}$$

$$\boldsymbol{A}_2 = \begin{array}{c} \\ y \\ xy \\ y^2 \\ x^2y \\ x^3y \\ y^4 \end{array} \begin{array}{c} \begin{array}{cccccccc} 1 & x & y & x^2 & x^3 & y^3 & x^3y & x^2y^2 \end{array} \\ \left[\begin{array}{cccccccc} -0.0549 & -0.4293 & 0.9129 & -0.1743 & & & & \\ -0.0320 & -0.7796 & -0.2861 & 0.4278 & & & & \\ -0.4800 & -1.3439 & -0.4683 & 0.0634 & & & & \\ & & & & & 1 & & \\ & & & & & & 1 & \\ & & & & & & & 1 \end{array}\right] \end{array}$$

Notice that the rows of $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ provide different vector representations for $\{x, y, x^2, xy, y^2\}$ compared to the ones provided by $\boldsymbol{N}$. This discrepancy will now be addressed next for the general case. Moreover, note that contrary to border basis methods, lower degree monomials are mapped onto higher degree ones. ◇

6.4.2 Reducing the equations

To reduce the number of equations under consideration, in each iteration, the equations $\boldsymbol{p}_1(\boldsymbol{x}) = \boldsymbol{N}_0\boldsymbol{b}(\boldsymbol{x})$ and $x_i\boldsymbol{p}_1(\boldsymbol{x}) = \boldsymbol{N}_i\boldsymbol{b}(\boldsymbol{x})$, are 'compressed' into the shift equations (6.2). These equations capture how the basis elements are mapped under multiplication, modulo the known equations. Similar to the approach in the Macaulay framework [12, 44], this compression can be achieved through solving a linear equation for $\boldsymbol{A}_i$. However, such methods often only perform this step at the end to obtain the final multiplication maps, while here, this step is performed at each iteration to obtain new equations.

Notice the similarity with border basis methods, where only the descriptions of the border elements, i.e., the first-degree multiples of the basis elements, are tracked [133]. Indeed, as illustrated in Example 6.1, these elements suffice to express all polynomials as a linear combination of the basis elements. By analogy, this suggests that, given an appropriate selection of $\boldsymbol{b}(\boldsymbol{x})$, the 'shift equations',

$$\boldsymbol{0} = x_i\boldsymbol{b}(\boldsymbol{x}) - \boldsymbol{A}_i \begin{bmatrix} \boldsymbol{b}(\boldsymbol{x}) \\ x_i\boldsymbol{p}_2(\boldsymbol{x}) \end{bmatrix},$$

generated in (6.2) capture the essential relations to express multiplication modulo the ideal in each iteration.

Equation (6.2) does seem to entail a loss of information, as the right-hand-side equations do not imply the left-hand-side ones. However, due to linear relations between the elements of $\boldsymbol{p}_1(\boldsymbol{x})$, the representation of $x_i\boldsymbol{p}_1(\boldsymbol{x})$ is not unique: Let the rows of $\boldsymbol{L}^\mathsf{T} = \begin{bmatrix}\boldsymbol{L}_1^\mathsf{T} & \boldsymbol{L}_2^\mathsf{T}\end{bmatrix}$ form an orthonormal basis for the null space of $\begin{bmatrix}\boldsymbol{N}_0{}^\mathsf{T} & \boldsymbol{Q}^\mathsf{T}\end{bmatrix}$. Then from the left-hand-side equation of (6.2), it follows that:

$$\boldsymbol{0} = \boldsymbol{L}^\mathsf{T}\begin{bmatrix}\boldsymbol{N}_i & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{I}\end{bmatrix}\begin{bmatrix}\boldsymbol{b}(\boldsymbol{x})\\ x_i\boldsymbol{p}_2(\boldsymbol{x})\end{bmatrix} = \boldsymbol{Z}\begin{bmatrix}\boldsymbol{b}(\boldsymbol{x})\\ x_i\boldsymbol{p}_2(\boldsymbol{x})\end{bmatrix}.$$

As $\boldsymbol{Z}$ is generally nonzero, this describes nontrivial linear relations among the working basis $\begin{bmatrix}\boldsymbol{b}(\boldsymbol{x})^\mathsf{T} & x_i\boldsymbol{p}_2(\boldsymbol{x})^\mathsf{T}\end{bmatrix}$. By the nature of the pseudoinverse it holds that $\operatorname{null}\begin{bmatrix}\boldsymbol{N}_0{}^\dagger & \boldsymbol{Q}^\dagger\end{bmatrix} = \operatorname{row}\boldsymbol{L}$. Taking this into account, the right hand side of (6.2) implies the left-hand-side of equation (6.2), up to the linear relations spanned by $\boldsymbol{Z}$:

$$\begin{aligned}
x_i\boldsymbol{p}(\boldsymbol{x}) &= \begin{bmatrix}\boldsymbol{N}_0\\ \boldsymbol{Q}\end{bmatrix}\boldsymbol{A}_i\begin{bmatrix}\boldsymbol{b}(\boldsymbol{x})\\ x_i\boldsymbol{p}_2(\boldsymbol{x})\end{bmatrix}\\
&= \left(\begin{bmatrix}\boldsymbol{N}_0\\ \boldsymbol{Q}\end{bmatrix}\boldsymbol{A}_i + \boldsymbol{L}^\mathsf{T}\boldsymbol{Z}\right)\begin{bmatrix}\boldsymbol{b}(\boldsymbol{x})\\ x_i\boldsymbol{p}_2(\boldsymbol{x})\end{bmatrix}\\
&= \begin{bmatrix}\boldsymbol{N}_0 & \boldsymbol{L}_1\\ \boldsymbol{Q} & \boldsymbol{L}_2\end{bmatrix}\begin{bmatrix}\boldsymbol{N}_0{}^\dagger & \boldsymbol{Q}^\dagger\\ \boldsymbol{L}_1^\mathsf{T} & \boldsymbol{L}_2^\mathsf{T}\end{bmatrix}\begin{bmatrix}\boldsymbol{N}_i & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{I}\end{bmatrix}\begin{bmatrix}\boldsymbol{b}(\boldsymbol{x})\\ x_i\boldsymbol{p}_2(\boldsymbol{x})\end{bmatrix} = \begin{bmatrix}\boldsymbol{N}_i & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{I}\end{bmatrix}\begin{bmatrix}\boldsymbol{b}(\boldsymbol{x})\\ x_i\boldsymbol{p}_2(\boldsymbol{x})\end{bmatrix}.
\end{aligned}$$

This observation shows that there is some room to choose the representations for $x_i\boldsymbol{b}(\boldsymbol{x})$, as these are only defined up to the relations in the ideal. As such, it is to be expected that the presented algorithm may make different choices in this regard compared to other approaches.

6.4.3 Convergence criterion

We now continue with the running Example to demonstrate how the new linear relations between the basis elements can be derived and how convergence may be detected afterwards.

Example 6.4.2. Let us continue with Example 6.4.1 and derive new equations using (6.3), where $(\boldsymbol{A}_i)_1 = (\boldsymbol{A}_i)_{:,\mathcal{I}_0}$, by constructing $\boldsymbol{C}$ and a basis for its null space, i.e., $\boldsymbol{T}$, again in reduced 'column echelon form' to guarantee a monomial

basis:

$$\boldsymbol{C}^{\mathsf{T}} = \begin{array}{c} \begin{array}{cccccc} 0 & 0 & 0 & 0 & 0 & 0 \end{array} \\ \left[\begin{array}{cccccc} 0.1439 & -0.0708 & 0.2524 & & & \\ 1.1181 & -0.5923 & 2.0144 & & & \\ 0.2194 & -0.1583 & 0.4484 & & & \\ 0.6776 & 0.4485 & -0.0662 & & & \\ 0.1743 & -0.4278 & -0.0634 & & & \\ 0.1355 & 0.2464 & -0.4248 & & & \\ & & & -1 & & \\ & & & 1 & & \\ 0.1162 & -0.2852 & -0.0423 & & & \end{array}\right] \end{array} \begin{array}{l} 1 \\ x \\ y \\ x^2 \\ x^3 \\ y^3 \\ x^4 \\ x^3y \\ x^3y \\ x^2y^2 \end{array},$$

$$\boldsymbol{T} = \begin{array}{c} \begin{array}{cccccc} 1 & x & y & x^3 & x^4 & x^3y \end{array} \\ \left[\begin{array}{cccccc} 1 & & & & & \\ & 1 & & & & \\ & & 1 & & & \\ -0.3148 & -2.4630 & -0.5000 & & & \\ & & & 1 & & \\ 0.6605 & 5.2654 & 1.1667 & & & \\ & & & & 1 & \\ & & & & & 1 \\ & & & & & 1 \\ -0.1728 & -1.4012 & -0.3333 & -1.5 & & \end{array}\right] \end{array} \begin{array}{l} 1 \\ x \\ y \\ x^2 \\ x^3 \\ y^3 \\ x^4 \\ x^3y \\ x^3y \\ x^2y^2 \end{array}.$$

First, notice that the fourth row of $\boldsymbol{C}$ eliminates the duplicate basis element that was added for x^3y. While it was obvious in this case that the underlying basis was of too high dimension, for more general bases, the images of different basis elements are not necessarily equal so that four new dimensions may indeed be required[2]. Furthermore, the row labels are not known in practice, so that this worst-case scenario should be considered.

The dimension of range $\boldsymbol{T}$ is six, indicating that there are again six underlying basis elements by (6.5), just like during the initialization. This stagnation of the number of basis elements indicates regularity [135]. Indeed, by Bezout's theorem, we expect there to be exactly six solutions and, as explained in Section 6.2, the coordinate ring is expected to be of the same dimension.

Remark that, had $\{xy^2, y^3\}$ been considered part of $\boldsymbol{p}_2(\boldsymbol{x})$ in Example 6.4.1, eight new dimensions would have to be added, leading to a nullity of at least eight and thus hampering the convergence of the algorithm. This demonstrates the importance of correctly choosing the set of polynomials $\boldsymbol{p}_2(\boldsymbol{x})$ with 'unknown images'. ◇

[2]Consider the case where $\boldsymbol{p}_2(\boldsymbol{x})$ comprises x^3 and y^3, so that x^4, x^3y, xy^3 and y^4 must be added to the basis.

Let us comment on the equations that are implicitly captured in $\boldsymbol{C}$. Reconsider the equations defined by row $\boldsymbol{C}$ in (6.3). Notice that these are multiples of the shift equations (6.2):

$$x_j \left(x_i \boldsymbol{b}(\boldsymbol{x}) - \boldsymbol{A}_i \begin{bmatrix} \boldsymbol{b}(\boldsymbol{x}) \\ x_i \boldsymbol{p}_2(\boldsymbol{x}) \end{bmatrix} \right) = \boldsymbol{0},$$

though reduced by those shift equations in subsequent steps. Again, there is a strong analogy with border basis methods, where only multiples of the shift equations are used to generate new relations, which are subsequently reduced [133, 134]. In the current approach, $\boldsymbol{T}$ then provides a way to rewrite all equations in terms of the new basis polynomials generated in (6.5).

There may be additional relations in row $\boldsymbol{C}$, however: consider a basis monomial $b(\boldsymbol{x})$ such that both $xb(\boldsymbol{x})$ and $yb(\boldsymbol{x})$ are elements of $\boldsymbol{p}_2(\boldsymbol{x})$. In such a case, too many new basis elements are added: both $yxb(\boldsymbol{x})$ and $xyb(\boldsymbol{x})$ are added in (6.2). When choosing a monomial basis, it can be worked out that[3] an equation of the form $0 = yxb(\boldsymbol{x}) - xyb(\boldsymbol{x})$ is spanned by the rows of $\boldsymbol{C}$, as previously illustrated in Example 6.4.2. As such, any such 'spurious dimension' is removed.

6.4.4 Building the multiplication maps

Finally, Example 6.4.3 below illustrates how the projective multiplication maps can be distilled once convergence is detected. Contrary to border basis algorithms, the solutions at infinity have to be taken into account in the presented approach. The full procedure is also summarized in Algorithm 4. Again, notice that the main iteration only involves standard linear algebra operations on the coefficient matrices. Moreover, note that it is not required to restrict oneself to 'reduced column-echelon form' bases for $\boldsymbol{T}$, as we did in the running example.

Example 6.4.3. Let us continue from example 6.4.2. Now that regularity has been detected, the multiplication maps can be constructed. Without going into too much detail, the solutions at infinity can be accounted for by 'homogenizing'[4] the equations. Here, one introduces a homogenization variable, e.g., z, for which there also exists a multiplication map $\boldsymbol{A}_0$ [12, 31].

To construct this map, the approach in [16] takes a generic linear combination of $\boldsymbol{N}_0 = \boldsymbol{T}_{\mathcal{J}_0,:}$, $\boldsymbol{N}_1 = \boldsymbol{A}_1\boldsymbol{T}$ and $\boldsymbol{N}_2 = \boldsymbol{A}_2\boldsymbol{T}$, e.g., $\boldsymbol{P} = \boldsymbol{N}_0 + \boldsymbol{N}_1 + 2\boldsymbol{N}_2$, where

[3] When $\boldsymbol{T}^\mathsf{T}$ is chosen in reduced row-echelon form, $\boldsymbol{Q}$ always consists of identity rows so that its pseudoinverse consists of identity rows that select these new monomials.

[4] The homogenization of a polynomial $f(x, y)$ of degree d is $f(z, x, y) = z^d f(\frac{x}{z}, \frac{y}{z})$. The affine roots (x, y) can then be obtained from the projective roots as $(1, \frac{x}{z}, \frac{y}{z})$, assuming $z \neq 0$. See Chapter 2 for details regarding solutions at infinity and homogenous multiplication maps.

$\mathcal{J}_0 = \{1, x, y, x^2, x^3, y^3\}$. In general, $\mathcal{J}_0$ corresponds to all the basis elements that were not added during the last iteration, in this case, the initialization. The maps can subsequently be defined as $\boldsymbol{A}_i = \boldsymbol{P}^\dagger \boldsymbol{N}_i$, resulting in:

$$\boldsymbol{A}_0 = \begin{array}{c} \begin{array}{cccccc} 1 & x & y & x^3 & x^4 & x^3y \end{array} \\ \left[\begin{array}{cccccc} 0.5227 & 1.0227 & -0.9773 & & & \\ -0.1591 & -0.2045 & -0.0227 & & & \\ 0.3182 & -0.4091 & 0.5000 & & & \\ -0.1406 & -1.0918 & -0.2500 & & & \\ -0.4733 & -3.6613 & -0.8106 & 1 & & \\ 0.3070 & 2.3765 & 0.5303 & & & \end{array}\right] \end{array}$$

$$\boldsymbol{A}_1 = \begin{array}{c} \begin{array}{cccccc} 1 & x & y & x^3 & x^4 & x^3y \end{array} \\ \left[\begin{array}{cccccc} -0.1591 & -0.2045 & -0.0227 & & & \\ 0.0682 & 0.3864 & 0.1136 & & & \\ 0.0455 & 0.4091 & -0.0455 & & & \\ -0.4733 & -3.6613 & -0.8106 & 1 & & \\ 3.5946 & 27.8129 & 6.1742 & -7 & 1 & -2 \\ -1.5606 & -12.0758 & -2.6818 & 3 & & 1 \end{array}\right] \end{array}$$

$$\boldsymbol{A}_2 = \begin{array}{c} \begin{array}{cccccc} 1 & x & y & x^3 & x^4 & x^3y \end{array} \\ \left[\begin{array}{cccccc} 0.3182 & -0.4091 & 0.5000 & & & \\ 0.0455 & 0.4091 & -0.0455 & & & \\ -0.1818 & 0.0000 & 0.2727 & & & \\ 0.3070 & 2.3765 & 0.5303 & & & \\ -1.5606 & -12.0758 & -2.6818 & 3 & & 1 \\ 0.6268 & 4.8496 & 1.0758 & -1.5 & & \end{array}\right] \end{array}$$

It can be verified that these maps commute. Moreover, as is expected from multiplication maps by the multidimensional Cayley–Hamilton theorem from [44], these matrices satisfy the homogenizations $f(z, x, y)$ and $g(z, x, y)$: Evaluating in the matrices results in residuals with a Frobenius norm of 5.7141e−14 and 5.7054e−14, respectively.

