Computational Methods for Model Reduction of Large-Scale Sparse Structured Descriptor Systems

Dissertation
zur Erlangung des akademischen Grades

doctor rerum naturalium
(Dr. rer. nat.)

von Mohammad Monir Uddin
(M. Sc. in Applied Mathematics)

geb. am 30.11.1978 in Chittagong, Bangladesh

genehmigt durch die Fakultät für Mathematik
der Otto-von-Guericke-Universität Magdeburg

Gutachter: Prof. Dr. Peter Benner
Prof. Dr. Matthias Heinkenschloss

eingereicht am: 26.02.2015
Verteidigung am: 30.06.2015

Acknowledgements

With my great pleasure, first, I would like to express my cordial thanks to my teacher and supervisor Prof. Peter Benner for his remarkable contribution and guidance to complete this thesis. I am very grateful to him for introducing me to this highly fascinating and applicable research area. I would like to express my hearty gratitude to my suprvisor Prof. Matthias Heinkenschloss for his useful suggessions for the thesis. I would also like to thank my advisor Dr. Jens Saak for his important suggestions, essential discussions and cordial guidance to finish this work successfully.

Further, I am thankful to Dr. Mian Ilyas Ahmad for his collaboration to discuss some problems regarding the thesis. I want to thank all the colleagues that I worked with during my time in the research group ’Computational Methods in System and Control Theory’. I am specially grateful to Dr. Lihong Feng, Dr. Ulrike Baur, Matthias Voigt and Patrick Kürschner for their valuable times to discuss some problems related to my thesis. Heiko Weichelt and Dr. Burkhard Kranz provided me data for numerical experiments and also some pictures. In different situations Dr. Jürgen Koch helped me a lot by giving proper suggestions and motivations. I am much obliged to all of them. I was also affiliated to the "International Max Plank Research School (IMPRS) for advanced Methods in Process and System Engineering (Magdeburg)".

Declaration of Honor

I hereby declare that I produced this thesis without prohibited assistance and that all sources of information that were used in producing this thesis, including my own publications, have been clearly marked and referenced.

In particular I have not wilfully:

- Fabricated data or ignored or removed undesired results.
- Misused statistical methods with the aim of drawing other conclusions than those warranted by the available data.
- Plagiarised data or publications or presented them in a disorted way.

I know that violations of copyright may lead to injunction and damage claims from the author or prosecution by the law enforcement authorities.

This work has not previously been submitted as a doctoral thesis in the same or a similar form in Germany or in any other country. It hast not previously been published as a whole.

February 26, 2015

....

(Mohammad Monir Uddin)

Abstract

Currently descriptor systems, i.e., the systems whose dynamics obey differential-algebraic equations (DAEs), play important roles in various disciplines of science and technology. In general, such systems are generated by finite element or finite difference methods. If the grid resolution becomes very fine, because many details must be resolved, the systems become very large. Moreover they are sparse, i.e., most of the elements in the matrices of the system are zero, which are not stored. A high dimensional system will always be complex, requiring a great deal of memory, thereby hindering computational performance significantly in simulation. Sometimes the systems are too large to store due to memory restrictions. Therefore, we seek to reduce the complexity of the model by applying model order reduction (MOR), i.e., we seek an approximation of the original model that well-approximates the behavior of the original model, yet is much faster to evaluate. We investigate efficient model reduction of sparse large-scale descriptor systems. We focus on the balancing based method *balanced truncation* (BT). A balanced truncation based method for such systems is introduced by Stykel (see, e.g., her PhD thesis, published in 2002). The author discusses a general framework of the BT method for a descriptor system. In general, the method is based on explicit computation of the spectral projectors onto the left and right deflating subspaces of the matrix pencil corresponding to the finite and infinite eigenvalues. Although these projectors are available for particular systems, computation is expensive. In this thesis, we focus on how to avoid computing such kind of projectors explicitly. Besides balanced truncation, the idea of avoidance of the projectors is extended to interpolation of transfer function, via *iterative rational Krylov algorithms* (IRKA) and *projection onto dominant eigenspace, of the Gramian* (PDEG) based model reduction methods. First, we discuss the model reduction problem for index 2 first order unstable descriptor systems arising from spatially discretized linearized Navier-Stokes equations. We apply our algorithms to the linearization of the von Kármán vortex shedding at a moderate Reynolds number. We demonstrate that the resulting reduced model can be used to accurately simulate the unstable linearized model and to design a stabilizing controller. Future work will include the realization of the resulting control law for the full nonlinear model. Second, we investigate model reduction of a finite element model of a spindle head configuration in a machine tool. The special feature of this spindle head is that it is partially driven by a set of piezo actuators. Due to this piezo actuation, the resulting model is a second order differential-algebraic system of index 1. We develop algorithms for both second-order-to-first-order and second-order-to-second-order reduction methods. We prove the real world capability of our methods in application to a very large-scale sparse FEM model of an adaptive spindle support employing piezo actuators. Finally, we focus on the model reduction of DAE systems with mechanical applications. In the constraint mechanics or multibody dynamics, the linearized equation of motion with holonomic constraints leads to second order index 3 descriptor systems. We develop efficient techniques to obtain second-order-to-first-order and second-order-to-second-order reduced models of such index 3 descriptor systems. The efficiency of the techniques is tested by applying them to several test examples. For implementing the BT and PDEG methods, we need to compute approximate low rank Gramian factors of the system by solving two continuous-time Lyapunov equations. Recently one of the most powerful methods to compute these Gramian factors for large-scale sparse dynamical systems is the *low-rank Cholesky factor alternating direction implicit* (LRCF-ADI) iteration. We also present updated versions of the LRCF-ADI method to solve the Lyapunov equations arising from descriptor systems. Moreover, several approaches for computing ADI shift parameters are discussed and proposed for an improvement of an existing method.