Applying the techniques of [29], the six projective solutions of the system can be extracted using a joint Schur decomposition, resulting in the projective common roots $(0.4913 \pm 0.6609\mathrm{i}, -0.1347 \mp 0.0125\mathrm{i}, 0.3217 \mp 0.3242\mathrm{i})$, $(-0.1645, 0.4513, 0.3566)$, and $(0, 1, 0)$ with multiplicity three. The latter multiple root lies at infinity, as indicated by the value $z = 0$. ◇

Algorithm 4 Algorithm to compute projective multiplication maps for a set of polynomial equations with a finite number of projective roots.

function MULTIPLICATION_MAPS($\boldsymbol{f}(\boldsymbol{x})$)
  $m \leftarrow \text{length}(\boldsymbol{x})$
  $d \leftarrow \deg(\boldsymbol{f}(\boldsymbol{x}))$
  $\boldsymbol{M} \leftarrow$ Macaulay matrix of $\boldsymbol{f}$ of degree d
  $\boldsymbol{N} \leftarrow \text{basis null } \boldsymbol{M}$
  $\mathcal{I}_0 \leftarrow \{\boldsymbol{x}^{\boldsymbol{\alpha}} \mid \|\boldsymbol{\alpha}\|_1 \leq d-1\}$
  $\boldsymbol{N}_0 \leftarrow \boldsymbol{N}_{\mathcal{I}_0,:}$
  $\forall i \in \{1,\ldots,m\} : \boldsymbol{N}_i \leftarrow \boldsymbol{N}_{x_i \cdot \mathcal{I}_0,:}$
  $\text{dim} \leftarrow \text{size}(\boldsymbol{N}, 2)$
  while true **do**
    $\boldsymbol{Q}^{\mathsf{T}} \leftarrow \text{basis null } \boldsymbol{N}_0$
    $\forall i \in \{1,\ldots,m\} : \boldsymbol{A}_i \leftarrow \begin{bmatrix} {\boldsymbol{N}_0}^{\dagger} & \boldsymbol{Q}^{\dagger} \end{bmatrix} \begin{array}{c} \\ \boldsymbol{N}_i \\ \boldsymbol{0} \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & \cdots & i+1 & \cdots & m+1 \end{array} \\ \begin{bmatrix} \boldsymbol{0} & \cdots & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \cdots & \boldsymbol{I} & \cdots & \boldsymbol{0} \end{bmatrix} \end{array}$
    construct $\boldsymbol{T}$ as defined in (6.4)
    $\boldsymbol{N}_0 \leftarrow \boldsymbol{T}_{1:\text{dim},:}$
    $\forall i \in \{1,\ldots,m\} : \boldsymbol{N}_i \leftarrow \boldsymbol{A}_i \boldsymbol{T}$
    if dim = size($\boldsymbol{T}$,2) **then**
      $\boldsymbol{P} \leftarrow$ random linear combination of $\boldsymbol{N}_i$
      $\forall i \in \{0,\ldots,m\} : \boldsymbol{A}_i \leftarrow \boldsymbol{P}^{\dagger} \boldsymbol{N}_i$
      return $\boldsymbol{A}_i$
    else
      $\text{dim} \leftarrow \text{size}(\boldsymbol{T}, 2)$
    end if
  end while
end function

6.5 Numerical experiments

In this section, we briefly investigate the numerical behavior of a naive MATLAB implementation of Algorithm 4. To this end, we use numerical experiments and compare it to MacaulayLab [188], a linear-algebra based MATLAB toolbox for multivariate polynomials and multiparameter eigenvalue problems.

The experiments[5] are set up as follows: each run considers a randomly generated system $\boldsymbol{p}(\boldsymbol{x})$ of m polynomial equations of degree d in $\boldsymbol{x} \in \mathbb{C}^m$. Each set of equations is solved using both Algorithm 4, where the solutions are extracted

[5]The experiments were run on a System HP EliteBook 845 G8 Notebook PC with an AMD Ryzen 7 PRO 5850U with Radeon Graphics, 1901 MHz, 8 Core and 16 Logical Processors and 32 GB of RAM.

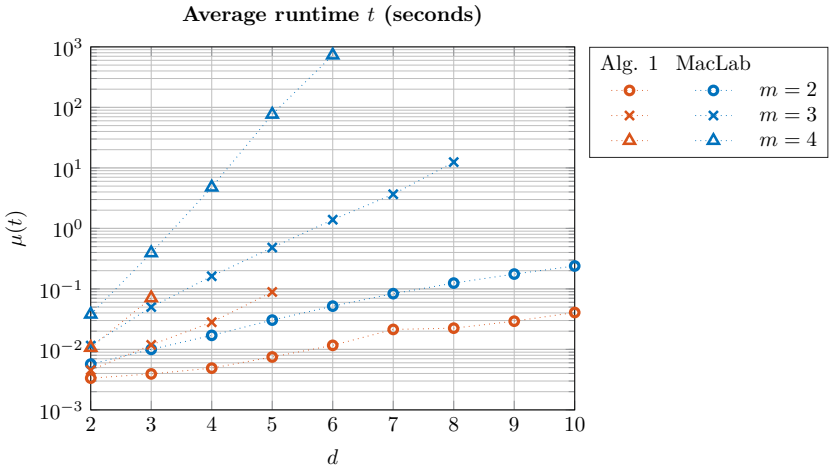


Figure 6.2: Comparison of the average runtime over five runs of solving a random system of m equations of degree d in m variables. Experiments ran up to degrees $10, 8, 6, 4$ for $m = 2, 3, 4, 5$, respectively. Missing data indicates that the algorithm failed to find all solutions on at least one of its runs.

from the multiplication maps using the techniques described in [29], and MacaulayLab. For each set of parameters (m, d), five runs are performed, the average runtime and the geometric mean of the maximal residual $\|\boldsymbol{p}(\boldsymbol{x}_k)\|_2$ across all solutions $\boldsymbol{x}_k$ are subsequently reported in Figures 6.2 and 6.3, respectively. Degrees up to $10, 8, 6$ and 4 were considered for $m = 2, 3, 4$ and 5, respectively. Whenever one of the runs failed to find all solutions, the entire parameter setting (m, d) is considered to have failed for the algorithm. In this case, no timings and residuals are incorporated in the figures for these parameter values (m, d). The relative tolerance that was used for rank decisions was 10^{-7}, that is, any singular value lower than $10^{-7}\sigma_1$, where σ_1 is the largest singular value, is considered to be zero.

First, we look at runtimes in Figure 6.2. The presented method, when it succeeds, consistently outperforms the MacaulayLab implementation. For $m = 2$, the timings seem to scale slightly better with increasing degree d, as the lines slightly

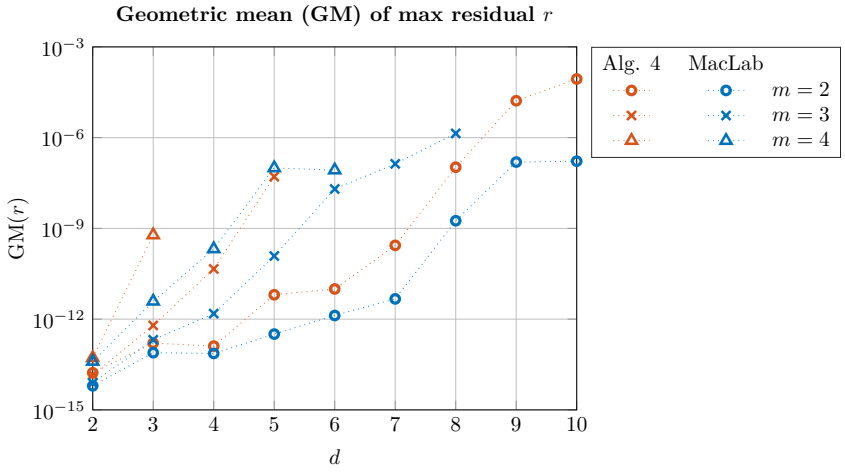


Figure 6.3: Comparison of the geometric mean of the residuals of five runs of solving a random system of m equations of degree d in m variables. Experiments ran onto degrees $10, 8, 6, 4$ for $m = 2, 3, 4, 5$, respectively. Missing data indicates that the algorithm failed to find all solutions on at least one of its runs.

diverge. There does not seem to be a noticeable scaling advantage for general m, partly due to the lack of data.

Figure 6.3 shows that the MacaulayLab implementation consistently yields more accurate results for a given problem complexity (m, d). Moreover, the accuracy of Algorithm 4 deteriorates significantly faster with increasing degrees, to the point where it cannot provide the solutions to some of the problems where the MacaulayLab implementation still can[6]. Since a convergence proof has yet to be provided, the possibility of the algorithm not converging cannot be ruled out entirely. However, the gradual loss in accuracy as the problem complexity increases suggests that numerical inaccuracies lie at the heart of the problem. The rank decisions may be impeded by these inaccuracies, which may

[6]Note that we did not attempt to push the MacaulayLab implementation to failure in terms of problem complexity.

lead to incorrect dimensions for the basis and as a consequence an incorrect, inconsistent set of equations.

As such, depending on the equations at hand, the algorithm may still succeed. For example, for $m = 4$, four out of five runs still succeed in finding the solutions with as residuals $0.13 \times 10^{-9}, 0.1587 \times 10^{-7}, 0.29 \times 10^{-9}$, and 0.06×10^{-9} in an average run time of $0.8147s$. The algorithm most likely has numerical stability issues, severely limiting its practical usefulness. This caveat should be addressed before a more elaborate comparison to existing methods is in order.

6.6 Conclusion

A novel algorithm for numerical multivariate polynomial root finding was presented. In contrast with existing approaches, the number of basis polynomials that are added in each iteration is decided using the rank of one of the coefficient matrices, rather than symbolically. This eliminates the need to construct the underlying polynomials. Instead, it relies solely on numerical linear algebra manipulations on the coefficient matrices. The algorithm does not consider any explicit ordering or grading of the polynomials either.

Even though numerical experiments suggest that the method converges for equations with a finite number of projective roots, a formal convergence proof has yet to be established. Moreover, while the abovementioned properties allow for great flexibility in the choice of underlying polynomials, numerical experiments showed that the stability of the proposed method in its current state seems lackluster at best, hindering any practical use.

Nonetheless, the additional flexibility in the underlying polynomials, reduced bookkeeping, and reliance on the well-established methods of linear algebra may provide fertile grounds for more numerically stable developments in the future. Better choices of bases may exist, for example, as well as more intricate ways to compute new relations among the basis elements. Accurately answering these questions requires a more in-depth analysis of the algorithm's stability, however.

Chapter 7

Conclusion and future outlook

As individual conclusions regarding the chapters have already been presented, we concentrate on organizing the overall contributions and their interconnections along the lines of the research objectives outlined in Chapter 1. Furthermore, we reflect on the underlying themes of the dissertation and consider the presented material within a broader scope. Finally, we turn towards the future, identifying possible paths for improvement and further research challenges.

7.1 Overview of the contributions

As motivated in Chapter 1, the overarching goal of the dissertation was to expand upon and improve the globally optimal system identification approach presented in [9, 36, 189]. We now briefly organize the contributions along the lines of the research objectives defined in Chapter 1.2.

Research objective 1

In Part I of the dissertation, more specifically, Chapter 4, a deterministic, globally optimal approach to tackle parameter estimation within a system identification context was extended to the case of finite-order, autonomous, scalar, linear, shift-invariant models. Using Wirtinger calculus to deal with the intricacies of optimization over complex decision variables, it was shown that the local minima of said problem arise as common roots of the first-order optimality conditions of a relaxation of the problem, which are essentially

polynomial equations. These results strongly depend on the polynomial nature of the models to be identified, the theoretical framework around which was explored in Chapters 2 and 3. More specifically, the former chapter provides an overview of the different ways to represent these models, while the latter chapter provides a more general parameterization of the difference equations in terms of the eigentuples of the model, similar to the characteristic polynomial for LTI models. These insights lead to two polynomial characterizations, based on the image and kernel model representations.

While the properties of affine models presented in Chapter 2 are well known, it is also shown how polynomials with roots at infinity lead to descriptor models. More specifically, it is demonstrated that such models can be straightforwardly associated with affine models through a linear coordinate transform, a technique stemming from the Algebraic geometry literature, allowing the properties of the affine models to be extended. Although not tackled explicitly, the generalized Cayley-Hamilton construction in Chapter 3 may be readily generalized to the descriptor case using these insights. This allows one to construct a difference equation representation from a given descriptor state-space model.

This simple connection between affine and descriptor systems also provides a simple means to parameterize descriptor models, allowing one to to extend the results of Chapter 4 to the descriptor case. Though some more formal care is required in this regard, we conjecture that the optimal descriptor models arise as *solutions at infinity* of the polynomial characterization presented in Chapter 4. If true, this would provide an interesting use case for computing the eigentuples at infinity, as they are often dismissed as not interesting. Moreover, the extension of the Popov-Belevitch-Hautus test presented in Chapter 2 gives theoretic insight into the minimality of said models.

Research objective 2

Chapter 4 also further explores the algebraic and system-theoretic properties of the local minima of misfit identification problems. More specifically, it was observed that the local minima always exhibit *nth order dynamics*, that is, they cannot be expressed using a lower model order than the predefined maximum. This was leveraged to simplify the polynomial equations, which in turn reduces the number of common roots of the characterizations. These observations lead to a Walsh-like interpolation condition for the misfit sequence, from which the dual, kernel-based characterization was derived.

In the LTI case, the linearity of the auxiliary variables can be exploited to transform the problem into a multiparameter eigenvalue problem, which is generally advantageous computationally [36, 104]. It was also demonstrated

that the resulting MEP is generally not equivalent to the original polynomial problem, in the sense that there are additional, spurious solutions.

Research objective 3

Chapter 4 also demonstrates that the common roots of the polynomial characterizations comprise infinite components, which hinders the convergence of eigenvalue-based root-finding techniques. Unfortunately, this limits the applicability of the algorithms presented in Part II of the dissertation concerning the multidimensional misfit problem presented in said chapter. The isolatedness of the local minima within this solution set was shown, so that Rabinowitch's trick can move these solutions to infinity, where they can be filtered out using the Macaulay method. However, this introduces additional variables, and thus considerable additional complexity [32]. As such, homotopy continuation methods are more practical for said problem. The one-dimensional instances of this problem [36, 103, 104], other globally optimal parameter estimation problems [9, 189], and globally optimal model order reduction problems [102] do not suffer from such infinite solution sets, so that the proposed algorithms can be applied. For these cases, it is especially useful to leverage their MEP structure.