Zusammenfassung

Heutzutage spielen Deskriptorsysteme, also solche Systeme deren Dynamik differentiell algebraischen Gleichungen gehorchen, eine wichtige Rolle in verschiedensten Disziplinen in Wissenschaft und Technik. Im Allgemeinen werden solche Systeme durch finite-Elemente- oder finite-Differenzen-Verfahren erzeugt. Wenn die Gitterauflösung sehr fein wird, da viele Details aufgelöst werden müssen, dann werden die Systeme sehr groß. Darüberhinaus sind sie dünn besetzt, d.h. die meisten Einträge der Systemmatrizen sind Nullen, die nicht gespeichert werden. Hochdimensionale Systeme benötigen viel Speicher, wodurch die Ausführungseffizienz von Simulationen merkbar beeinträchtigt wird. Manchmal sind die Systeme sogar zu groß für die begrenzten verfügbaren Speicherkapazitäten. Daher sind wir bestrebt die Komplexität der Modelle durch eine Modellordnungsreduktion (MOR) zu verringern, d.h. wir suchen nach einer Approximation des Originalsystems, die das Verhalten des Originalsystems möglichst genau wiedergibt, dabei aber viel schneller auszuwerten ist. Wir untersuchen effiziente Modellreduktion von großen dünnbesetzten Deskriptorsystemen. Dabei konzentrieren wir uns auf die Methode des balancierten Abschneidens (BT für balanced truncation). Ein Verfahren für das balancierte Abschneiden von solchen Systemen wurde von Stykel (vgl. ihre Dissertation aus dem Jahr 2002) vorgestellt. Die Autorin diskutiert darin eine allgemein gültige Verfahrensweise für das balancierte Abschneiden von Deskriptorsystemen. Im allgemeinen basiert das Verfahren auf der expliziten Berechnung von Spektralprojektoren zu den linken und rechten Eigenräumen zu den endlichen und unendlichen Eigenwerten. Obwohl diese Projektoren für ausgewählte Systeme bekannt sind, ist ihre Berechnung teuer. In dieser Arbeit konzentrieren wir uns darauf, die explizite Berechnung der Projektoren zu vermeiden. Diese Idee (Vermeidung der Projektoren) wird außerdem übertragen auf die Interpolation der Übertragungsfunktion durch den iterativen rationalen Krylov Algorithmus (IRKA) und die Projektion auf dominante Eigenräume der Systemgramschen (PDEG für projection onto dominant eigenspaces of the Gramian) zur Modellreduktion. Zunächst betrachten wir das Modellreduktionsproblem für instabile Index-2 Deskriptorsysteme erster Ordnung, wie sie bei ortsdiskretisierten linearisierten Navier-Stokes-Gleichungen entstehen. Wir wenden unseren Algorithmus auf die von Kármán'sche Wirbelstraße mit moderaten Reynoldszahlen an. Dabei zeigen wir, dass das reduzierte System verwendet werden kann, um das instabile Originalsystem genau zu simulieren und auch eine stabilisierende Regelung zu entwerfen. Zukünftige Arbeiten umfassen die Realisierung des Regelgesetzes für das vollständige nichtlineare Modell. Als zweites untersuchen wir Modellreduktion eines finite-Elemente-Modells einer Werkzeugspindel in einer Werkzeugmaschine. Das auszeichnende Merkmal dieser ist, dass sie teilweise von Piezoaktuatoren bewegt wird. Bedingt durch diese Piezoaktoren ist das resultierende Modell ein differentiell-algebraisches System vom Index 1. Wir entwickeln Algorithmen sowohl für Methoden der Reduktion erster als auch zweiter Ordnung. Wir demonstrieren die Verwendbarkeit unserer Methoden in Anwendung auf ein sehr großskaliges dünn-besetztes FEM Modell einer Werkzeughalterung, welche Piezoaktoren verwendet. Schließlich konzentrieren wir uns auf die Modellreduktion von DAE-Systemen aus mechanischen Anwendungen. In der beschränkten Mechanik oder Mehrkörperdynamik führen die linearisierten Bewegungsgesetze mit holonomen Beschränkungen auf Index 3 Deskriptorsysteme. Wir entwickeln effiziente Techniken um zweiter-Ordnung-zu-zweiter-Ordnung wie auch zweiter-Ordnung-zu-erster-Ordnung reduzierte Modelle für derartige Index 3 Systeme zu erhalten. Die Effizienz der Techniken wird durch Anwendung auf diverse Beispielsysteme getestet. Zur Implementierung der BT und PDEG Methoden müssen wir approximativ Gramsche-Matrizen im Niedrigrangformat berechnen. Dies geschieht durch das Lösen zweier zeitkontinuierlicher Lyapunovgleichungen. Eines der mächtigsten Verfahren zur Berechnung dieser Gramschen-Matrizen für große, dünn besetzte dynamische Systeme ist die Niedrigrang-Choleskyfaktor-ADI (LRCF-ADI) Iteration. Wir zeigen auch eine überarbeitete Version der LRCF-ADI Methode zum Lösen von Lyapunovgleichungen die aus Deskriptorsystemen entstehen. Darüberhinaus werden verschiedene Zugänge zur Berechnung der ADI Parameter diskutiert und eine Verbesserung einer existierenden Methode vorgeschlagen.

Contents

List of Figures xv

List of Tables xix

List of Algorithms xxi

Notations and Symbols xxiv

List of Acronyms xxvi

1 Introduction 1

1.1 Motivation 1

1.2 Thesis outline 5

2 Preliminaries 7

2.1 Theory of systems 7

2.1.1 State space form of dynamical systems 7

2.1.2 Controllability and observability of the systems 9

2.1.3 System Hankel singular values 12

2.1.4 Realizations 12

2.1.5 Second order systems 13

2.2 Model reduction 15

2.2.1 Balanced truncation 17

2.2.2 Interpolatory projections 20

2.2.3 Model reduction of a second order system 22

2.3 The LRCF-ADI iteration and related issues 24
2.4 Model reduction of descriptor systems 29
2.4.1 Descriptor systems 29
2.4.2 MOR of structured descriptor systems 30

3 First Order Index 2 Unstable Descriptor Syatems 33

3.1 Motivating example 34
3.2 BT for unstable systems 35
3.3 BT for index 2 unstable descriptor systems 37
3.4 Solution of the projected Lyapunov equations 39
3.4.1 GS-LRCF-ADI for index 2 unstable systems 40
3.4.2 ADI shift parameter selection 42
3.5 Riccati-based feedback stabilization from ROM 44
3.6 Numerical results 46
3.6.1 Test examples and hardware 46
3.6.2 GS-LRCF-ADI and balancing based MOR 47
3.6.3 Comparison of BT with interpolatory technique 51

4 Second Order Index 1 Descriptor Systems 53

4.1 Motivating example 55
4.2 Second-order-to-first-order reduction techniques 57
4.2.1 Balancing based method 58
4.2.2 Interpolatory method 60
4.3 Second-order-to-second-order MOR techniques 61
4.3.1 Balancing based method 62
4.3.2 Dominant eigenspace projection of the Gramian 64
4.4 Efficient solution of the Lyapunov equation 65
4.4.1 Generalized sparse (GS)-LRCF-ADI iteration 66
4.4.2 ADI shift parameter selection 68
4.5 Numerical results 69
4.5.1 Second-order-to-first-order reduction 70