While the goal of alleviating the computational complexity of the polynomial root-finding step for such globally optimal identification approaches was not clearly achieved, the algorithms presented in Part II of the dissertation do provide new perspectives in this regard, which may be further explored. Chapter 5 outlines a framework to use the shift-invariant structure of block-Macaulay matrices to compute bases for their null and row spaces, without explicitly constructing the Macaulay matrix itself by reusing intermediate basis computations. Numerical experiments have found that there exist cases for which this approach is advantageous in terms of computational load, though these cases remain scarce, and the benefit was often small. In addition, it is hard to investigate the asymptotic behavior due to the combinatorial scaling in complexity. Chapter 6 provides a novel way to compute the multiplication maps, without fully constructing the often large null space associated with a Macaulay matrix by intermediately reducing the generating equations. This method relies on a novel rank-based check to determine the number of new basis polynomials required.

However, many questions are left unanswered regarding both algorithms: The convergence of the algorithm of Chapter 6 remains unproven, and its numerical stability requires further attention before it is of practical use. Furthermore, an analysis of the asymptotic complexity of both algorithms is required to

formally investigate whether these may (theoretically) improve upon the computational load. Such a result can also lead to more insight into the cases in which these algorithms could be beneficial. Other design choices can be implemented within the framework presented in Chapter 5 as well, to make it better suited for the specific application of root finding. For example, an algorithm that operates on similar principles, but with a different block-row decomposition, was recently published [195]. The included complexity analysis suggests an improved asymptotic complexity. Moreover, the authors indicate that intermediate refinement steps using iterative algorithms can improve the accuracy. Nevertheless, both approaches provide interesting novel tools and may serve as a point of departure for further investigations.

7.2 General considerations

To correctly evaluate the importance of these results, they should, of course, be viewed within a wider context. Indeed, the existence of viable alternatives or promising new possibilities greatly affects the value of the presented insights.

7.2.1 Engineering practice

The proposed methodology demonstrates that globally optimal parameter estimation can be achieved for the misfit modeling of autonomous, scalar, linear, shift-invariant systems. This misfit framework does not rely on statistical assumptions, as it simply finds the model with the closest set of model-compliant trajectories, in a least-squares sense. With this guarantee for global optimality, this amounts to a parameter estimation framework that is free from heuristics such as local optimality.

Yet, as of now, the problem sizes that can be tackled using the proposed methodology remain too stringent for any meaningful use in a practical system identification setting. While the improvement of computational aspects may help alleviate this complexity bottleneck, it is an entirely different problem that is inherently much more complex due to the combinatorial nature of the number of solutions. The additional computational resources required are often better employed differently in practice: Either the problem size is increased, allowing for larger amounts of data, or more expressive models are fitted, e.g., by increasing the model order or choosing a broader model class altogether.

The discussion in Chapter 1.1.2 indicates that system identification includes more steps than just the parameter estimation phase. Most importantly, the

quality of a model is, in the end, determined by its ability to make reliable predictions on the system's output, given independent input data. This ability is mostly assessed during the model validation phase, which often relies on statistics. With this in mind, none of the abovementioned strategies guarantee a better result in terms of predictive capability. More data generally decreases the variance of the estimated model parameters. Though here, one is inherently limited by the quality of said data, diminishing returns in terms of variance, as well as the chosen model class, which may lead to bias [120]. More complex model classes, on the other hand, can achieve better fits, though they may be prone to 'overfitting', where the model captures the random fluctuations and disturbances rather than only the underlying system dynamics[1]. Similarly, the strategy of attaining an optimal fit during the parameter estimation phase may intuitively lead to better predictions, though it may also simply provide a better fit to the noise within the data. Improving the fit without choosing a more complex model class may have a lower risk of overfitting, however, as the model class remains inherently less expressive.

In conclusion, there is no hard evidence that globally optimal parameter estimation intrinsically leads to a better predictive capability. The process of system identification is mostly limited by the quality of the given data, which is statistical in nature. Consequently, 'optimality', be it local or global for that matter, can itself be considered a heuristic for accurate model building.

As such, it could be argued that, in the context of practical system identification, statistics and heuristics are inescapable. Another limitation in this regard is that the employed least-squares criterion is statistically most relevant in the case of Gaussian disturbances. While this is the most common type of perturbation, it may not be the most relevant for all applications. As the proposed approach strongly hinges on the polynomial nature of both the objective function and the model, it is not as easily adapted to other norms like the 1-norm.

7.2.2 Other globally optimal approaches

Other methodologies to achieve global optimality for polynomial optimization problems exist as well. The sum-of-squares (SOS) framework [105], for example, does not explicitly characterize the local minima. Instead, this framework relies on the observation that expressing a polynomial as a sum of squares of other polynomials certifies its nonnegativity. For a given polynomial basis, computing this decomposition amounts to a semidefinite program. Similarly, SOS-based certificates also exist for polynomials constrained to regions characterized by polynomial (in)equalities. Now, imposing such a nonnegativity certificate

[1] Note that precautions can be taken to prevent this, for example, by applying regularization.

on $f - \gamma$, where f is the objective function, makes sure that the auxiliary variable γ serves as a lower bound to the objective. Then, maximizing this lower bound γ still leads to a semidefinite program, which is convex by nature [20, 109]. Increasing the degree of the basis then leads to a hierarchy of convex relaxations, and an associated, monotonously increasing series of lower bounds, which converge to the global minimum [105]. Moreover, the duals of these problems form a hierarchy of semidefinite moment relaxations, relying on moment matrices from which only the global minimizers can be extracted after the convergence criterion is satisfied [78, 109]. This framework has found applications in the computation of invariant sets in constrained control [30, 96, 111, 113], determining regions of attraction [77, 95], nonlinear optimal control [106], and stability analysis [112], among others.

In comparison, the approach presented in this dissertation also characterizes stationary points other than the global minima, most notably, all local minima. Moreover, the presented characterization relies on more explicit algebraic relations between the model parameters. As such, besides a methodology for globally optimal parameter estimation, the presented approach also provides insight into the theoretical properties of the stationary points. This is valuable information that may serve to study and improve locally optimal approaches. Indeed, we demonstrated that the local minima exhibit *nth order dynamics*, leading to an interpolation condition. This demonstrates that it can be worthwhile to search for similar structure in other (globally optimal) parameter estimation approaches, e.g., those presented in [9, 189]. Such information does not emerge from the SOS approach, which may be considered a considerable disadvantage. Furthermore, in light of the 'heuristic' nature of globally optimal parameter estimation in itself, finding all local minima may also be considered beneficial, as this allows one to validate the performance of multiple potentially interesting models. Finally, while the SOS framework is computationally demanding as well, this focus on the global minima may be more advantageous in this regard, though a proper comparison is required to draw any conclusions.

While both the FONC-based approach and the SOS framework provide globally optimal methodologies for more general polynomial optimization problems, specific instances within this class admit alternative, problem-specific strategies. One example of such a problem is that of optimal low-rank tensor approximation. Here, one searches for the canonical polyadic decomposition (CPD), that is, a decomposition into rank-1 terms, of the rank R tensor closest to the given tensor in terms of Frobenius norm [37]. Under mild conditions, the CPD of a low-rank tensor is unique [166] and can be obtained through joint generalized eigenvalue problems derived from the tensor. A stable variation of this computational approach is provided by the generalized eigenspace decomposition (GESD) algorithm [50].

For the optimal approximation of 'noisy' low-rank tensors, there exist deterministic bounds on the amount of noise within which the existence and uniqueness of the best approximation are guaranteed [48]. Moreover, upper bounds on the estimation error of a GESD-based computation compared to the actual optimum have been derived as well [50]. In practice, such algebraic estimates serve as good initializations for more costly iterative optimization algorithms, where the aforementioned bounds can be used to gauge the optimality of the resulting approximation. Such bounds have also been extended to the problem of optimal low-rank, positive definite tensor approximation [49]. These results indicate the value of good initial estimates combined with iterative optimization. Even though this strategy may not guarantee global optimality for every conceivable problem instance, it is a highly practical approach that is much less computationally demanding compared to the abovementioned general strategies. As such, it is worthwhile to explore for system identification problems as well.

7.2.3 Computational strategies

In the dissertation, we have attempted to mitigate some of the computational bottlenecks of the presented approach, focusing on two domains: exploiting problem-specific properties and extending more general-purpose computational tools. It should be noted that there are other, often complementary methodologies applicable that can reduce this computational load. However, an advantage of focusing on more general-purpose algorithms is that they can be applied to the many other applications of polynomial and MEP root finding as well, examples of which can be found in Chapter 1.

For example, in the LTI case, the linearity of the auxiliary variables may be exploited by rephrasing the problem as an MEP. We showed that, while this is also possible for the multidimensional misfit problem of Chapter 4, the current computational tools cannot handle the resulting equations due to their infinite solution set. Moreover, these linearly occurring variables may be considered an instance of a sparse monomial support, which can be exploited using toric geometry [15, 16].

Techniques relying on the *real radical* of an ideal allow one to only compute the real-valued common roots. Such techniques have been implemented in the SOS and border bases frameworks [107, 108]. As one is often only interested in the real models in system identification, such techniques may be applied to filter out the spurious complex models. This is most useful in the LTI case [36, 104], as the mD case relies on a parameterization in terms of complex eigentuples. Such complex tuples can be parameterized using a real and a complex part, but

this approach fixes the number of complex-conjugate pairs beforehand. As such, the same problem should then be solved for all possible configurations of the number of conjugate eigentuples.

Lastly, note that permuting the eigentuples and the corresponding initial states does not affect the value of the objective function in the considered problems. This property naturally extends to the common roots of the FONC, as well as the polynomial characterizations following from them. Such symmetries in polynomials can be exploited through symmetry reduction, which has been explored in the SOS framework, as well as in homotopy continuation [62, 193].

Finally, while homotopy continuation has proven more useful for the mD misfit problems in this dissertation, algebraic root-finding techniques have their own strengths when broadening the scope beyond these specific problems. For one, these techniques are more closely related and more applicable to the system-theoretic interpretation of polynomials. Most importantly, NLA-based algebraic techniques can be readily applied to sets of equations where there are more equations than variables, which homotopy continuation cannot tackle off the shelf [24, 43, 176].

7.2.4 Low-rank generalized Hankel matrix approximation

A perspective that was not explored in the dissertation is that of low-rank generalized Hankel matrix approximation. These structured matrices, introduced in Chapter 2, are of rank n when built from a model-compliant sequence belonging to an nth order autonomous, scalar, linear, shift-invariant model. Moreover, given an adequate set of grid points, it has been shown that any low-rank Hankel matrix can be associated with a model-compliant sequence [6, 72]. Because of this connection, the misfit problem can be viewed as optimally approximating a generalized Hankel matrix defined by the given data sequence, by a low-rank Hankel matrix w.r.t. some weighted metric. Note that this approximation problem can also be extended to different metrics by adjusting the weights. This connection has been explored in the LTI case [101].

7.3 Future research

For both parts of the dissertation, we explore potential next steps in this line of research. We focus on the overarching topics of the dissertation, as well as some paths complementary to the conclusions already drawn in the individual chapters.

7.3.1 Globally optimal parameter estimation

To further build upon the presented methodology, a course of action could be to expand it to a broader class of models. For starters, the approach may be formally extended to the case of descriptor systems, during which the connection with the roots at infinity of the affine case may be investigated. Other interesting classes include finite-order, linear, shift-invariant models with inputs or multiple outputs. Such an extension most likely faces similar challenges concerning the model representation and its parameterization. To face these challenges, the scope should be expanded to polynomial module theory, which underpins much of the multidimensional behavioral systems theory literature [31, 139]. For the 1D case, initial steps have already been taken in this regard [85]. However, further efforts to characterize the minima more specifically may prove fruitful.

It was also observed that the current eigentuple-based parameterization does not handle multiple eigentuples well, leading to an infinite number of stationary points and the related computational limitations. As such, it may be of interest to revisit the problem for different parameterizations without such singularities. An interesting starting point here would be *Hilbert schemes*, which studies the parameterizations of polynomials for particular configurations of the associated varieties [151]. This may come at the cost of generality in terms of the described eigentuples, however.

In the LTI case, previous research has attempted to gauge the complexity of the parameter estimation problem by providing upper bounds to the total number of local minimizers by using Thom-Porteous formula [101]. A similar endeavor is interesting for the case of multidimensional models as well. However, as these results rely on the MEP-based characterization of the minima, this may be significantly more difficult in the mD case because of the many additional solutions that are introduced through this reformulation.

Finally, one can hope to further distinguish local minima from saddle points and local maximizers by resorting to second-order optimality information, that is, by imposing the Hessian to be a positive semidefinite matrix. For polynomial optimization problems, the eigenvalues of the Hessian are also defined by a polynomial expression in the model parameters. Imposing nonnegativity of the eigenvalues then leads to polynomial inequalities, which may provide additional insights into the properties of the minimizers. Computationally, such inequalities can be handled by the sum of squares framework [20, 109]. Moreover, it may be of interest to combine both frameworks.

7.3.2 Polynomial root finding

As previously indicated, there are still some obvious ways to continue the research w.r.t. the algorithms presented in Chapters 5 and 6: For the latter, a formal convergence proof for the multiplication map algorithm of 6, as well as a numerical stability analysis are of major significance. Insight in these aspects may lead to ways to improve the numerical behavior of the algorithm. Intuitively, the approach consists of many operations involving non-unitary matrices to solve the subproblems, each causing some of the error buildup. In the current setting, this is unavoidable, though a convergence proof may provide more insight into how such steps can be replaced by unitary operations without affecting the convergence. One may also attempt to generalize this approach to solving MEP's.

For the algorithms of Chapter 5, a numerical stability analysis can provide useful insight into their numerical behavior as well. As previously indicated, the presented block-row decomposition may also not be optimal for the application of root finding using block-Macaulay matrices. This decomposition is recursive and fully disjoint, eliminating overlap between the different block rows. While, at first sight, this seems promising when the degree of the Macaulay increases asymptotically, for the application of root finding, there is an upper bound to the required degree that is useful, that is, when a 'gap' is found. As such, experimenting with alternative decompositions that better handle the trade-off between asymptotic and initial scaling may be of interest as well. Moreover, the required degree is often not known beforehand. Since the algorithm roughly doubles the row degree in each iteration, the obtained Macaulay matrix may be of an unnecessarily large degree. Reducing such overshoot must be taken into account for such decompositions. An asymptotic computational complexity analysis may also prove useful for both algorithms to gauge their complexity. However, its use may be debated given the combinatorial nature of the problem, which makes it hard to achieve the asymptotic regime.

Another development that would be interesting to the presented misfit problem is that of MEP solvers that can handle infinite solution sets. Indeed, this would allow one to more efficiently exploit the linearity of the auxiliary variables in the multidimensional misfit problem, which may greatly reduce the computational cost of the approach. As homotopy continuation can currently handle such problems quite well in a numerical setting, this seems to be the more promising candidate to generalize, compared to algebraic methods. The linearity of these variables can be alternatively exploited by leveraging their limited monomial support, as previously mentioned, which may be done algebraically [15, 16].

Finally, looking further into the connection with polynomial optimization and

the algorithm presented in Chapter 6 can be interesting. For polynomial optimization problems, techniques have been outlined to obtain only the global minimum. These techniques either rely on the critical value polynomial [74] or evaluating the objective function in the multiplication matrices [21, 43], allowing the global minimum to be found as the smallest eigenvalue of a multiplication map through an inverse power method. However, the bottleneck in this approach is setting up the multiplication maps themselves, not solving the coupled eigenvalue problem. The algorithm presented in Chapter 6, on the other hand, relies on multiplication map-like objects in its iterations as well, even before an adequate polynomial basis has been acquired. It may be hypothesized that these premature multiplication maps can reveal some approximate information about the common roots before convergence. Here, the NLA nature of the algorithm may make such an analysis more feasible. With appropriate error bounds, such information may be used to deflate the problem and steer convergence towards only the most promising eigentuples in terms of their objective value. While the idea is definitely far-fetched, the potential payoff makes for an intriguing research path.

Appendix A

Jordan canonical form, commuting matrices and invariant subspaces

In this appendix, some useful properties of commuting matrices are outlined. Commuting matrices arise in state-space models associated with linear, shift-invariant mD models and as multiplication matrices in polynomial coordinate rings.

Let $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{n \times n}$ be a pair of commuting matrices. Commutativity implies some interesting interactions between the (generalized) eigenvectors of $\boldsymbol{A}, \boldsymbol{B}$. Before we get to these properties, we first take a closer look at the generalized eigenvectors and the Jordan canonical form (JCF).

A.1 Jordan canonical form and invariant subspaces

For a detailed treatment of the JCF, we refer the reader to [25, 61, 82].

A vector $\boldsymbol{v}$ is a rank k generalized eigenvector of $\boldsymbol{A}$ associated with the eigenvalue λ if:

$$(\boldsymbol{A} - \lambda \boldsymbol{I})^k \boldsymbol{v} = \boldsymbol{0},$$

$$(\boldsymbol{A} - \lambda \boldsymbol{I})^{k-1} \boldsymbol{v} \neq \boldsymbol{0}.$$

Note that generalized eigenvectors associated with different eigenvalues are linearly independent. This will become evident from their construction in the derivations below, and by the linear independence of eigenvectors associated with different eigenvalues. Thus, the different generalized eigenspaces are disjoint vector spaces.

A.1.1 Construction of generalized eigenvectors

These vectors can be found through the computation of the Jordan Canonical form. Moreover, the generalized eigenvectors associated with some eigenvalue λ form a chain (or multiple chains): the rank k generalized eigenvectors are mapped onto the rank $k-1$ eigenvectors[1] by $(\boldsymbol{A}-\lambda\boldsymbol{I})$. Finally, the rank 1 generalized eigenvectors are just eigenvectors associated with λ, such that they lie in the null space of $(\boldsymbol{A}-\lambda\boldsymbol{I})$. The total dimension of the space spanned by the generalized eigenvectors is equal to the algebraic multiplicity of λ. We formalize this with the following derivation.

Let λ be an eigenvalue of $\boldsymbol{A}$ with algebraic multiplicity m_a and geometric multiplicity[2] m_g. Now let $\boldsymbol{V}$ be an orthogonal basis for the m_g-dimensional eigenspace associated with λ and let $\boldsymbol{Q}$ be an orthonormal basis for the orthogonal complement of this space. We then perform the similarity transformation:

$$\begin{bmatrix}\boldsymbol{V}^\mathsf{T}\\ \boldsymbol{Q}^\mathsf{T}\end{bmatrix}(\boldsymbol{A}-\lambda\boldsymbol{I})\begin{bmatrix}\boldsymbol{V} & \boldsymbol{Q}\end{bmatrix}=\begin{bmatrix}\boldsymbol{0} & \boldsymbol{V}^\mathsf{T}(\boldsymbol{A}-\lambda\boldsymbol{I})\boldsymbol{Q}\\ \boldsymbol{0} & (\boldsymbol{Q}^\mathsf{T}\boldsymbol{A}\boldsymbol{Q}-\lambda\boldsymbol{I})\end{bmatrix}.$$

Now, since this matrix is block triangular, its characteristic equation is the product of the characteristic equations of the diagonal blocks: $\det(\boldsymbol{A}-\lambda\boldsymbol{I})=(x-\lambda)^{m_g}\det\big(\boldsymbol{Q}^\mathsf{T}\boldsymbol{A}\boldsymbol{Q}-\lambda\boldsymbol{I}\big)$. It is then clear that $\boldsymbol{Q}^\mathsf{T}\boldsymbol{A}\boldsymbol{Q}$ has the eigenvalue λ with an algebraic multiplicity m_a-m_g.

Let $\boldsymbol{V}_2$ be the eigenspace of $\boldsymbol{Q}^\mathsf{T}\boldsymbol{A}\boldsymbol{Q}$ w.r.t. λ and note that:

$$\begin{aligned}\begin{bmatrix}\boldsymbol{V}^\mathsf{T}(\boldsymbol{A}-\lambda\boldsymbol{I})\boldsymbol{Q}\\ (\boldsymbol{Q}^\mathsf{T}\boldsymbol{A}\boldsymbol{Q}-\lambda\boldsymbol{I})\end{bmatrix}\boldsymbol{V}_2&=\begin{bmatrix}\boldsymbol{V}^\mathsf{T}(\boldsymbol{A}-\lambda\boldsymbol{I})\boldsymbol{Q}\boldsymbol{V}_2\\ \boldsymbol{0}\end{bmatrix}\\ &=\begin{bmatrix}\boldsymbol{V}^\mathsf{T}\\ \boldsymbol{Q}^\mathsf{T}\end{bmatrix}(\boldsymbol{A}-\lambda\boldsymbol{I})\boldsymbol{Q}\boldsymbol{V}_2.\end{aligned}$$

Because $\begin{bmatrix}\boldsymbol{V} & \boldsymbol{Q}\end{bmatrix}$ is an orthogonal basis for $\mathbb{R}^n$, it is clear that $(\boldsymbol{A}-\lambda\boldsymbol{I})\boldsymbol{Q}\boldsymbol{V}_2$ lies fully in the subspace spanned by $\boldsymbol{V}$, i.e. the eigenspace of $\boldsymbol{A}$ w.r.t. λ. So

[1] To complete the chain, note that $\boldsymbol{0}$ is always a rank 0 eigenvector.

[2] Each matrix has at least one eigenvalue and associated eigenvector, since $\det(\boldsymbol{A}-\lambda\boldsymbol{I})$ has a solution.

under the map $(\boldsymbol{A} - \lambda \boldsymbol{I})$, the vectors in range $\boldsymbol{Q}\boldsymbol{V}_2$ get mapped onto range $\boldsymbol{V}$, which get mapped onto $\boldsymbol{0}$. As such, $\boldsymbol{Q}\boldsymbol{V}_2$ clearly span the rank 2 generalized eigenvectors. If the full algebraic multiplicity is not accounted for yet, this procedure can be continued for $\boldsymbol{Q}^\mathsf{T}\boldsymbol{A}\boldsymbol{Q}$ to find the higher rank eigenvectors.

A.1.2 Jordan chains and elementary invariant subspaces

Next, let $\boldsymbol{p}, \boldsymbol{q}$ be two linearly independent rank $k > 1$ generalized eigenvectors and let them be orthogonal to all rank $< k$ generalized eigenvectors, as per the construction above. Notice that these vectors cannot be mapped to the same rank $k-1$ generalized eigenvector $\boldsymbol{v}$ under $(\boldsymbol{A} - \lambda \boldsymbol{I})$. Indeed, it would then hold that $(\boldsymbol{A} - \lambda \boldsymbol{I})\boldsymbol{p} = (\boldsymbol{A} - \lambda \boldsymbol{I})\boldsymbol{q} = \boldsymbol{v}$, which implies that $\boldsymbol{q} - \boldsymbol{p}$ lies in the null space of $(\boldsymbol{A} - \lambda \boldsymbol{I})$. However, this is impossible since they are orthogonal to all eigenvectors associated with λ. By taking linear combinations, this argument can be easily extended to show that any number of rank k vectors is to be mapped to the same number of rank $k-1$ vectors, implying that each linearly independent rank k eigenvector defines its own chain of lower-rank eigenvectors, disjoint from the space spanned by other rank k chains.

These generalized eigenvectors thus define chains of vectors that are mapped onto one another by $(\boldsymbol{A} - \lambda \boldsymbol{I})$ from a higher rank to a lower rank. The space spanned by the vectors in such a chain then forms an *invariant subspace* of the matrix $\boldsymbol{A}$, that is, a subspace range $\boldsymbol{V}$ such that range $\boldsymbol{A}\boldsymbol{V} \subseteq$ range $\boldsymbol{V}$. A well-known example of such an invariant subspace is the space spanned by an eigenvector $\boldsymbol{v}$ of $\boldsymbol{A}$: $\boldsymbol{A}\boldsymbol{v} = \lambda \boldsymbol{v} \in$ range $\boldsymbol{v}$. Indeed, by the observations above, one can choose a basis $\boldsymbol{V}_\lambda \in \mathbb{R}^{n \times m_a}$ with as its columns the generalized eigenvectors associated with λ, such that each rank $k > 1$ column of $\boldsymbol{V}_\lambda$ is mapped onto exactly one rank $k-1$ column by $(\boldsymbol{A} - \lambda \boldsymbol{I})$. In this basis, the map defined by $\boldsymbol{A}$ is described by the Jordan block $\boldsymbol{J}_\lambda$ associated with λ, defined as: $\boldsymbol{J}_\lambda = {\boldsymbol{V}_\lambda}^\dagger \boldsymbol{A} \boldsymbol{V}_\lambda$. For example, let the ith column be mapped onto the $(i-1)$th column:

$$\begin{aligned} \boldsymbol{V}_\lambda \boldsymbol{e}_{i-1} &= (\boldsymbol{A} - \lambda \boldsymbol{I}) \boldsymbol{V}_\lambda \boldsymbol{e}_i \\ \boldsymbol{e}_{i-1} &= {\boldsymbol{V}_\lambda}^\dagger \boldsymbol{V}_\lambda (\boldsymbol{J}_\lambda - \lambda \boldsymbol{I}) {\boldsymbol{V}_\lambda}^\dagger \boldsymbol{V}_\lambda \boldsymbol{e}_i \\ \boldsymbol{e}_{i-1} &= (\boldsymbol{J}_\lambda - \lambda \boldsymbol{I}) \boldsymbol{e}_i. \end{aligned}$$

Similar bases can be constructed for all q eigenvalues of $\boldsymbol{A}$, leading to the full JCF, using the basis $\boldsymbol{V} = \begin{bmatrix} \boldsymbol{V}_{\lambda_1} & \cdots & \boldsymbol{V}_{\lambda_q} \end{bmatrix}$.

Finally, note that no other generalized eigenvector maps onto $\boldsymbol{V}_\lambda \boldsymbol{e}_{i-1}$ under this map, i.e., the $(i-1)$th row of $(\boldsymbol{J}_\lambda - \lambda \boldsymbol{I})$ is only nonzero in the ith column. This implies that ${\boldsymbol{e}_{i-1}}^\mathsf{T}(\boldsymbol{J}_\lambda - \lambda \boldsymbol{I}) = {\boldsymbol{e}_i}^\mathsf{T}$, such that $\boldsymbol{A}$ maps the $(i-1)$th row of

$\boldsymbol{V}_\lambda{}^\dagger$ to its ith row. As such, any chain of right generalized eigenvectors,

$$\boldsymbol{V}_\lambda \boldsymbol{e}_i \xrightarrow{(\boldsymbol{A}-\lambda \boldsymbol{I})} \boldsymbol{V}_\lambda \boldsymbol{e}_{i-1} \xrightarrow{(\boldsymbol{A}-\lambda \boldsymbol{I})} \dots \xrightarrow{(\boldsymbol{A}-\lambda \boldsymbol{I})} \boldsymbol{V}_\lambda \boldsymbol{e}_{i-l}$$

implies the existence of the following "reversed" chain of left generalized eigenvectors:

$$\boldsymbol{e}_{i-l}{}^\mathsf{T} \boldsymbol{V}_\lambda{}^\dagger \xrightarrow{(\boldsymbol{A}-\lambda \boldsymbol{I})} \boldsymbol{e}_{i-l+1}{}^\mathsf{T} \boldsymbol{V}_\lambda{}^\dagger \xrightarrow{(\boldsymbol{A}-\lambda \boldsymbol{I})} \dots \xrightarrow{(\boldsymbol{A}-\lambda \boldsymbol{I})} \boldsymbol{e}_i{}^\mathsf{T} \boldsymbol{V}_\lambda{}^\dagger. \tag{A.1}$$

Example A.1.1. To illustrate these rather abstract observations, let us take a closer look at an example JCF of some matrix with an eigenvalue λ, with an algebraic multiplicity of six, involving two Jordan chains: one of length four and one of length two. As was just derived, there exists a basis $\boldsymbol{V}_\lambda$ that leads to the following form:

$$\boldsymbol{J}_\lambda = \boldsymbol{V}_\lambda{}^\dagger \boldsymbol{A} \boldsymbol{V}_\lambda = \left[\begin{array}{cccccc} \lambda & 1 & & & & \\ & \lambda & 1 & & & \\ & & \lambda & 1 & & \\ & & & \lambda & 0 & \\ \hline & & & & \lambda & 1 \\ & & & & & \lambda \end{array}\right].$$

Notice how, for example, $\boldsymbol{J}_\lambda$ maps $\boldsymbol{e}_4$ as: $\boldsymbol{J}_\lambda \boldsymbol{e}_4 = \lambda \boldsymbol{e}_4 + \boldsymbol{e}_3$, or, in the original coordinates, that $\boldsymbol{A}\boldsymbol{V}_\lambda \boldsymbol{e}_4 = \boldsymbol{V}_\lambda(\boldsymbol{e}_4 + \boldsymbol{e}_3)$, since range $\boldsymbol{A}\boldsymbol{V}_\lambda \subseteq$ range $\boldsymbol{V}_\lambda$. However, $\boldsymbol{J}_\lambda \boldsymbol{e}_5 = \lambda \boldsymbol{e}_5$, is not mapped to $\boldsymbol{e}_4$, breaking the Jordan form into two blocks, each corresponding to one Jordan chain. As such, $\boldsymbol{V}_\lambda \boldsymbol{e}_1, \boldsymbol{V}_\lambda \boldsymbol{e}_5$ are the eigenvectors of $\boldsymbol{A}$ associated with λ. Each block then corresponds to an invariant subspace of $\boldsymbol{A}$ range $\boldsymbol{A}\boldsymbol{V}_\lambda \begin{bmatrix} \boldsymbol{e}_1 & \boldsymbol{e}_2 & \boldsymbol{e}_3 & \boldsymbol{e}_4 \end{bmatrix} \subseteq$ range $\boldsymbol{V}_\lambda \begin{bmatrix} \boldsymbol{e}_1 & \boldsymbol{e}_2 & \boldsymbol{e}_3 & \boldsymbol{e}_4 \end{bmatrix}$ and, similarly, range $\boldsymbol{A}\boldsymbol{V}_\lambda \begin{bmatrix} \boldsymbol{e}_5 & \boldsymbol{e}_6 \end{bmatrix} \subseteq$ range $\boldsymbol{V}_\lambda \begin{bmatrix} \boldsymbol{e}_5 & \boldsymbol{e}_6 \end{bmatrix}$. Similarly, $\boldsymbol{e}_4{}^\mathsf{T} \boldsymbol{V}_\lambda{}^\dagger$ and $\boldsymbol{e}_6{}^\mathsf{T} \boldsymbol{V}_\lambda{}^\dagger$ are left eigenvectors of $\boldsymbol{A}$ and the Jordan chains now correspond to ascending rows of $\boldsymbol{V}_\lambda{}^\dagger$, e.g., $\boldsymbol{e}_3{}^\mathsf{T} \boldsymbol{V}_\lambda{}^\dagger \boldsymbol{A} = (\lambda \boldsymbol{e}_3{}^\mathsf{T} + \boldsymbol{e}_4{}^\mathsf{T}) \boldsymbol{V}_\lambda{}^\dagger$. ◇

A.2 Joint Schur triangularization

For more in-depth depositions on joint triangularization, we refer the reader to [82, 143].