4.5.2 Second-order-to-second-order reduction 70

5 Second Order Index 3 Descriptor Systems 77

5.1 Motivating examples 78

5.2 Index reduction 79

5.2.1 Reformulation of the dynamical systems 79

5.2.2 Equivalent finite spectra 82

5.3 Model reduction 85

5.3.1 Second-order-to-first-order reduction 85

5.3.2 Second-order-to-second-order reduction 89

5.4 Solution of the projected Lyapunov equations 93

5.4.1 GS-LRCF-ADI iteration 93

5.4.2 ADI shift parameter selection 97

5.5 Numerical results 99

5.5.1 Test examples and hardware 99

5.5.2 GS-LRCF-ADI iteration 101

5.5.3 Second-order-to-first-order reduction 102

5.5.4 Second-order-to-second-order-reduction 103

6 Conclusion 109

6.1 Summary 109

6.2 Future work 111

Bibliography 113

List of Figures

3.1 Initial discretization of the *von Kármán vortex street* with coordinates, boundary parts and observation points. 35

3.2 Flow field for Re=300. 45

3.3 Comparisons of the heuristic and adaptive shifts in computing the low-rank Gramian factors using the GS-LRCF-ADI iteration. 48

a $\tilde{R}$. 48

b $\tilde{L}$. 48

c $\tilde{R}$. 48

d $\tilde{L}$. 48

3.4 Comparison of the full and reduced models in frequency domain. . . . 49

a Sigma plot. 49

b Absolute error. 49

c Relative error. 49

3.5 Step responses of 1st input to 1st output of full and reduced-order models and respective absolute deviations. 50

a Step response. 50

b Absolute error. 50

3.6 Step responses of 1st input to 1st output of stabilized full and reduced-order models and respective absolute deviations. 50

a Step response. 50

b Absolute error. 50

3.7 Errors between the full and 50 dimensional reduced systems computed by BT and IRKA using the system Mod-5. 52

a Absolute error. 52

b Relative error. 52

4.1 Piezo-actuator based mechanical system 55

4.2 The adaptive spindle support: CAD-model (left) and real component mounted on the test bench (right). 56

4.3 Detail of the finite element mesh of the adaptive spindle support. . . 56

4.4 Comparison of different dimensional reduced systems obtained by different tolerances. 71

a Sigma plot 71

b Absolute error 71

c Relative error 71

4.5 Comparison of different dimensional reduced systems with the original system. 72

a Sigma plot 72

b Relative error 72

4.6 Relative error for different dimensional ROMs computed by Algorithms 8 and 9. 73

4.7 Comparison of different dimensional reduced systems obtained by different balancing levels using truncation tolerance 10^{-5}. 74

a Sigma plot 74

b Absolute error 74

c Relative error 74

4.8 Relative error between full and different dimensional reduced models via balanced truncation. 74

a Velocity-velocity balancing. 74

b Position-position balancing. 74

4.9 1st, 2nd, 3rd and 4th rows respectively, show the 1st input to 1st output, 9th input to 1st output, 1st input to 9th and 9th input to 9th output relations (left) and the respective relative errors of full and different dimensional reduced systems. 75

4.10 Relative error between full and different dimensional reduced models via projecting onto the dominant eigen space of the Gramians. . . 76

a Velocity Gramian. 76

b Position Gramian. 76

4.11 Eigenvalues of the ROM via projecting onto the dominant eigenspace of the position Gramian. 76

5.1 A spring-mass-damper system with holonomic constraint. 100

5.2 Triple chain oscillation. 100

5.3 Comparison of full and first order reduced model of the DSMS example.102

a Sigma plot. 102

b Absolute error. 102

5.4 Comparison of different dimensional first order reduced and original models in the frequency domain for the TCOM example. 103

a Transfer function. 103

b Absolute error. 103

c Relative error. 103

5.5 Comparison of full 55 dimensional ROMs for the TCOM example in the time domain. 104

a Step response. 104

b Absolute error. 104

c Relative error. 104

5.6 Comparison of balanced truncation and IRKA with a 60 dimensional reduced model of the TCOM example. 106

a Absolute error. 106

b Relative error. 106

5.7 Comparison of full and reduced models via balancing on different levels for the TCOM model. 107

a Sigma plot. 107

b Absolute error. 107

c Relative error. 107

5.8 Comparison of full and reduced models via balancing on different levels for the DSMS model. 107

a Sigma plot. 107

b Absolute error. 107

5.9 PDEG based 45 dimensional ROMs for the TCOM example. 108
a Absolute error. 108
b Relative error. 108
5.10 Eigenvalues of the ROMs obtained by the PDEG method. 108

List of Tables

3.1 The number of differential and algebraic variables of different discretization levels of the model. 46

3.2 The performances of the heuristic and adaptive shifts in the GS-LRCF-ADI iteration. 46

3.3 Dimensions of original and reduced systems of the different sizes models for a fixed balanced truncation tolerance. 47

3.4 Balanced truncation tolerances and dimensions of reduced models. . 47

3.5 Comparisons of balanced truncation and IRKA for different sized models and their 50 dimensional reduced models. 51

4.1 Balancing transformations for the second order index 1 descriptor systems. 63

4.2 Comparison of the normalized residual norms using heuristic and adaptive shifts at different iteration steps in Algorithm 12. 69

4.3 ROMs obtained by using different truncation tolerances. 70

4.4 Average execution time and speedup against full order model for computing the maximum Hankel singular value at a given sampling frequency. 73

5.1 Eigenvalues for the matrix polynomials $Q(\lambda)$, $\tilde{Q}(\lambda)$ and $\bar{Q}(\lambda)$, defined in (5.25-5.27). 84

5.2 The performances of the heuristic and adaptive shifts in the GS-LRCF-ADI method with respect to iteration number. 101

5.3 The performances of the heuristic and adaptive shifts in the GS-LRCF-ADI method with respect to computational time. 101

5.4 Comparisons of balancing and IRKA based methods for different dimensional ROMs with TCOM example. 105

List of Algorithms

1 LR-SRM. 19

2 IRKA for MIMO systems. 22

3 G-LRCF-ADI iteration. 25

4 G-LRCF-ADI iteration (for a real low-rank Gramian factor). 27

5 G-LRCF-ADI iteration (updated). 28

6 LR-SRM for unstable index 2 DAEs. 39

7 GS-LRCF-ADI for unstable index 2 DAEs. 43

8 LR-SRM for second order index 1 systems. 59

9 IRKA for second order index 1 systems. 61

10 SOLR-SRM for second order index 1 system. 63

11 PDEG for second order index 1 system. 65

12 SOGS-LRCF-ADI for the second order index 1 systems. 68

13 LR-SRM for second order index 3 systems. 88

14 IRKA for second order index 3 systems. 90

15 SOLR-SRM for second order index 3 system. 92

16 PDEG for second order index 3 system. 93

17 SOGS-LRCF-ADI for the second order index 3 systems. 97

Notations and Symbols

$\mathbb{R}$	filed of real numbers
$\mathbb{C}$	field of complex numbers
$\mathbb{C}^-$ $(\mathbb{C}^+)$	left (right) half-plane
$\mathbb{R}^{m\times n}$	set of all real matrices of order $m \times n$
$\mathbb{C}^{m\times n}$	set of all real matrices of order $m \times n$
$\in$	belongs to
$\mathrm{Re}\,(\mu)$	real part of $\alpha \in \mathbb{C}$
$\mathrm{Im}\,(\mu)$	imaginary part of $\alpha \in \mathbb{C}$
I_s	identity matrix of order s (default for an $n \times n$ identity matrix in I)
A^T	transpose of A
A^*	complex conjugate transpose of A
A^{-1}	inverse of A
$\subseteq$	subset
$\approx$	approximately equal to
$\ll$ $(\gg)$	much less (greater)
$\mathsf{G}(s)$	transfer function or transfer function matrix
$\Vert\cdot\Vert_{\mathcal{H}_\infty}$	$\mathcal{H}_\infty$-norm
$\sigma_{\text{max}}(A)$	largest singular value of A
Σ	diagonal matrix containing singular values
$\mathrm{diag}\,(d_1, \cdots, d_2)$	diagonal matrix with $d_1, \cdots, d_2$ on the diagonal
$\square$	end of proof
$\Lambda(A)$	spectrum of A
$\lambda_j(A)$	j-th eigenvalue of A