Remark that the JCF describes the elementary invariant subspaces of a matrix, i.e., a single Jordan chain. We now consider the case with two commuting matrices $\boldsymbol{A}, \boldsymbol{B}$ and to show how these invariant subspaces interact. Notice that for any eigenvector $\boldsymbol{v}$ of $\boldsymbol{A}$, it holds that:

$$\boldsymbol{A}\boldsymbol{B}\boldsymbol{v} = \boldsymbol{B}\boldsymbol{A}\boldsymbol{v} = \lambda \boldsymbol{B}\boldsymbol{v},$$

implying that $\boldsymbol{Bv}$ is again an eigenvector of $\boldsymbol{A}$ associated with λ. As such, the eigenvectors of $\boldsymbol{A}$ associated with λ form a $\boldsymbol{B}-$invariant subspace. Formally, let $\boldsymbol{V}_\lambda$ be an orthogonal basis for this eigenspace of $\boldsymbol{A}$, it holds that $\operatorname{range} \boldsymbol{BV}_\lambda \subseteq \operatorname{range} \boldsymbol{V}_\lambda$.

Now, we show that any $\boldsymbol{B}$-invariant space $\operatorname{range} \boldsymbol{V}$ contains an eigenvector of $\boldsymbol{B}$. Let $\boldsymbol{V}$ be an orthonormal basis and take an eigenvector $\boldsymbol{x}$: $\boldsymbol{V}^\mathsf{T}\boldsymbol{BVx} = \mu\boldsymbol{x}$. Now note that $\boldsymbol{VV}^\mathsf{T}\boldsymbol{p}$ is an identity operation for any vector $\boldsymbol{p} \in \operatorname{range} \boldsymbol{V}$, since $\boldsymbol{V}^\mathsf{T}$ acts fully decomposes $\boldsymbol{p}$ into its coordinates w.r.t. the basis $\boldsymbol{v}$. By $\operatorname{range} \boldsymbol{BV} \subseteq \operatorname{range} \boldsymbol{V}$, it thus holds that $\boldsymbol{B}(\boldsymbol{Vx}) = \boldsymbol{VV}^\mathsf{T}\boldsymbol{B}(\boldsymbol{Vx}) = \mu\boldsymbol{Vx}$, i.e., $\boldsymbol{Vx}$ is an eigenvector of $\boldsymbol{B}$ and clearly lies in $\operatorname{range} \boldsymbol{V}$.

We now show how this leads to a joint Schur transformation. Since the $\boldsymbol{B}$-invariant space $\boldsymbol{V}_\lambda$ is the eigenspace of $\boldsymbol{A}$, there is some $\boldsymbol{B}-$eigenvector $\boldsymbol{v}$ in this eigenspace. As such there exists a common eigenvector $\boldsymbol{v}$ of $\boldsymbol{A}, \boldsymbol{B}$. Then let $\boldsymbol{Q}$ be an orthonormal basis for the orthogonal complement of $\boldsymbol{v}$ and take the following common, orthogonal similarity transformation:

$$\boldsymbol{A}_\text{sim} = \begin{bmatrix} \boldsymbol{v}^\mathsf{T} \\ \boldsymbol{Q}^\mathsf{T} \end{bmatrix} \boldsymbol{A} \begin{bmatrix} \boldsymbol{v} & \boldsymbol{Q} \end{bmatrix} = \begin{bmatrix} \lambda & \boldsymbol{v}^\mathsf{T}\boldsymbol{AQ} \\ \boldsymbol{0} & \boldsymbol{Q}^\mathsf{T}\boldsymbol{AQ} \end{bmatrix},$$

$$\boldsymbol{B}_\text{sim} = \begin{bmatrix} \boldsymbol{v}^\mathsf{T} \\ \boldsymbol{Q}^\mathsf{T} \end{bmatrix} \boldsymbol{B} \begin{bmatrix} \boldsymbol{v} & \boldsymbol{Q} \end{bmatrix} = \begin{bmatrix} \mu & \boldsymbol{v}^\mathsf{T}\boldsymbol{BQ} \\ \boldsymbol{0} & \boldsymbol{Q}^\mathsf{T}\boldsymbol{BQ} \end{bmatrix}.$$

Note that $\boldsymbol{A}_\text{sim}$ and $\boldsymbol{B}_\text{sim}$ also commute, such that from the bottom-right corner of this equality, it follows that $\boldsymbol{Q}^\mathsf{T}\boldsymbol{BQQ}^\mathsf{T}\boldsymbol{AQ} = \boldsymbol{Q}^\mathsf{T}\boldsymbol{AQQ}^\mathsf{T}\boldsymbol{BQ}$. As such, there again exists a common eigenvector for these matrices, such that the procedure can be repeated until a fully upper triangular form is achieved for both matrices.

This argument can be readily extended to any number of pairwise commuting matrices. For example, consider three commuting matrices $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$. As was previously shown, there exist eigenvalues λ and μ of $\boldsymbol{A}$ and $\boldsymbol{B}$, respectively. Consequently, $\operatorname{range} \boldsymbol{V}_\lambda \cap \operatorname{range} \boldsymbol{V}_\mu = \operatorname{range} \boldsymbol{V}_{\lambda,\mu} \neq \{\boldsymbol{0}\}$. By the $\boldsymbol{C}$-invariance of $\boldsymbol{V}_\lambda$ and $\boldsymbol{V}_\mu$, it holds that $\operatorname{range} \boldsymbol{CV}_{\lambda,\mu} \subseteq \operatorname{range} \boldsymbol{V}_\lambda \cap \operatorname{range} \boldsymbol{V}_\mu = \operatorname{range} \boldsymbol{V}_{\lambda,\mu}$, such that this space is also $\boldsymbol{C}$-invariant and thus contains an eigenvector of $\boldsymbol{C}$.

The existence of a joint eigenvector suffices to perform a step of the joint Schur decomposition. By bringing all commuting matrices in a jointly triangular form, which reveals the eigenvalues on the diagonal, the corresponding diagonal entries can be paired into tuples: the eigentuples.

A.3 Jordan form and commutativity

For a more elaborate treatment of this subject, we refer the reader to [61, 143] and more precisely Theorem 2.4.6.3 in [82].

A.3.1 Decomposition into joint invariant subspaces associated with eigentuples

Instead of looking at just the eigenvectors, let us take a closer look at the generalized eigenspaces of commuting matrices $\boldsymbol{A}$ and $\boldsymbol{B}$. Again, notice that:

$$(\boldsymbol{A} - \lambda \boldsymbol{I})^k \boldsymbol{B}\boldsymbol{v} = \boldsymbol{B}(\boldsymbol{A} - \lambda \boldsymbol{I})^k \boldsymbol{v} = \boldsymbol{0}.$$

While it may or may not hold that $\boldsymbol{B}(\boldsymbol{A} - \lambda \boldsymbol{I})^{k-1}\boldsymbol{v} \neq \boldsymbol{0}$, it is clear that $\boldsymbol{B}\boldsymbol{v}$ is a rank $\leq k$ generalized eigenvector of $\boldsymbol{A}$ associated with λ. So, letting $\boldsymbol{V}_\lambda$ be the basis associated with the λ-Jordan block of $\boldsymbol{A}$, then range $\boldsymbol{V}_\lambda$ is $\boldsymbol{B}$-invariant. Moreover, let the columns of $\boldsymbol{V}_\lambda^{(k)}$ be the subset of rank $\leq k$ generalized eigenvectors in $\boldsymbol{V}_\lambda$, then range $\boldsymbol{V}_\lambda^{(k)}$ is $\boldsymbol{B}$-invariant for any k.

Let us look at how the generalized eigenspaces of $\boldsymbol{A}$ interact with those of $\boldsymbol{B}$. Let $\boldsymbol{V}_\mu$ be a basis for a generalized eigenspace of $\boldsymbol{B}$ associated with an eigenvalue μ, i.e., the space spanned by all generalized eigenvectors associated with μ. As was previously established, the elementary invariant subspaces of a matrix are spanned by a single chain of generalized eigenvectors, i.e., the generalized eigenvectors associated with one subblock of the JCF. As such, the $\boldsymbol{B}$-invariant subspace range $\boldsymbol{V}_\lambda$ can be written as a direct sum of such elementary invariant subspaces.

Consequently, there must exist some values of λ and μ such that their intersection range $\boldsymbol{V}_\lambda \cap$ range $\boldsymbol{V}_\mu =$ range $\boldsymbol{V}_{\mu,\lambda}$ is nontrivial, i.e., range $\boldsymbol{V}_{\mu,\lambda} \neq \{\boldsymbol{0}\}$. Moreover, notice that $\boldsymbol{V}_\mu$ is $\boldsymbol{A}$-invariant by the same argument used to show that $\boldsymbol{V}_\lambda$ is $\boldsymbol{B}$-invariant, such that range $\boldsymbol{A}\boldsymbol{V}_{\mu,\lambda} \subseteq$ range $\boldsymbol{V}_\mu$. By the $\boldsymbol{A}$-invariance of range $\boldsymbol{V}_\lambda$, it also holds that range $\boldsymbol{A}\boldsymbol{V}_{\mu,\lambda} \subseteq$ range $\boldsymbol{V}_\lambda$, implying that $\boldsymbol{A}\boldsymbol{V}_{\mu,\lambda} \subseteq$ range $\boldsymbol{V}_\lambda \cap$ range $\boldsymbol{V}_\mu =$ range $\boldsymbol{V}_{\lambda,\mu}$. Range $\boldsymbol{V}_{\lambda,\mu}$ thus defines an $\boldsymbol{A}$-invariant subspace, and by a similar argument, a $\boldsymbol{B}$-invariant subspace.

Because generalized eigenspaces associated with different eigenvalues are disjoint spaces, it holds that range $\boldsymbol{V}_{\lambda_1,\mu} \cap$ range $\boldsymbol{V}_{\lambda_2,\mu} = \{\boldsymbol{0}\}$ whenever $\lambda_1 \neq \lambda_2$, and similarly for distinct values of μ. In such cases, the $\boldsymbol{B}$-invariant subspace range $\boldsymbol{V}_\mu$ thus contains at least two disjoint invariant subspaces: range $\boldsymbol{V}_{\lambda_1,\mu}$ and range $\boldsymbol{V}_{\lambda_2,\mu}$, so that there are at least two Jordan blocks associated with μ. This shows that there is at least one joint invariant subspace associated with

each distinct eigentuple (λ, μ), the dimension of which is no greater than the individual generalized eigenspaces associated with λ and μ.

Note that this argument can be readily extended to a larger number of commuting matrices to show that there always exists an invariant subspace associated with each distinct eigentuple.

A.3.2 Interaction with different Jordan chains

Remark that range $\boldsymbol{V}_\lambda^{(k)}$ being $\boldsymbol{B}$-invariant has strong implications about how generalized eigenvectors of $\boldsymbol{A}$ can be mapped under $\boldsymbol{B}$. Indeed, this implies that $\boldsymbol{B}$ can map generalized eigenvectors associated with a λ-Jordan block of $\boldsymbol{A}$ to linear combinations involving other eigenvectors associated with different λ-Jordan chains, as long as the resulting generalized eigenvector is at most of the same rank[3]. For a rank k generalized eigenvector, this can then be written out as $\boldsymbol{B}\boldsymbol{V}_\lambda^{(k)}\boldsymbol{v} = \boldsymbol{V}_\lambda^{(k)}\boldsymbol{x}$. A similar analysis can be carried out for the left generalized eigenvectors to show that rank k left eigenvectors of $\boldsymbol{A}$ can only be mapped to linear combinations of rank $\leq k$ left eigenvectors associated with the same eigenvalue by $\boldsymbol{B}$.

Let $\mathcal{N}$ denote the ordered set of column indices corresponding to the nonzero columns of $(\boldsymbol{A} - \lambda\boldsymbol{I})\boldsymbol{V}_\lambda^{(k)}$, which form the basis $\boldsymbol{V}_\lambda^{(k-1)}$. Then the next rank $k-1$ generalized eigenvector in the $(\boldsymbol{A} - \lambda\boldsymbol{I})\boldsymbol{V}_\lambda^{(k)}$-chain defined by $\boldsymbol{V}_\lambda^{(k)}\boldsymbol{v}$ is then mapped as follows:

$$\boldsymbol{B}\boldsymbol{V}_\lambda^{(k-1)}\boldsymbol{v}_\mathcal{N} = \boldsymbol{B}(\boldsymbol{A} - \lambda\boldsymbol{I})\boldsymbol{V}_\lambda^{(k)}\boldsymbol{v} = (\boldsymbol{A} - \lambda\boldsymbol{I})\boldsymbol{V}_\lambda^{(k)}\boldsymbol{x} = \boldsymbol{V}_\lambda^{(k-1)}\boldsymbol{x}_\mathcal{N}. \quad \text{(A.2)}$$

In a sense, $\boldsymbol{B}$ thus maps $\boldsymbol{A}$-Jordan chains to linear combinations of other $\boldsymbol{A}$-Jordan chains: if some rank k eigenvector $\boldsymbol{V}_\lambda^{(k)}\boldsymbol{v}$ is mapped to a linear combination of other generalized eigenvectors, any subsequent eigenvector in its chain is mapped to the same linear combination of the subsequent chain elements associated with these other generalized eigenvectors.

Now consider the map described by $\boldsymbol{B}$ in the JCF basis, i.e., $\boldsymbol{V}_\lambda{}^\dagger\boldsymbol{B}\boldsymbol{V}_\lambda$, and assume the entry at position (j, i) is nonzero, that is, $\boldsymbol{e}_j{}^\mathsf{T}\boldsymbol{V}_\lambda{}^\dagger\boldsymbol{B}\boldsymbol{V}_\lambda\boldsymbol{e}_i \neq 0$. Note that this not only implies that the generalized eigenvector $\boldsymbol{V}_\lambda\boldsymbol{e}_i$ is mapped to a linear combination involving $\boldsymbol{V}_\lambda\boldsymbol{e}_j$, but it also implies that the left eigenvector $\boldsymbol{e}_j{}^\mathsf{T}\boldsymbol{V}_\lambda{}^\dagger$ is mapped onto a linear combination involving $\boldsymbol{e}_i{}^\mathsf{T}\boldsymbol{V}_\lambda{}^\dagger$. However, the rank of the right generalized eigenvector $\boldsymbol{V}_\lambda\boldsymbol{e}_i$ is not equal to that of the left

[3]These interactions can be better captured using the lesser-known Weyr canonical form, rather than a Jordan form. Indeed, all members of a commuting family of matrices can be put in a simultaneous upper triangular form, while one of the matrices is in Weyr from [82, 143].

eigenvector $\boldsymbol{e}_i^{\mathsf{T}}\boldsymbol{V}_\lambda{}^\dagger$. Rather, by (A.1), if the former is of rank k_1 and the total length of the chain is l_1, then the latter is of rank $l_1 - k_1 + 1$. Consequently, a vector in some $\boldsymbol{A}$-Jordan chain can only be mapped to linear combinations involving a rank k_2 vector in a different Jordan chain of length l_2, if $k_2 \leq k_1$ and $l_1 - k_1 \leq l_2 - k_2 \to l_1 - l_2 + k_2 \leq k_1$. This ensures that only the highest rank eigenvectors in a chain can map onto vectors in shorter λ-Jordan chains under $\boldsymbol{B}$, imposing additional zero entries in $\boldsymbol{V}_\lambda{}^\dagger\boldsymbol{B}\boldsymbol{V}_\lambda$.