List of Acronyms

ADI	alternating direction implecit
BT	balanced truncation
CARE	continuous-time algebraic Riccati equation
DAEs	differential-algebraic equations
DoF	degrees of freedom
FDM	finite difference method
FEM	finite element method
LRCF-ADI	low-rank Cholesky factor-ADI
LR-SRM	low-rank square root method
LTI	liner time-invariant
LQR	linear-quadratic regulator
G-LRCF-ADI	generalized-LRCF-ADI
GS-LRCF-ADI	generalized sparse-LRCF-ADI
IRKA	iterative rational Krylov algorithm
MOR	model order reduction
MIMO	multi-input multi-output
ODEs	ordinary differential equations
ROM	reduced order model
SISO	single-input single-output
SOGS-LRCF-ADI	second order GS-LRCF-ADI
SOLR-SRM	second order low-rank square root method
spd	symmetric positive definite
SRM	square-root method
SVD	singular value decomposition

Chapter 1

Introduction

1.1 Motivation

Before implementing new ideas or decisions in different disciplines of science, engineering, and technology, an experiment is required. The classical approach of this experiment would require a laboratory with a lot of new equipment, which is an expensive method to demonstrate a concept. The modern approach, rather less expensive and often easier to apply than experiments, to explore scientific ideas to convince others of their validity is through computer simulation. In simulation, one needs to convert a physical model into a mathematical model. Often also in real-life applications the mathematical models are represented by linear time-invariant (LTI) continuous-time systems. In many cases, these systems are subject to additional algebraic constraints, leading to differential-algebraic equations (DAEs) or descriptor systems. These descriptor systems are in either first or second order form. The mathematical models are generated in many different ways. In many applications, the systems are obtained by finite element (FEM) or finite difference (FDM) discretization. In order to model a system accurately, a sufficient number of grid points must be generated because many geometrical details must be resolved. Sometimes physical systems consist of several bodies and each body is composed of a large number of disparate devices. Therefore, the mathematical models become more detailed and different coupling effects must be included. In either case, the resulting systems are typically very *large* and *sparse*. Moreover, often they might be *well-structured*.

A *large*[1] *-scale* system leads to additional memory requirements and enormous computational efforts. They also prevent frequent simulations which is often required in many applications. Sometimes, the generated systems are too large to store due to the restriction of computer memory. To circumvent these complexities reducing the

[1] the notation of large is constantly changing with the increasing capability of the computational hardware.

size of the systems is unavoidable. The method to reduce a higher dimensional to a lower one is called *model order reduction* (MOR). See e.g., [5, 23, 18, 95, 106, 7] for motivations, applications, restrictions, and techniques of MOR.

The fundamental aim of MOR is to replace the high dimensional dynamical systems by substantially lower dimensional systems, while the responses of the original and reduced systems should be approximated to the largest possible extent. In some cases, some important features such as stability, passivity, definiteness, symmetry and so forth of the original system must be preserved in the reduced systems.

The techniques to reduce the state space dimension for a LTI continuous-time ODE system are well established. See e.g., [5, 23, 7, 63] for an overview. In a broad sense, there are two techniques, namely, *Gramian* based methods and *moment matching* based methods. The Gramian based methods include *optimal Hankel norm approximation* [59], *singular perturbation approximation* [52, 83, 27], *dominant subspaces projection* [80, 94], *frequency weighted balanced truncation* [50, 134], *dominant pole algorithm* and *balanced truncation* (BT) [89, 117, 103]. On the other hand, moment matching can be implemented efficiently via *rational Krylov* methods discussed in [49, 51, 54, 129, 56, 6]. The concept of projection for rational interpolation of the transfer function was first proposed in [124]. In [62] Grimme showed how to obtain the required projection using the rational Krylov method of Ruhe [98]. Later on, the authors in [64, 6] generalize Grimme's idea to generate a reduced model which is an optimal $\mathcal{H}_2$ approximation to the original system in the sense that it minimizes the $\mathcal{H}_2$ norm. There the implementing algorithm is called IRKA, i.e., iterative rational Krylov algorithm.

Among all the aforementioned methods, currently balanced truncation (BT) and the interpolatory method via IRKA are the most commonly used techniques for large-scale dynamical systems. In this thesis we also focus on these two prominent methods.

The system theoretic method balanced truncation has an a priori *error bound*. That means for a given system, the method can generate a best approximate system with respect to a given tolerance. Besides this, balanced truncation preserves the stability of the original systems, i.e., if the given system is stable, the method ensures a stable reduced system. Although these two important properties make balanced truncation superior to the other methods, the main disadvantage of this method is to solve two continuous-time algebraic Lyapunov equations for the original model which requires enormous computational resources. On the other hand, the recently developed, interpolatory method via IRKA is attractive to the model reduction community since it is computationally efficient. It requires only matrix-vector products or linear solves. Unfortunately, this prominent method has neither an a priori error bound nor guaranteed stability preservation.

The idea of both BT and IRKA has been extended to large-scale descriptor systems. Another model reduction method for DAE systems was introduced in [2, 3], which is called index-aware model order reduction (IMOR). However, this thesis is not concerned with the IMOR method. The balanced truncation based model reduction technique for DAE systems was first introduced by Stykel in [111]. See, for example [112, 113, 87] for details. The author discusses the general framework of the BT method for a descriptor system. In principle, the proposed method is based on splitting the descriptor system into proper and improper subsystems corresponding to the deflating subspaces of the associated matrix pencil with respect to the finite and infinite eigenvalues, and then reducing only the order of the proper subsystem. To implement the method one requires explicit computation of the spectral projectors onto the deflating subspaces. Although the projectors are available for particular structured systems, they are expensive to compute.

Recently, the BT methods for the large-scale structured (first order) descriptor systems of index 1 and 2 have been developed respectively, in [53] and [70], that avoid the computation of spectral projectors. Instead, they implicitly perform an index reduction by elimination of the algebraic part of the system and conversion into the equivalent form of ODE systems. However, the conversion technique from DAEs to equivalent ODEs, and implementation criterion for index 1 and index 2 systems are separate. The authors in [53] show that for the structured index 1 systems, from the algebraic element of the system, one can find the value of the algebraic variables, and by inserting it into the differential equation and into the output equation, one can get rid of the algebraic element and find an equivalent ODE system. The BT method is then applied to the ODE system. At the end they show that an explicit implementation of the ODE system is not required, but all computations can be performed based on the orginal DAE matrices.