Example A.3.1. Let us illustrate all the observations above by looking at a matrix $\boldsymbol{B}$ that commutes with $\boldsymbol{A}$ from Example A.1.1. Consider the map $\boldsymbol{B}$ defines in the JCF basis $\boldsymbol{V}_\lambda$ used in said example:

$$\boldsymbol{V}_\lambda{}^\dagger\boldsymbol{B}\boldsymbol{V}_\lambda = \left[\begin{array}{cccc|cc} x_4 & x_3 & x_2 & x_1 & y_2 & y_1 \\ & x_4 & x_3 & x_2 & & y_2 \\ & & x_4 & x_3 & & \\ & & & x_4 & & \\ \hline & & x_6 & x_5 & y_6 & y_5 \\ & & & x_6 & & y_6 \end{array}\right] = \left[\begin{array}{c|c} \boldsymbol{C} & \boldsymbol{D} \\ \hline \boldsymbol{E} & \boldsymbol{F} \end{array}\right].$$

Notice that the rank 4 eigenvector $\boldsymbol{V}_\lambda\boldsymbol{e}_4$ is mapped onto $\boldsymbol{V}_\lambda\boldsymbol{x}$ under $\boldsymbol{B}$. Since this is the highest rank eigenvector, it can be mapped to any other eigenvector. The Toeplitz structure of $\boldsymbol{C}$ and $\boldsymbol{D}$ is explained by (A.2). Indeed, notice that the next element in this Jordan chain is $\boldsymbol{V}_\lambda\boldsymbol{e}_3 = (\boldsymbol{A} - \lambda\boldsymbol{I})\boldsymbol{V}_\lambda\boldsymbol{e}_4$ and that $(\boldsymbol{A} - \lambda\boldsymbol{I})\boldsymbol{V}_\lambda = \boldsymbol{V}_\lambda\begin{bmatrix}\boldsymbol{0} & \boldsymbol{e}_1 & \boldsymbol{e}_2 & \boldsymbol{e}_3 & \boldsymbol{0} & \boldsymbol{e}_5\end{bmatrix}$, such that $\boldsymbol{B}\boldsymbol{V}_\lambda\boldsymbol{e}_3 = \boldsymbol{V}_\lambda\begin{bmatrix}x_2 & x_3 & x_4 & \boldsymbol{0} & x_6 & \boldsymbol{0}\end{bmatrix}^{\mathsf{T}}$. The same reasoning can be applied to subsequent elements in this Jordan chain to explain the full Toeplitz structure.

The upper triangular structure in both $\boldsymbol{C}$ and $\boldsymbol{E}$ can be explained by the "rank conditions": $k_2 \leq k_1$ and $2 + k_2 \leq k_1$. Taking a closer look at $\boldsymbol{E}$, note that only the eigenvectors $\boldsymbol{V}_\lambda\boldsymbol{e}_3, \boldsymbol{V}_\lambda\boldsymbol{e}_4$, of rank $k_1 = 3$ and $k_1 = 4$, respectively, can map onto $\boldsymbol{V}_\lambda\boldsymbol{e}_5, \boldsymbol{V}_\lambda\boldsymbol{e}_6$, which are of ranks $k_1 = 1$ and $k_1 = 2$. This constrains the first two columns of $\boldsymbol{E}$ to be zero and imposes the upper triangular structure.

For the second Jordan chain, one can also consider the image of the largest rank vector: $\boldsymbol{B}\boldsymbol{V}_\lambda\boldsymbol{e}_6\boldsymbol{V}_\lambda\boldsymbol{y}$. In this case, the first "rank condition" explains the upper triangular structure and the fact that the last two rows of $\boldsymbol{D}$ are zero, while the Toeplitz structure is explained by (A.2).

Lastly, note that, since each eigentuple (λ, μ) defines its own invariant subspace, the off-diagonal blocks $\boldsymbol{D}$ and $\boldsymbol{E}$ can only be nonzero if all eigenvalues of $\boldsymbol{V}_\lambda{}^\dagger\boldsymbol{B}\boldsymbol{V}_\lambda$ are equal. Indeed, in case there are two different eigenvalues, it was established that there exist two $\boldsymbol{A}$- and $\boldsymbol{B}$-invariant subspaces $\boldsymbol{V}_{\lambda,\mu_1}$ and $\boldsymbol{V}_{\lambda,\mu_2}$. Clearly, the only two $\boldsymbol{A}$-invariant subspaces associated with λ are spanned by the bases corresponding to each Jordan block: $\operatorname{range}\boldsymbol{V}_\lambda\begin{bmatrix}\boldsymbol{e}_1 & \boldsymbol{e}_2 & \boldsymbol{e}_3 & \boldsymbol{e}_4\end{bmatrix}$ and $\operatorname{range}\boldsymbol{V}_\lambda\begin{bmatrix}\boldsymbol{e}_5 & \boldsymbol{e}_6\end{bmatrix}$. If these two subspaces coincide with generalized eigenspaces

of $\boldsymbol{B}$ associated with two distinct eigenvalues μ_1 and μ_2, then both these spaces are $\boldsymbol{B}$-invariant. It would then hold that $\boldsymbol{B}\boldsymbol{V}_\lambda \boldsymbol{e}_6 = \boldsymbol{V}_\lambda \begin{bmatrix} \boldsymbol{e}_5 & \boldsymbol{e}_6 \end{bmatrix} \boldsymbol{y}$, such that $y_1 = y_2 = 0$ and $\boldsymbol{B}\boldsymbol{V}_\lambda \boldsymbol{e}_4 = \boldsymbol{V}_\lambda \begin{bmatrix} \boldsymbol{e}_1 & \boldsymbol{e}_2 & \boldsymbol{e}_3 & \boldsymbol{e}_4 \end{bmatrix} \boldsymbol{x}$, such that $x_5 = x_6 = 0$. $\diamond$

Bibliography

[1] Abbott, J., Bigatti, A., Kreuzer, M., and Robbiano, L. Computing ideals of points. *J. Symb. Comput. 30* (2000), 341–356.

[2] Adali, T., and Schreier, P. J. Optimization and estimation of complex-valued signals: Theory and applications in filtering and blind source separation. *IEEE Signal Processing Magazine 31*, 5 (2014), 112–128.

[3] Ali, M., Chughtai, S. S., and Werner, H. Consistent identification of two-dimensional systems. In *Proceedings of the 2010 American Control Conference* (2010), pp. 3464–3469.

[4] Alsubaie, F. F. *$\mathcal{H}_2$ Optimal Model Reduction for Linear Dynamic Systems and the Solution of Multiparameter Matrix Pencil Problems*. PhD thesis, Imperial College London, 2019.

[5] Atkinson, F. V., and Mingarelli, A. B. *Multiparameter Eigenvalue Problems: Sturm-Liouville Theory*. CRC Press, 2010.

[6] Attasi, S. Modelling and recursive estimation for double indexed sequences. In *System Identification Advances and Case Studies*, vol. 126. Elsevier, 1976, pp. 289–348.

[7] Auzinger, W., and Stetter, H. J. An elimination algorithm for the computation of all zeros of a system of multivariate polynomial equations. In *Numerical Mathematics Singapore 1988: Proceedings of the International Conference on Numerical Mathematics*. Birkhäuser Basel, Basel, 1988, pp. 11–30.

[8] Ball, J. A., and Vinnikov, V. Overdetermined multidimensional systems: State space and frequency domain methods. *Mathematical Systems Theory in Biology, Communications, Computation, and Finance* (2003), 63–119.

[9] Batselier, K., Dreesen, P., and De Moor, B. Prediction error method identification is an eigenvalue problem. *IFAC Proceedings Volumes 45*, 16 (2012), 221–226.

[10] Batselier, K., Dreesen, P., and De Moor, B. The geometry of multivariate polynomial division and elimination. *SIAM Journal on Matrix Analysis and Applications 34*, 1 (2013), 102–125.

[11] Batselier, K., Dreesen, P., and De Moor, B. A fast recursive orthogonalization scheme for the Macaulay matrix. *Journal of Computational and Applied Mathematics 267* (2014), 20–32. Publisher: North-Holland.

[12] Batselier, K., Dreesen, P., and De Moor, B. On the null spaces of the Macaulay matrix. *Linear Algebra and its Applications 460* (2014), 259–289.

[13] Batselier, K., and Wong, N. Computing the state difference equations for discrete overdetermined linear mD systems. *Automatica 64* (2016), 254–261.

[14] Batselier, K., and Wong, N. Inverse multivariate polynomial root-finding: Numerical implementations of the affine and projective Buchberger–Möller algorithm. *Journal of Computational and Applied Mathematics 320* (2017), 15–29.

[15] Bender, M. R., and Telen, S. Toric eigenvalue methods for solving sparse polynomial systems. *Mathematics of Computations 91* (2020), 2397–2429.

[16] Bender, M. R., and Telen, S. Yet another eigenvalue algorithm for solving polynomial systems. In *Varieties, polyhedra, computation*. EMS Press, Berlin, 2025, pp. 355–394.

[17] Beylkin, G., and Mohlenkamp, M. J. Algorithms for numerical analysis in high dimensions. *SIAM Journal on Scientific Computing 26*, 6 (2005), 2133–2159.

[18] Billings, S. A., and Coca, D. Finite element Galerkin models and identified finite element models - a comparative study. *IFAC Proceedings Volumes 35*, 1 (2002), 307–312. 15th IFAC World Congress.

[19] Björk, J.-E. *Rings of differential operators*, vol. 21. Elsevier, 1979.

[20] Blekherman, G., Parrilo, P. A., and Thomas, R. R. *Semidefinite Optimization and Convex Algebraic Geometry*, vol. 13. SIAM, Philadelphia, PA, 2013.

[21] Bleylevens, I. W. M., Hochstenbach, M. E., and Peeters, R. L. M. Polynomial optimization and a Jacobi-Davidson type method for commuting matrices. *Applied Mathematics and Computation 224* (2013), 564–580.

[22] Bose, N. K. *Applied Multidimensional Systems Theory.* Springer International Publishing, 2017.

[23] Brachat, J., Comon, P., Mourrain, B., and Tsigaridas, E. Symmetric tensor decomposition. *Linear Algebra and its Applications 433*, 11 (2010), 1851–1872.

[24] Breiding, P., and Timme, S. HomotopyContinuation.jl: A Package for Homotopy Continuation in Julia. In *International Congress on Mathematical Software* (2018), Springer, pp. 458–465.

[25] Bronson, R. *Matrix methods: An introduction.* Gulf Professional Publishing, 1991.

[26] Büchberger, B. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal.* PhD thesis, University of Innsbruck, 1965.

[27] Chui, C. K., and Chen, G. *Signal processing and systems theory : selected topics.* Springer, 1992.

[28] Coca, D., and Billings, S. A. Identification of finite dimensional models of infinite dimensional dynamical systems. *Automatica 38*, 11 (2002), 1851–1865.

[29] Corless, R. M., Gianni, P. M., and Trager, B. M. A reordered Schur factorization method for zero-dimensional polynomial systems with multiple roots. In *Proc. of the International Symposium on Symbolic and Algebraic Computation* (1997), p. 133–140.

[30] Cotorruelo, A., Hosseinzadeh, M., Ramirez, D. R., Limon, D., and Garone, E. Reference dependent invariant sets: Sum of squares based computation and applications in constrained control. *Automatica 129* (2021), 109614.

[31] Cox, D. A., Little, J., and O'Shea, D. *Using Algebraic Geometry*, 2nd ed. Springer, 2005.

[32] Cox, D. A., Little, J. B., and O'Shea, D. *Ideals, Varieties and Algorithms.* Springer-Verlag, 2007.

[33] Datta, R. S. Using computer algebra to find Nash equilibria. In *Proceedings of the 2003 International Symposium on Symbolic and Algebraic Computation* (2003), ISSAC '03, Association for Computing Machinery, p. 74–79.

[34] Dayton, B. H., and Zeng, Z. Computing the multiplicity structure in solving polynomial systems. *Proc. of the International Symposium on Symbolic and Algebraic Computation* (2005), 116–123.

[35] De Cock, K., and De Moor, B. Multiparameter eigenvalue problems and shift-invariance. *IFAC-PapersOnLine, Part of Special Issue: 24th International Symposium on Mathematical Theory of Networks and Systems (MTNS2020) 54*, 9 (2021), 159–165.

[36] De Moor, B. Least squares optimal realisation of autonomous LTI systems is an eigenvalue problem. *Communications in Information and Systems 20*, 2 (2020), 163–207.

[37] de Silva, V., and Lim, L.-H. Tensor rank and the ill-posedness of the best low-rank approximation problem. *SIAM Journal on Matrix Analysis and Applications 30*, 3 (2008), 1084–1127.

[38] Dickenstein, A. Biochemical reaction networks: An invitation for algebraic geometers. In *Contemporary Mathematics 656*. 2016, pp. 69–97. Part of special issue: 1st Mathematical congress of the Americas.

[39] Dickenstein, A., and Emiris, I. Z. *Solving Polynomial Equations: Foundations, Algorithms, and Applications*. Springer, 2005. Series Title: Algorithms and combinatorics.

[40] Dillabough, M., Shang, H., and McLellan, P. J. A state space approach for boundary control of distributed parameter systems. In *IFAC Proceedings Volumes* (2006), vol. 39, pp. 341–346. 6th IFAC Symposium on Advanced Control of Chemical Processes.

[41] Ding, T., Sznaier, M., and Camps, O. Robust identification of 2-D periodic systems with applications to texture synthesis and classification. In *Proceedings of the 45th IEEE Conference on Decision and Control* (2006), pp. 3678–3683.

[42] Dong, B., Yu, B., and Yu, Y. A homotopy method for finding all solutions of a multiparameter eigenvalue problem. *SIAM Journal on Matrix Analysis and Applications 37*, 2 (2016), 550–571.

[43] Dreesen, P. *Back to the Roots: Polynomial System Solving Using Linear Algebra*. PhD thesis, KU Leuven, 2013.

[44] Dreesen, P., Batselier, K., and De Moor, B. Multidimensional realisation theory and polynomial system solving. *International Journal of Control 91*, 12 (2018), 2692–2704.

[45] El Korso, M. N., Boyer, R., Renaux, A., and Marcos, S. Statistical resolution limit for the multidimensional harmonic retrieval model: hypothesis test and cramér-rao bound approaches. *EURASIP Journal on Advances in Signal Processing 2011*, 1 (2011), 12.

[46] Emiris, I. Z., and Mourrain, B. Computer algebra methods for studying and computing molecular conformations. *Algorithmica 25* (1999), 372–402.

[47] Emiris, I. Z., and Mourrain, B. Matrices in elimination theory. *Journal of Symbolic Computation 28*, 1 (1999), 3–44.

[48] Evert, E., and De Lathauwer, L. Guarantees for existence of a best canonical polyadic approximation of a noisy low-rank tensor. *SIAM Journal on Matrix Analysis and Applications 43*, 1 (2022), 328–369.

[49] Evert, E., and De Lathauwer, L. On best low rank approximation of positive definite tensors. *SIAM Journal on Matrix Analysis and Applications 44*, 2 (2023), 867–893.

[50] Evert, E., Vandecappelle, M., and De Lathauwer, L. A recursive eigenspace computation for the canonical polyadic decomposition. *SIAM Journal on Matrix Analysis and Applications 43*, 1 (2022), 274–300.