In [70], the author shows that for the structured index 2 DAEs, an index reduction can be performed by projection to the inherent or hidden manifold on which the solution evolves. It is possible to explicitly construct the projector onto the differential element of the system from the given problem data. Finally, by exploiting the properties of the projector, the paper shows that explicit formulation of the projected ODE system is not necessary.

The BT technique discussed in [70] is only applicable for stable systems. In applications such as in the flow control problem, we may obtain structured index 2 unstable descriptor systems. Zhou et. al. [135] discuss an efficient BT technique for an unstable standard system. One of the major contributions of this thesis is the development of a BT algorithms of a class of structured index 2 unstable descriptor systems by combining the results in [70, 135]. For such systems, we also show that a Riccati-based boundary feedback stabilization matrix for the original model (one of the challenging tasks in the flow control problems, see e.g., [12]) can be computed efficiently from the ROM.

Recently, the idea of interpolatory MOR via IRKA was extended to descriptor systems in [68]. There, the theory of the techniques is based on spectral projectors. For a particularly structured (index 1 and index 2) DAE system, the authors show that in the implementation, explicit computation of the projectors is not required.

In many applications in real life, particularly in structural mechanics, multibody dynamics, multiphysics, electric circuits and so forth, the governing mathematical models are in second order form. In many cases the systems are in descriptor form. For the MOR of second order systems, in general, one first converts the systems into first order form. The reduction methods are then applied to the converted system. In this case the structure of the original system is dissolved. And hence one can not go back to the second order form if it is required for the simulation using any software designed for second order systems. Moreover, structure preserving reduced models allow meaningful physical interpretation and provide more accurate approximation, which we will see later. Model reduction of second order systems, including second-order-to-first-order and second-order-to-second-order, has received lot of attention during the recent decades. See, e.g. [76, 11, 104, 105, 16, 96, 29, 66, 20] and references therein for motivations and techniques. All of these articles are devoted to second order standard systems. In this thesis we show the efficient model reduction of structured second order DAEs, which arise in different applications. We discuss both second-order-to-first-order and second-order-to-second order reducing techniques. In the case of second-order-to-first-order reduction, we develop algorithms for the BT and interpolatory methods following the concepts in [70, 68]. For second-order-to-second-order reduction besides balanced truncation, we also investigate the dominant subspaces projection method, which is computationally efficient. Note that this technique originated in [80, 94] for a standard state space system. In this thesis we call this PDEG (projection onto dominant eigenspace of the Gramian) method.

For implementing BT and PDEG based model reduction, the most expensive task is the computation of two Gramian factors by solving two continuous-time algebraic Lyapunov equations. During the recent decades several efficient approaches have been proposed in the literature [46, 93, 81, 108, 22], exploiting the fact that often all coefficient matrices are sparse and the number of inputs and outputs are very small compared to the number of DoFs. The alternating direction implicit (ADI) based method low-rank Cholesky factor (LRCF-)ADI iteration [81, 22] is the most attractive for a large-scale sparse dynamical system. The recent update of this prominent method is available in [19, 20]. This thesis also discusses an updated version of the LRCF-ADI iteration to solve the Lyapunov equations of the structured DAE systems.

A set of ADI shift parameters plays a crucial role in the fast convergence of the LRCF-ADI iteration. Several approaches are proposed in the literature for selecting a set of shift parameters. See for example [21] for an overview of different shift selection approaches. The Penzl's heuristic [93] is one of the most applied approaches for large-scale dynamical systems. For the descriptor systems considered here, computing some large magnitude Ritz values (approximated eigenvalues) is in particular a challenging task. We discuss the issues involved to resolve such problems. Recently another promising shift selection strategy was proposed in [21], in which the LRCF-ADI algorithm automatically updates the shift parameters to ensure fast convergence. There the method is called adaptive approach. We propose a modification of the adaptive shift parameter selection approach.

Note: In this thesis we discuss the model reduction methods of structured descriptor systems. To perform the methods, we transform the descriptor systems into the equivalent form of ODE systems by exploiting the knowledge of the structure of the systems. Therefor, the theories of ODE systems can be applied here. In this case, the important contribution is that we never form the ODE systems explicitly.

1.2 Thesis outline

Chapter 2 is a review of the literature. This chapter contains notations, fundamental concepts, and results from linear algebra, system theory, model reduction and related issues. The concepts of this chapter are used throughout the thesis.

In Chapter 3, we discuss model reduction techniques for unstable index 2 descriptor systems arising from flow control problems. In particular, we consider linearized Navier-Stokes equations. Their spatial discretization by finite elements leads to index-2 DAEs. This causes some technical difficulties for the application of model reduction based on balanced truncation. This chapter shows how to overcome these challenges. To compute the controllability and observability Gramian factors, we need to solve two projected algebraic Lyapunov equations of the Bernoulli stabilized system. We present an algorithm to solve such Lyapunov equations efficiently. To ensure fast convergence of the algorithm we also discuss shift parameter computation approaches. As an illustrative example, we apply our algorithms to the linearization of the *von Kármán vortex shedding* at a moderate Reynolds number. It is demonstrated that the resulting reduced model can be used to accurately simulate the unstable linearized model and to design a stabilizing controller. The balancing based results are compared with that of IRKA.

Chapter 4 focuses on model reduction of a class of second order index 1 systems arising from constraint mechanics, multiphysics, mechatronics, or electrics fields. Particularly, we consider a finite element model of a spindle head configuration in a machine tool. The special feature of this spindle head is that it is partially driven by a set of piezo actuators. Due to this piezo actuation the resulting model is a second order differential algebraic system of index 1. In the first part of this chapter, we show that a suitable first order companion form of the second order system reduces the computational demands in the implementation. Then we focus on second-order-to-first-order reduction methods. In this case we discuss both the BT and IRKA techniques elaborately. Next we discuss the second-order-to-second-order reduction methods using the BT and PDEG methods. We also discuss efficient techniques, including an adaptive shift selection approach, to solve a Lyapunov equation obtained from second order index 1 DAEs. The proposed methods are applied to a structural FEM model of a micro-mechanical piezo-actuators based adaptive spindle support (ASS). Numerical results illustrate the capability of the techniques.