[51] Farah, M., Mercère, G., Ouvrard, R., and Poinot, T. Combining least-squares and gradient-based algorithms for the identification of a co-current flow heat exchanger. *International Journal of Control 92*, 1 (2019), 191–203.

[52] Farah, M., Mercère, G., Ouvrard, R., Poinot, T., and Ramos, J. Identification of 2D Roesser models by using linear fractional transformations. In *2014 European Control Conference (ECC)* (2014), pp. 382–387.

[53] Faugère, J.-C. A new efficient algorithm for computing Gröbner bases without reduction to zero (F5). In *Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation* (2002), ACM, pp. 75–83.

[54] Faugère, J.-C. A new efficient algorithm for computing Gröbner bases (F4). *Journal of Pure and Applied Algebra 139* (1999), 61–88.

[55] Fornasini, E., and Marchesini, G. State-space realization theory of two-dimensional filters. *IEEE Trans. on Automatic Control 21* (1976), 484–492.

[56] Fornasini, E., and Marchesini, G. Doubly-indexed dynamical systems: State-space models and structural properties. *Mathematical systems theory 12*, 1 (1978), 59–72.

[57] Fornasini, E., and Marchesini, G. Algebraic realization theory of two-dimensional filters. In *Variable Structure Systems with applications to economics and biology* (1974), p. 64–82.

[58] Fornasini, E., Rocha, P., and Zampieri, S. State space realization of 2-D finite-dimensional behaviours. *SIAM Journal on Control and Optimization 31*, 6 (1993), 1502–1517.

[59] Fraanje, R., and Verhaegen, M. A spatial canonical approach to multidimensional state-space identification for distributed parameter systems. In *Proceedings of the Fourth International Workshop on Multidimensional Systems - NDS 2005* (2005), IEEE, pp. 217–222.

[60] Gantmacher, F. R. *The Theory of Matrices, Volume 2.* Chelsea Publishing Company, 1974.

[61] Gantmacher, F. R. *The Theory of Matrices, Volume 1.* Chelsea Publishing Company, 1977.

[62] Gatermann, K., and Parrilo, P. A. Symmetry groups, semidefinite programs, and sums of squares. *Journal of Pure and Applied Algebra 192*, 1 (2004), 95–128.

[63] Gelfand, I. M., Kapranov, M. M., and Zelevinsky, A. V. *Discriminants, Resultants, and Multidimensional Determinants.* Birkhäuser, Boston, MA, 1994.

[64] Gheorghiu, C. I., Hochstenbach, M. E., Plestenjak, B., and Rommes, J. Spectral collocation solutions to multiparameter Mathieu's system. *Applied Mathematics and Computation 218*, 24 (2012), 11990–12000.

[65] Golub, G. H., and Van Loan, C. F. *Matrix Computations.* Johns Hopkins University Press, 2013.

[66] Govindarajan, N., Widdershoven, R., Chandrasekaran, S., and De Lathauwer, L. A fast algorithm for computing Macaulay null spaces of bivariate polynomial systems. *SIAM Journal on Matrix Analysis and Applications 45*, 1 (2024), 368–396.

[67] Guo, D., Lu, H., and Qu, X. A fast low rank Hankel matrix factorization reconstruction method for non-uniformly sampled magnetic resonance spectroscopy. *IEEE Access 5* (2017), 16033–16039.

[68] Haardt, M., Brunner, C., and Nossek, J. Joint estimation of 2-D arrival angles, propagation delays, and Doppler frequencies to determine realistic directional simulation models for smart antennas. In *Proc. IEEE Digital Signal Processing Workshop* (1998), vol. 5.

[69] Haardt, M., and Nossek, J. 3-D unitary esprit for joint 2-D angle and carrier estimation. In *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing* (1997), vol. 1, pp. 255–258 vol.1.

[70] Hansen, L. P., and Sargent, T. J. *Recursive Models of Dynamic Linear Economies*, 6th ed. Princeton University Press, 2018.

[71] Hansson, A., and Verhaegen, M. Distributed system identification with ADMM. In *Proceedings of the IEEE Conference on Decision and Control* (2014), pp. 290–295.

[72] Hanzon, B. Attasi nD systems and polynomial system solving. *IFAC-PapersOnLine 54* (2021), 145–150. 24th International Symposium on Mathematical Theory of Networks and Systems MTNS 2020.

[73] Hanzon, B., and Jibetean, D. Global minimization of a multivariate polynomial using matrix methods. *Journal of Global Optimization 27*, 1 (2003), 1–23.

[74] Hanzon, B., and Mustata, A. On the role of the Critical Value Polynomial in algebraic optimization. In *MTNS2014* (2014). Extended abstract.

[75] Hartshorne, R. *Algebraic Geometry*, vol. 52 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1977.

[76] Hashemi, S., Boroomand, B., and Movahedian, B. Exponential basis functions in space and time: A meshless method for 2D time dependent problems. *Journal of Computational Physics 241* (2013), 526–545.

[77] Henrion, D., and Korda, M. Convex computation of the region of attraction of polynomial control systems? In *Proc. of the 2013 European Control Conference (ECC)* (2013), pp. 676–681.

[78] Henrion, D., and Lasserre, J.-B. Detecting global optimality and extracting solutions in gloptipoly. In *Positive Polynomials in Control*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005, pp. 293–310.

[79] Hiptmair, R., Moiola, A., and Perugia, I. A survey of Trefftz methods for the Helmholtz equation. In *Building bridges: connections and challenges in modern approaches to numerical partial differential equations*. Springer, 2016, pp. 237–279.

[80] Hochstenbach, M. E., Kosir, T., and Plestenjak, B. A Jacobi-Davidson type method for the two-parameter eigenvalue problem. *SIAM Journal on Matrix Analysis and Applications 26*, 2 (2005), 477–497.

[81] Hochstenbach, M. E., Kosir, T., and Plestenjak, B. Numerical methods for rectangular multiparameter eigenvalue problems, with applications to finding optimal ARMA and LTI models. *Numerical Linear Algebra with Applications 31*, 2 (2024), e2540.

[82] Horn, R. A., and Johnson, C. R. *Matrix Analysis*, 2nd ed. Cambridge University Press, 2013.

[83] Hydon, P. *Difference Equations by Differential Equation Methods*. Cambridge University Press, 2014.

[84] Janczak, A., Rogers, E., and Cai, Z. Subspace identification of process dynamics for iterative learning control. In *Proceedings of the Eight International Workshop on Multidimensional Systems - NDS* (2013), pp. 1–6.

[85] Janssens, V., De Cock, K., Lagauw, S., and Vanpoucke, L. The quest for global optimality within linear system identification. Master's thesis, KU Leuven, 2024.

[86] Jiang, T., Sidiropoulos, N. D., and Ten Berge, J. M. Almost-sure identifiability of multidimensional harmonic retrieval. *IEEE Transactions on Signal Processing 49*, 9 (2001), 1849–1859.

[87] Kaczorek, T. *Two-Dimensional Linear Systems*. Springer-Verlag, 1985.

[88] Kaczorek, T. The singular general model of 2D systems and its solution. *IEEE Transactions on Automatic Control 33*, 11 (1988), 1060–1061.

[89] Kaczorek, T. Generalization of Cayley-Hamilton theorem for n-D polynomial matrices. *IEEE Trans. on Automatic Control 50* (2005), 671–674.

[90] Kailath, T. *Linear Systems*. Prentice-Hall International, 1998. Series Title: Prentice-Hall information and system sciences series.

[91] Kalouptsidis, N. *Signal Processing Systems, Theory and Design*. John Wiley & Sons, 1997.

[92] Kehrein, A., and Kreuzer, M. Characterizations of border bases. *Journal of Pure and Applied Algebra 196*, 2 (2005), 251–270.

[93] Kenkel, J. L. *Dynamic Linear Economic Models*, 1at ed. Routledge, 1974.

[94] Koblitz, N. *Algebraic Aspects of Cryptography*, vol. 3 of *Algorithms and Computation in Mathematics*. Springer Verlag, Berlin / Heidelberg, 1998.

[95] Korda, M., Henrion, D., and Jones, C. N. Inner approximations of the region of attraction for polynomial dynamical systems. *IFAC Proceedings Volumes 46*, 23 (2013), 534–539. 9th IFAC Symposium on Nonlinear Control Systems.

[96] Korda, M., Henrion, D., and Jones, C. N. Convex computation of the maximum controlled invariant set for polynomial control systems. *SIAM Journal on Control and Optimization 52*, 5 (2014), 2944–2969.

[97] Kukelova, Z., Bujnak, M., and Pajdla, T. Automatic Generator of Minimal Problem Solvers. In *Computer Vision – ECCV 2008* (Berlin, Heidelberg, 2008), Springer Berlin Heidelberg, pp. 302–315.

[98] Kung, S.-Y., Levy, B., Morf, M., and Kailath, T. New results in 2-D systems theory, part II: 2-D state-space models—realization and the notions of controllability, observability, and minimality. *Proceedings of the IEEE 65*, 6 (1977), 945–961.

[99] Kuo, Y.-C., and Lee, T.-L. Computing the unique CANDECOMP/-PARAFAC decomposition of unbalanced tensors by homotopy method. *Linear Algebra and its Applications 556* (2018), 238–264.

[100] Kurek, J. Basic properties of q-dimensional linear digital systems. *International Journal of Control 42*, 1 (1985), 119–128.

[101] Lagauw, S. *Global Solution to Problems in System Identification and Model Order Reduction*. PhD thesis, KU Leuven, 2026.

[102] Lagauw, S., Agudelo, O. M., and De Moor, B. Globally optimal SISO H2-norm model reduction using Walsh's theorem. *IEEE Control Systems Letters 7* (2023), 1670–1675.

[103] Lagauw, S., Vanpoucke, L., and De Moor, B. Exact characterization of the global optima of least squares realization of autonomous LTI models as a multiparameter eigenvalue problem. In *2024 European Control Conference* (2024), pp. 3446–3451.

[104] Lagauw, S., Vanpoucke, L., and De Moor, B. Exact solution to the least squares realization problem as a multiparameter eigenvalue problem. *Automatica 185* (2026), 112792.

[105] Lasserre, J. B. *An Introduction to Polynomial and Semi-Algebraic Optimization*. Cambridge Texts in Applied Mathematics. Cambridge University Press, 2015.

[106] Lasserre, J. B., Henrion, D., Prieur, C., and Trélat, E. Nonlinear optimal control via occupation measures and LMI-relaxations. *SIAM Journal on Control and Optimization 47*, 4 (2008), 1643–1666.

[107] Lasserre, J. B., Laurent, M., Mourrain, B., Rostalski, P., and Trébuchet, P. Moment matrices, border bases and real radical computation. *Journal of Symbolic Computation 51* (2013), 63–85.

[108] Lasserre, J.-B., Laurent, M., and Rostalski, P. Semidefinite characterization and computation of zero-dimensional real radical ideals. *Foundations of Computational Mathematics 8*, 5 (2008), 607–647.

[109] Laurent, M. Sums of squares, moment matrices and optimization over polynomials. In *Emerging Applications of Algebraic Geometry*. Springer, New York, NY, 2009, pp. 157–270.

[110] Lazard, D. Gröbner bases, Gaussian elimination and resolution of systems of algebraic equations. In *Proceedings of the European Computer Algebra Conference on Computer Algebra (EUROCAL '83)* (1983), Springer-Verlag, pp. 146–156.

[111] Legat, B., and Jungers, R. M. Geometric control of algebraic systems. *IFAC-PapersOnLine 54*, 5 (2021), 79–84. 7th IFAC Conference on Analysis and Design of Hybrid Systems ADHS 2021.

[112] Legat, B., Parrilo, P., and Jungers, R. Certifying unstability of switched systems using sum of squares programming. *SIAM Journal on Control and Optimization 58*, 4 (2020), 2616–2638.

[113] Legat, B., Tabuada, P., and Jungers, R. M. Sum-of-squares methods for controlled invariant sets with applications to model-predictive control. *Nonlinear Analysis: Hybrid Systems 36* (2020), 100858.

[114] Legat, B., Weisser, T., Kapelevich, L., Huchette, J., Bhatia, A., Votroubek, T., Forets, M., Moat, A., Coffrin, C., C., C., Saba, E., Hanson, E., TagBot, J., Lubin, M., Tu, S., and Kelman, T. Sumofsquares.jl: Sum of squares programming for julia, 2021.

[115] 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 (1987), 1507–1517.

[116] Lemmerling, P., and De Moor, B. Misfit versus latency. *Automatica 37*, 12 (2001), 2057–2067.

[117] Lewis, F. A review of 2-D implicit systems. *Automatica 28*, 2 (1992), 345–354.

[118] Lewis, F. L. A survey of linear singular systems. *Circuits Systems and Signal Processing 5* (1986), 3–36.

[119] Li, Y., Razavilar, J., and Liu, K. R. A high-resolution technique for multidimensional NMR spectroscopy. *IEEE Transactions on Biomedical Engineering 45*, 1 (1998), 78–86.

[120] Ljung, L. *System Identification: Theory for the User*. Pearson Education, 1998.

[121] Ljung, L. Estimating linear time-invariant models of nonlinear time-varying systems. *European Journal of Control 7*, 2 (2001), 203–219.

[122] Lomadze, V. The PBH test for multidimensional ltid systems. *Automatica 49*, 9 (2013), 2933–2937.

[123] Macaulay, F. S. *The algebraic theory of modular systems*. Cambridge University Press, 1916.

[124] Marinari, M. G., Möller, H. M., and Mora, T. Gröbner bases of ideals defined by functionals with an application to ideals of projective points. *Applicable Algebra in Engineering, Communication and Computing 4* (1993), 103–145.

[125] Markovsky, I., and Dörfler, F. Behavioral systems theory in data-driven analysis, signal processing, and control. *Annual Reviews in Control 52* (2021), 42–64.

[126] Mertzios, B. G., and Christodoulou, M. A. On the generalized Cayley-Hamilton theorem. *IEEE Trans. on Automatic Control 31* (1986), 156–157.

[127] Miri, S., and Aplevich, J. D. Modeling and realization of n-dimensional linear discrete systems. *Multidimensional Systems and Signal Processing 9*, 3 (1998), 241–253.

[128] Mokios, K., Sidiropoulos, N., Pesavento, M., and Mecklenbrauker, C. On 3D harmonic retrieval for wireless channel sounding. In *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing* (2004), vol. 2, pp. ii–89.

[129] Möller, H. M., and Buchberger, B. The construction of multivariate polynomials with preassigned zeros. In *Computer Algebra* (1982), Springer Berlin Heidelberg, pp. 24–31.

[130] Morf, M., Levy, B., and Kung, S.-Y. New results in 2-D systems theory, part I: 2-D polynomial matrices, factorization, and coprimeness. *Proceedings of the IEEE 65*, 6 (1977), 861–872.

[131] Morton, K. W., and Mayers, D. F. *Numerical solution of partial differential equations: an introduction*. Cambridge university press, 2005.

[132] Mourrain, B. Isolated points, duality and residues. *Journal of Pure and Applied Algebra 117-118* (1997), 469–493.

[133] Mourrain, B. A new criterion for normal form algorithms. *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes. AAECC 1999* (1999), 430–443.

[134] Mourrain, B., and Trebuchet, P. Generalized normal forms and polynomial system solving. In *Proceedings of the 2005 International Symposium on Symbolic and Algebraic Computation* (New York, NY, USA, 2005), ISSAC '05, Association for Computing Machinery, p. 253–260.