Chapter 5 is about the MOR of second order index 3 descriptor systems. In particular, we consider the models arising from constraint mechanics or multibody dynamics. In the beginning of the chapter we review the index reduction techniques to convert the DAEs to equivalent ODEs. As in Chapter 4, this chapter also first contributes second order to first order reduction. In this case both BT and IRKA are discussed elaborately. Then we include the BT and PDEG methods for structure preserving model reduction techniques of second order index 3 DAEs. This chapter also discusses the LRCF-ADI iteration for solving projected algebraic Lyapunov equations arising from second order index 3 descriptor systems. Computation of ADI shift parameters, the crucial objects of the LRCF-ADI method, is also discussed here for both heuristic and adaptive approaches. The proposed strategies are applied to several test examples and numerical results are discussed to demonstrate the performance.

We summarize our work in Chapter 6, including some important remarks regarding future directions of research.

Chapter 2

Preliminaries

The purpose of this chapter is to establish notation and introduce important concepts or results from the literature. First we discuss some important properties of the linear time-invariant (LTI) continuous-time systems as they are involved in this thesis. Then we briefly introduce the ideas of model reduction and the techniques of model reduction used throughout the thesis. In the implementation of some model reduction methods, the low-rank factors of the system Gramians are the important ingredients. One of the powerful methods for computing the Gramian factors is the LRCF-ADI iteration. The LRCF-ADI method and related issues are discussed to compute the low-rank Gramian factors by solving the Lyapunov equations. We discuss the background theory only for the non-descriptor generalized systems. This is relevant since the model reduction approach for our descriptor systems is applied to the converted ODE systems. This issue is also discussed at the end of the chapter. The profound discussion of a topic or proofs of the theorems, lemmas, etc. are omitted in this chapter since details are available in the referenced literature.

2.1 Theory of systems

This section gives the fundamental properties of the systems and their realizations from the system theory and linear algebra points of view. The generalized state space form of the systems is considered first. The results of the second order and descriptor systems are discussed subsequently.

2.1.1 State space form of dynamical systems

We describe generalized LTI continuous-time systems as

$$\begin{aligned} \mathcal{E}\dot{x}(t) &= \mathcal{A}x(t) + \mathcal{B}u(t); \quad x(t_0) = x_0, \quad t \geq t_0 \\ y(t) &= \mathcal{C}x(t) + \mathcal{D}_a u(t) \end{aligned} \tag{2.1}$$

where $x(t) \in \mathbb{R}^n$ are the states, $u(t) \in \mathbb{R}^m$ are the inputs and, $y(t) \in \mathbb{R}^p$ are the measurement outputs. The matrices $\mathcal{E}$, $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$ and $\mathcal{D}_a$ are of appropriate dimensions. If $m = p = 1$, the system is referred to as a single-input single-output (SISO) system, otherwise it is called a multi-input multi-output (MIMO) system. In the MIMO case, we assume that the number of inputs and outputs are much less than the number of states, i.e., $m, p \ll n$. The dynamical system (2.1) is called *asymptotically stable* if all the finite eigenvalues of the *matrix pencil* [60]

$$\mathcal{P}_c(\lambda) = \lambda\mathcal{E} - \mathcal{A}, \tag{2.2}$$

with $\lambda \in \mathbb{C}$ lie in the left complex plane ($\mathbb{C}^-$). If any eigenvalue of the pencil $\mathcal{P}_c(\lambda)$ lies in $\mathbb{C}^+$ (right complex plane), then the system is called unstable. If $\mathcal{E} = I$, the identity matrix, the system is called a *standard state space system*. For an invertible $\mathcal{E}$, one can convert the generalized state space system (2.1) into standard state space form.

Let the matrix $\mathcal{E}$ be invertible and $\mathcal{A}_s = \mathcal{E}^{-1}\mathcal{A}$, $\mathcal{B}_s = \mathcal{E}^{-1}\mathcal{B}$. Then the solution of the system (2.1) is

$$x(t) = e^{\mathcal{A}_s(t-t_0)}x_0 + \int_{t_0}^{t} e^{\mathcal{A}_s(t-\tau)}\mathcal{B}_s u(\tau)d\tau, \tag{2.3a}$$

$$y(t) = \mathcal{C}e^{\mathcal{A}_s(t-t_0)}x_0 + \int_{t_0}^{t} \mathcal{C}e^{\mathcal{A}_s(t-\tau)}\mathcal{B}_s u(\tau)d\tau + \mathcal{D}_a u(t). \tag{2.3b}$$

Given the initial value x_0 and the input $u(t)$, the behavior of the dynamical system (2.1) can be characterized by $y(t)$ in (2.3), which is called *system response*. In the time domain analysis of the LTI system, the two most commonly used responses are the *step-response* and the *impulse-response*. In a relaxed system (i.e., $x(0) = 0$), the unit step response and the unit impulse response are the respective outputs of the systems when the unit step function and the unit impulse are used. Another important measurement to study the characteristic of the LTI system is the *frequency response*. In order to determine the frequency response, applying the *Laplace transformation*, [1] the system in (2.1) turns out to be

$$s\mathcal{E}X(s) - x_0 = \mathcal{A}X(s) + \mathcal{B}U(s), \tag{2.4a}$$

$$Y(s) = \mathcal{C}X(s) + \mathcal{D}_a U(s), \tag{2.4b}$$

where $X(s)$, $U(s)$ and $Y(s)$ are, respectively, the Laplace transformations of $x(t)$, $u(t)$ and $y(t)$. Considering $x_0 = 0$ and inserting $X(s)$ from (2.4a) into (2.4b) we obtain

$$Y(s) = \mathrm{G}(s)U(s), \tag{2.5}$$

[1] The Laplace transformation of a function $f(t)$, defined for all real numbers $t \geq 0$, is the function $F(s)$, defined by $F(s) = \mathcal{L}[f(t)] = \int_0^\infty f(t)e^{-st}\,dt$. The parameter s is the complex number: $s = a + ib$, with real number a, b.

where

$$\mathrm{G}(s) = \mathcal{C}(s\mathcal{E} - \mathcal{A})^{-1}\mathcal{B} + \mathcal{D}_a, \tag{2.6}$$

known as the *transfer function* (in the SISO case) of the system (2.1). In the case of MIMO systems, $\mathrm{G}(s)$ is the $p \times m$ transfer matrix and can be written as

$$\mathrm{G}(s) = \begin{bmatrix} \mathrm{G}_{11}(s) & \mathrm{G}_{12}(s) & \cdots & \mathrm{G}_{1m}(s) \\ \mathrm{G}_{21}(s) & \mathrm{G}_{22}(s) & \cdots & \mathrm{G}_{2m}(s) \\ \vdots & \vdots & \cdots & \vdots \\ \mathrm{G}_{p1}(s) & \mathrm{G}_{p2}(s) & \cdots & \mathrm{G}_{pm}(s) \end{bmatrix}, \tag{2.7}$$

where $\mathrm{G}_{il} = \mathcal{C}(i,:)(s\mathcal{E}-\mathcal{A})\mathcal{B}(:,l)+\mathcal{D}_a(i,l)$ with $i = 1, 2, \cdots, p$ and $l = 1, 2, \cdots, m$. In fact, the transfer function is the input-output relation of the dynamical system in the complex domain. The behavior of the dynamical systems can be fully characterized by its transfer function.