[135] Mourrain, B., and Trébuchet, P. Border basis representation of a general quotient algebra. In *Proceedings of the 37th International Symposium on Symbolic and Algebraic Computation* (New York, NY, USA, 2012), ISSAC '12, Association for Computing Machinery, p. 265–272.

[136] Mourrain, B., and Trébuchet, P. Stable normal forms for polynomial system solving. *Theoretical Computer Science 409*, 2 (2008), 229–240. Symbolic-Numerical Computations.

[137] Mourrain, B., and Trébuchet, P. Border basis representation of a general quotient algebra. In *Proc. of the 37th Interantional Symposium on Symbolic and Algebraic Computation* (2012), pp. 265–272.

[138] Nocedal, J., and Wright, S. *Numerical Optimization*. Springer, 2006.

[139] Oberst, U. Multidimensional constant linear systems. *Acta Applicandae Mathematicae 20* (1990), 1–175.

[140] Oberst, U. Variations on the fundamental principle for linear systems of partial differential and difference equations with constant coefficients. *Applicable Algebra in Engineering, Communication and Computing 6*, 4 (1995), 211–243.

[141] Oberst, U. Canonical state representations and Hilbert functions of multidimensional systems. *Acta Applicandae Mathematicae 94*, 2 (2006), 83–135.

[142] Oberst, U., and Pauer, F. The constructive solution of linear systems of partial difference and differential equations with constant coefficients. *Multidimensional Systems and Signal Processing 12*, 3 (2001), 253–308.

[143] O'Meara, K. C., Clark, J., and Vinsonhaler, C. I. *Advanced Topics in Linear Algebra*. Oxford University Press, 2011.

[144] Pham, D., Mercère, G., Ouvrard, R., Poinot, T., and Pálsson, H. Fouling detection in a parallel flow heat exchanger via a Roesser model identification procedure. In *IFAC-PapersOnLine* (2017), vol. 50, pp. 12866–12871. 20th IFAC World Congress.

[145] Pintelon, R., and Schoukens, J. *System Identification: A Frequency Domain Approach*. Wiley, 2004.

[146] Plestenjak, B. A continuation method for a right definite two-parameter eigenvalue problem. *SIAM Journal on Matrix Analysis and Applications 21*, 4 (2000), 1163–1184.

[147] Plestenjak, B. MultiParEig: Toolbox for multiparameter eigenvalue problems, 2024.

[148] Plestenjak, B., Gheorghiu, C. I., and Hochstenbach, M. E. Spectral collocation for multiparameter eigenvalue problems arising from separable boundary value problems. *Journal of Computational Physics 298* (2015), 585–601.

[149] Plestenjak, B., and Hochstenbach, M. E. Roots of bivariate polynomial systems via determinantal representations. *SIAM Journal on Scientific Computing 38*, 2 (2016), A765–A788.

[150] Pons, A., and Gutschmidt, S. Multiparameter spectral analysis for aeroelastic instability problems. *Journal of Applied Mechanics 85*, 6 (2018).

[151] Qin, Z. *Hilbert Schemes of Points and Infinite Dimensional Lie Algebras*, vol. 228 of *Mathematical Surveys and Monographs*. American Mathematical Society, Providence, RI, 2018.

[152] Rabinowitsch, J. L. Zum Hilbertschen Nullstellensatz. *Mathematische Annalen 102*, 1 (1930), 520–520.

[153] Ramos, J. A subspace algorithm for identifying 2-D separable in denominator filters. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing 41*, 1 (1994), 63–67.

[154] Ramos, J. A., and Mercère, G. Subspace algorithms for identifying separable-in-denominator 2D systems with deterministic–stochastic inputs. *International Journal of Control 89*, 12 (2016), 2584–2610.

[155] 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 91*, 12 (2018), 2743–2771.

[156] Rapisarda, P. Discrete Roesser state models from 2D frequency data. *Multidimensional Systems and Signal Processing 30* (2019).

[157] Regalia, P. *Adaptive IIR filtering in Signal Processing and Control*. CRC Press, 1994.

[158] Reid, G., Tang, J., and Zhi, L. A complete symbolic-numeric linear method for camera pose determination. In *Proc. 1997 Int. Symp. Symb. Algebraic Comput. (ISSAC)* (2003), ACM, pp. 215–223.

[159] Rocha, P., and Willems, J. State for 2-D systems. *Linear Algebra and Its Applications 122-124* (1989), 1003–1038.

[160] Rocha, P., and Willems, J. Controllability of 2-D systems. *IEEE Transactions on Automatic Control 36*, 4 (1991), 413–423.

[161] Rodriguez, J. I., Du, J.-H., You, Y., and Lim, L.-H. Fiber product homotopy method for multiparameter eigenvalue problems. *Numerische Mathematik 148*, 4 (2021), 853–888.

[162] Roesser, R. P. A discrete state-space model for linear image processing. *IEEE Trans. on Automatic Control 20* (1975), 1–10.

[163] Rogers, E., Galkowski, K., Paszke, W., Moore, K. L., Bauer, P. H., Hladowski, L., and Dabkowski, P. Multidimensional control systems: case studies in design and evaluation. *Multidimensional Systems and Signal Processing 26*, 4 (2015), 895–939.

[164] Sahnoun, S., Usevich, K., and Comon, P. Multidimensional esprit for damped and undamped signals: Algorithm, computations, and perturbation analysis. *IEEE Transactions on Signal Processing 65*, 22 (2017), 5897–5910.

[165] Schur, J. Zur theorie der vertauschbaren matrizen. *Journal für die reine und angewandte Mathematik 130* (1905), 66–76.

[166] Sidiropoulos, N. D., and Bro, R. On the uniqueness of multilinear decomposition of N-way arrays. *Journal of Chemometrics 14*, 3 (2000), 229–239.

[167] Sommese, A. J., and Wampler, C. W. *The Numerical Solution of Systems of Polynomials Arising in Engineering and Science*. World Scientific Publishing Company, Singapore, 2005.

[168] Sørensen, M., and De Lathauwer, L. Multidimensional harmonic retrieval via coupled canonical polyadic decomposition—part I: Model and identifiability. *IEEE Transactions on Signal Processing 65*, 2 (2016), 517–527.

[169] Stetter, H. J. *Numerical Polynomial Algebra*. SIAM, 2004.

[170] Strauss, W. A. *Partial differential equations: An introduction*. John Wiley & Sons, 2007.

[171] Strogatz, S. H. *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*, 2nd ed. CRC Press, Boca Raton, FL, 2015.

[172] Sturmfels, B. *Solving Systems of Polynomial Equations*. American Mathematical Society, 2002.

[173] Söderström, T. *Errors-in-Variables Methods in System Identification*. Springer International Publishing, 2018. Series Title: Communications and Control Engineering.

[174] Sørensen, M., and De Lathauwer, L. Blind signal separation via tensor decomposition with vandermonde factor: Canonical polyadic decomposition. *IEEE Transactions on Signal Processing 61*, 22 (2013), 5507–5519.

[175] Telen, S. Numerical root finding via Cox rings. *Journal of Pure and Applied Algebra 224*, 9 (2020), 106367.

[176] Telen, S. *Solving Systems of Polynomial Equations*. PhD thesis, KU Leuven, 2020.

[177] Telen, S., Mourrain, B., and Barel, M. V. Solving polynomial systems via truncated normal forms. *SIAM Journal on Matrix Analysis and Applications 39*, 3 (2018), 1421–1447.

[178] Telen, S., and Van Barel, M. A stabilized normal form algorithm for generic systems of polynomial equations. *Journal of Computational and Applied Mathematics 342* (2018), 119–132.

[179] Telen, S., Van Barel, M., and Verschelde, J. A robust numerical path tracking algorithm for polynomial homotopy continuation. *SIAM Journal on Scientific Computing 42*, 6 (2020), A3610–A3637.

[180] Tisseur, F., and Meerbergen, K. The quadratic eigenvalue problem. *SIAM Review 43*, 2 (2001), 235–286.

[181] Trébuchet, P., Mourrain, B., and Bucero, M. A. Border basis for polynomial system solving and optimization. In *ICMS 2016* (2016), Springer, pp. 212–220.

[182] Umansky, D., and Patzold, M. A two-dimensional autoregressive model for mimo wideband mobile radio channels. In *IEEE GLOBECOM 2008 - Global Telecommunications Conference* (2008), pp. 1–6.

[183] Van Overschee, P., and De Moor, B. N4sid: Subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica 30* (1994), 75–93.

[184] Vanderstukken, J., and De Lathauwer, L. Systems of polynomial equations, higher-order tensor decompositions, and multidimensional harmonic retrieval: A unifying framework. part I: The canonical polyadic decomposition. *SIAM Journal on Matrix Analysis and Applications 42*, 2 (2021), 883–912.

[185] Vanderstukken, J., Kürschner, P., Domanov, I., and De Lathauwer, L. Systems of polynomial equations, higher-order tensor decompositions, and multidimensional harmonic retrieval: A unifying framework. part II: The block term decomposition. *SIAM Journal on Matrix Analysis and Applications 42*, 2 (2021), 913–953.

[186] Vanpoucke, L., and De Moor, B. Constructing multidimensional difference equations from a state-space representation using the Generalized Cayley-Hamilton theorem. *IEEE Control Systems Letters 8* (2024), 2259–2264.

[187] Vanpoucke, L., Lagauw, S., and De Moor, B. Exploiting the Macaulay matrix structure for the globally optimal identification of autonomous systems, 2024. Poster presented at ENRSI 2024.

[188] Vermeersch, C. *The (Block) Macaulay Matrix: Solving Systems of Multivariate Polynomial Equations and Multiparameter Eigenvalue Problems.* PhD thesis, KU Leuven, 2023.

[189] Vermeersch, C., and De Moor, B. Globally optimal least-squares arma model identification is an eigenvalue problem. *IEEE Control Systems Letters 3* (2019), 1062–1067.

[190] Vermeersch, C., and De Moor, B. A column space based approach to solve systems of multivariate polynomial equations. *IFAC-PapersOnLine, Part of Special Issue: 24th International Symposium on Mathematical Theory of Networks and Systems (MTNS) 54*, 9 (2021), 137–144.

[191] Vermeersch, C., and De Moor, B. Two complementary block Macaulay matrix algorithms to solve multiparameter eigenvalue problems. *Linear Algebra and its Applications 654* (2022), 177–209.

[192] Verschelde, J. Algorithm 795: PHCpack: A general-purpose solver for polynomial systems by homotopy continuation. *ACM Transactions on Mathematical Software 25*, 2 (1999), 251–276.

[193] Verschelde, J., and Cools, R. Symmetric homotopy construction. *Journal of Computational and Applied Mathematics 50*, 1 (1994), 575–592.

[194] Wampler, C. W., and Sommese, A. J. Numerical algebraic geometry and algebraic kinematics. *Acta Numerica 20* (2011), 469–567.

[195] Widdershoven, R. *Fast and Robust Methods for Noisy Overdetermined Systems of Multivariate Polynomial Equations*. PhD thesis, KU Leuven, 2025.

[196] Widdershoven, R., Govindarajan, N., and De Lathauwer, L. Overdetermined systems of polynomial equations: Tensor-based solution and application. In *2023 31st European Signal Processing Conference (EUSIPCO)* (2023), pp. 650–654.

[197] Willems, J. From time series to linear system - part I. finite dimensional linear time invariant systems. *Automatica 22* (1986), 561–580.

[198] Willems, J. C. From time series to linear system – part III. approximate modeling. *Automatica 23*, 1 (1987), 87–115.

[199] Willems, J. C., Rapisarda, P., Markovsky, I., and De Moor, B. L. A note on persistency of excitation. *Systems and Control Letters 54*, 4 (2005), 325–329.

[200] Yang, R., Xie, L., and Zhang, C. H2 and mixed H2/H∞ control of two-dimensional systems in Roesser model. *Automatica 42*, 9 (2006), 1507–1514.

[201] Ying, J., Lu, H., Wei, Q., Cai, J.-F., Guo, D., Wu, J., Chen, Z., and Qu, X. Hankel matrix nuclear norm regularized tensor completion for n-dimensional exponential signals. *IEEE Transactions on Signal Processing 65*, 14 (2017), 3702–3717.

[202] Zandi, S., Boroomand, B., and Soghrati, S. Exponential basis functions in solution of problems with fully incompressible materials: A mesh-free method. *Journal of Computational Physics 231*, 21 (2012), 7255–7273.

[203] Zeng, Z. A numerical elimination method for polynomial computations. *Theoretical Computer Science 409*, 2 (2008), 318–331.

[204] Zeng, Z. The closedness subspace method for computing the multiplicity structure of a polynomial system. In *Contemporary Mathematics*, vol. 496. American Mathematical Society, Providence, Rhode Island, 2009, pp. 347–362.

[205] Zerz, E. *Topics in Multidimensional Linear Systems Theory*. Springer-Verlag London, 2000.

[206] Zvonarev, N., and Golyandina, N. Low-rank signal subspace: parameterization, projection and signal estimation. *Statistics and Its Interface 16* (2023), 117–132.

[207] Åke Björck, and Golub, G. H. Numerical methods for computing angles between linear subspaces. *Mathematics of Computation 27*, 123 (1973), 579–594.

[208] Åström, K., and Wittenmark, B. On self tuning regulators. *Automatica 9*, 2 (1973), 185–199.

Statement on the use of Generative AI

I used the generative AI assistance tools: ChatGPT and MS Copilot. ChatGPT was used as a search engine to learn about particular topics. The main use was to briefly clarify key concepts and get a good starting point before turning to literature for a more in-depth study. ChatGPT was used for carrying out a literature review. This was used mainly as a complementary tool in addition to classical search engines, to get an overview of the related topics and the relevant references as a starting point. The resulting statements and sources were always checked during a more in-depth analysis afterward. MS Copilot was used to generate visuals and tables. This was done by providing the data and/or a description of the desired visual or table. The main use was to quickly and adaptively generate such visuals and the result was always finetuned afterward. The text, code, and images in this thesis are my own (unless otherwise specified). Generative AI has only been used in accordance with the KU Leuven guidelines and appropriate references have been added. I have reviewed and edited the content as needed and I take full responsibility for the content of the thesis.

List of publications

Published

- **Vanpoucke, L.**, and De Moor, B. "Constructing multidimensional difference equations from a state-space representation using the Generalized Cayley-Hamilton theorem". IEEE Control Systems Letters 8 (2024), 2259–2264. DOI: 10.1109/LCSYS.2024.3411835
- Lagauw, S., **Vanpoucke, L.**, and De Moor, B. "Exact characterization of the global optima of least squares realization of autonomous LTI models as a multiparameter eigenvalue problem". 3446–3451. DOI: 10.23919/ECC64448.2024.10591182
- Lagauw, S., **Vanpoucke, L.**, and De Moor, B. "Exact solution to the least squares realization problem as a multiparameter eigenvalue problem". Automatica 185 (2026), 112792. DOI: 10.1016/j.automatica.2025.112792

Submitted

- **Vanpoucke L.**, Lagauw S., De Moor B. "Globally optimal least squares misfit identification of multidimensional, autonomous, linear, shift-invariant systems", Tech. rep. `https://ftp.esat.kuleuven.be/pub/stadius//lvanpouc/misfit.pdf`, Leuven, Belgium: KU Leuven 2026. *Submitted to Automatica*

FACULTY OF ENGINEERING SCIENCE
DEPARTMENT OF ELECTRICAL ENGINEERING
Celestijnenlaan 200A box 2402
B-3001 Leuven
lukas.vanpoucke@esat.kuleuven.be