Definition 2.1. *The transfer function $G(s)$ as defined in (2.5) is called* proper *if* $\lim_{s\to\infty} G(s) < \infty$ *and* strictly proper *if* $\lim_{s\to\infty} G(s) = 0$. *Otherwise $G(s)$ is called* improper.

The frequency response of the dynamical system (2.1) which is defined by

$$\mathrm{G}(\mathrm{j}\omega) = \mathcal{C}(\mathrm{j}\omega\mathcal{E} - \mathcal{A})^{-1}\mathcal{B} + \mathcal{D}_a, \tag{2.8}$$

where $\omega \in \mathbb{R}$ is the frequency, is the value of the transfer function on the imaginary axis.

2.1.2 Controllability and observability of the systems

Controllability and *observability* are basic concepts in control theory; they are useful tools for solving many problems in system theory. The applications of these concepts can be found in [78, 41, 61, 133, 82, 109]. The ideas of controllability and observability of the system also play crucial roles in the MOR methods. The Gramian based MOR methods are in general based on the principle of the system *controllability Gramian* and *observability Gramian*.

Definition 2.2. *The system in (2.1) is said to be controllable in $t_0 \leq t \leq t_f$, if there exists an admissible input $u(t)$ such that the system can be driven from initial state $x(t_0)$ to any final state $x(t_f)$.*

To explain the idea of controllability, at the time $t_0 = 0$, let $x_0 = 0$. Then the relation in (2.3a) yields [109]

$$\begin{aligned} x(t_f) = &\int_0^{t_f} e^{\mathcal{A}_s(t_f-\tau)}\mathcal{B}_s u(\tau)d\tau, \\ = &\int_0^{t_f} \left\{ I + \mathcal{A}_s(t_f-\tau) + \frac{{\mathcal{A}_s}^2}{2!}(t_f-\tau)^2 + \cdots \right\} \mathcal{B}_s u(\tau)d\tau, \\ = &\mathcal{B}_s \int_0^{t_f} u(\tau)d\tau + \mathcal{A}_s\mathcal{B}_s \int_0^{t_f} (t_f-\tau)u(\tau)d\tau + \\ &\mathcal{A}_s^2\mathcal{B}_s \int_0^{t_f} \frac{(t_f-\tau)^2}{2!} u(\tau)d\tau + \cdots . \end{aligned} \tag{2.9}$$

We see in (2.9) that $x(t_f)$ is the linear combination of $\mathcal{B}_s, \mathcal{A}_s\mathcal{B}_s, \cdots, \mathcal{A}_s^{n-1}\mathcal{B}_s$. Therefore, it can be said that a final state $x(t_f)$ is controllable iff the controllability matrix

$$\begin{bmatrix} \mathcal{B}_s & \mathcal{A}_s\mathcal{B}_s & \cdots & \mathcal{A}_s^{n-1}\mathcal{B}_s \end{bmatrix}$$

has full rank. The system (2.1) is said to be controllable if every state of the system is controllable, i.e. the controllability matrix is full [78].

Definition 2.3. *The system in (2.1) is said to be observable in $t_0 \le t \le t_f$, if for a given input $u(t)$ the initial state $x(t_0)$ can be uniquely determined from the given output $y(t)$.*

Observability is the dual concept of the controllability. Analogous to the controllability, one can observe that the system (2.1) is observable if the observability matrix

$$\begin{bmatrix} \mathcal{C} \\ \mathcal{C}\mathcal{A}_s \\ \vdots \\ \mathcal{C}\mathcal{A}_s^{n-1} \end{bmatrix}$$

is nonsingular.

For a controllable, observable, and stable LTI system (2.1) the controllability Gramian and observability Gramian are defined respectively by

$$\mathcal{P} = \int_0^\infty e^{\mathcal{A}_s t}\mathcal{B}_s\mathcal{B}_s^T e^{\mathcal{A}_s^T t}dt \tag{2.10}$$

and

$$\tilde{\mathcal{Q}} = \int_0^\infty e^{\mathcal{A}_s^T t}\mathcal{C}^T\mathcal{C}e^{\mathcal{A}_s t}dt, \tag{2.11}$$

where $\mathcal{A}_s$ and $\mathcal{B}_s$ are defined above. It can be shown that the controllability Gramian ($\mathcal{P}$) as defined in (2.10) is the solution of the continuous-time algebraic Lyapunov equation [55]

$$\mathcal{A}\mathcal{P}\mathcal{E}^T + \mathcal{E}\mathcal{P}\mathcal{A}^T = -\mathcal{B}\mathcal{B}^T, \tag{2.12}$$

which is denoted as the *controllability Lyapunov equation*. Analogously, the observability Gramian defined in (2.11) is the solution of the continuous-time algebraic *observability Lyapunov equation*

$$\mathcal{A}^T\mathcal{Q}\mathcal{E} + \mathcal{E}^T\mathcal{Q}\mathcal{A} = -\mathcal{C}^T\mathcal{C}, \tag{2.13}$$

where $\tilde{\mathcal{Q}} = \mathcal{E}^{-T}\mathcal{Q}\mathcal{E}^{-1}$. If the system (2.1) is asymptotically stable, both of the Lyapunov equations have unique solutions. The following Lemma is important to relatate the stability, controllability, and observability of a system.

Lemma 2.1 ([42]). *For a stable system (2.1), the solutions of the Lyapunov equations (2.12) and (2.13) are unique and symmetric positive definite iff the system is controllable and observable, respectively.*

The controllability and observability Gramians also have an interpretation from a physical point of view [59]. Consider the following two relations

$$J_c = \min_u \int_{-\infty}^{0} u^*(t)u(t)dt, \quad x(0) = x_0,\, t \leq 0, \tag{2.14a}$$

$$J_o = \int_{0}^{-\infty} y^*(t)y(t)dt, \quad u(t) = 0,\, x(0) = x_0,\, t \geq 0, \tag{2.14b}$$

where J_c defines the required minimum energy to drive the system from zero state to the state x_0 and J_o is the obtained energy observed at the output under the zero input and the initial condition x_0. The functionals J_c and J_o can be determined from

$$J_c = x_0^*\mathcal{P}^{-1}x_0 \tag{2.15}$$

and

$$J_o = x_0^*\mathcal{Q}x_0. \tag{2.16}$$

The relation in (2.15) states that any state $x_0 = x(t)$ that lies in an eigenspace of $\mathcal{P}^{-1}$ corresponding to large eigenvalues requires more input energy to control. Since the eigenvectors of $\mathcal{P}^{-1}$ corresponding to large eigenvalues are equal to the eigenvectors of $\mathcal{P}$ with small eigenvalues, it can be said that the state $x_0 = x(t)$ is difficult to control if it lies in an eigenspace of $\mathcal{P}$ corresponding to a small eigenvalue. Likewise, from (2.16) it can be said that the state that lies along one of the eigenvectors of $\mathcal{Q}$ with small eigenvalues is difficult to observe. We can assume that the states that are difficult to control and observe are less important. The balancing based MOR methods are based on identifying and truncating the less important states from the systems. We will revisit these issues later in this chapter.

2.1.3 System Hankel singular values

The system Hankel singular values, or more simply, Hankel singular values (HSVs), play a crucial role in the balancing based model reduction that we will see later. In general, the HSVs of the system are the singular values of the Hankel operator (see e.g., [5]). In [59], Glover shows that the system's HSVs are the positive square roots of the eigenvalues of the product of the controllability and observability Gramians, i.e.,

$$\sigma_i^h = \sqrt{\lambda_i(\mathcal{P}\mathcal{Q})} = \sqrt{\lambda_i(\mathcal{Q}\mathcal{P})}, \quad i = 1, 2, \cdots, n, \tag{2.17}$$

where λ_i denotes the eigenvalues. Since the controllability Gramian and the observability Gramian are symmetric positive definite, they have always Cholesky decomposition:

$$\mathcal{P} = \mathcal{R}_c\mathcal{R}_c^T \quad \text{and} \quad \mathcal{Q} = \mathcal{L}_c\mathcal{L}_c^T. \tag{2.18}$$

It can be shown that (see, e.g., [117, 77])

$$\begin{aligned}\sigma_i^h =& \sqrt{\lambda_i(\mathcal{P}\mathcal{Q})}\\ =& \sqrt{\lambda_i(\mathcal{R}_c\mathcal{R}_c^T\mathcal{L}_c\mathcal{L}_c^T)}\\ =& \sqrt{\lambda_i((\mathcal{R}_c^T\mathcal{L}_c)^T(\mathcal{R}_c^T\mathcal{L}_c))}\\ =& \sigma_i(\mathcal{R}_c^T\mathcal{L}_c), \quad \text{for}\, i = 1, 2, \cdots, n,\end{aligned}$$

where σ_i denotes a singular value of $\mathcal{R}_c^T\mathcal{L}_c$. This means, the HSVs of the systems are the singular values of the product of the two Gramian factors. To compute the system's HSVs in practice, we therefore use the Gramian factors of the systems.

2.1.4 Realizations

The transfer function of an LTI system is invariant under *state space transformations* or *coordinate transformations* [17]. For instance, if we replace $x(t)$ in (2.1) with

$$\tilde{x}(t) = \mathcal{T}x(t), \tag{2.19}$$

where the nonsingular matrix $\mathcal{T}$ is a coordinate transformation [17], we obtain a transformed system in which

$$(\mathcal{E}, \mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}_a) \Leftrightarrow (\mathcal{T}\mathcal{E}\mathcal{T}^{-1}, \mathcal{T}\mathcal{A}\mathcal{T}^{-1}, \mathcal{T}\mathcal{B}, \mathcal{C}\mathcal{T}^{-1}, \mathcal{D}_a). \tag{2.20}$$

The invariance of the transfer function under coordinate transformations of the system can be shown by

$$\begin{aligned}\tilde{\mathrm{G}}(s) =& (\mathcal{C}\mathcal{T}^{-1})(s\mathcal{T}\mathcal{E}\mathcal{T}^{-1} - \mathcal{T}\mathcal{A}\mathcal{T}^{-1})^{-1}(\mathcal{T}\mathcal{B}) + \mathcal{D}_a\\ =& \mathcal{C}(s\mathcal{E} - \mathcal{A})^{-1}B + \mathcal{D}_a = \mathrm{G}(s).\end{aligned}$$

Here we see that $(\mathcal{E},\mathcal{A},\mathcal{B},\mathcal{C},\mathcal{D}_a)$ and $(\mathcal{T}\mathcal{E}\mathcal{T}^{-1},\mathcal{T}\mathcal{A}\mathcal{T}^{-1},\mathcal{T}\mathcal{B},\mathcal{C}\mathcal{T}^{-1},\mathcal{D}_a)$ are essentially two different realizations of the same transfer function $\mathrm{G}(s)$. Since the input/output relations of a system is not changing under coordinate transformations, a system may have infinitely many realizations. Among them there exist realizations where the dimension (r) of the system is minimum or the system consists of minimum number of degree of freedoms (DoF). This number r is called the *McMillan degree* of the system.

Definition 2.4. *A realization $(\mathcal{E}_r,\mathcal{A}_r,\mathcal{B}_r,\mathcal{C}_r,\mathcal{D}_a)$ of the transfer function $G(s)$ in (2.5) of McMillan degree r is called a minimal realization.*

A state space realization of a transfer function $\mathrm{G}(s)$ is minimal iff the system is controllable and observable. Note that although the McMillan degree is unique, the coordinate transformations leads to many minimum realizations of the same system. Fundamentally, the concept of MOR is to find a realization of a given system where the dimension of the system is as small as possible. We will study this in the coming section.

2.1.5 Second order systems

We consider second order LTI continuous-time systems

$$\begin{aligned} \mathcal{M}\ddot{\xi}(t)+\mathcal{D}\dot{\xi}(t)+\mathcal{K}\xi(t)&=\mathcal{H}u(t),\\ y(t)&=\mathcal{L}_1\xi(t)+\mathcal{L}_2\dot{\xi}(t)+\mathcal{D}_a u(t), \end{aligned} \tag{2.21}$$

where $\mathcal{M},\mathcal{D},\mathcal{K}\in\mathbb{R}^{n_\xi\times n_\xi}$, input matrix $\mathcal{H}\in\mathbb{R}^{n_\xi\times p}$ and output matrices $\mathcal{L}_1,\mathcal{L}_2\in\mathbb{R}^{m\times n_\xi}$. Such systems usually appear in mechanics [8] or structural and multibody dynamics [48, 40], where the velocity is taken into account in the modeling, and thus the acceleration becomes part of the system. In mechanics, usually, the matrices $\mathcal{M}$, $\mathcal{D}$, and $\mathcal{K}$ are known as mass, damping and stiffness matrices, respectively, and the vector $\xi(t)$ is known as mechanical displacement. Such systems also appear in electrical engineering when RLC circuits are designed for nodal analysis [131]. There the matrices $\mathcal{M}$, $\mathcal{D}$, and $\mathcal{K}$ are called the conductance, capacitance and susceptance matrices, respectively, and the vector $\xi(t)$ is denoted as electric charge. However, the second order form of the system (2.21) can be converted into the first order form (2.1). In the literature [116] several transformations are shown to convert the second order system into the first order from which can all be proved to be equivalent. The most common transformation using $z(t) := \begin{bmatrix} \xi(t) \\ \dot{\xi}(t) \end{bmatrix}$ is

$$\begin{aligned} \underbrace{\begin{bmatrix} \mathcal{F} & 0 \\ 0 & \mathcal{M} \end{bmatrix}}_{E} \underbrace{\begin{bmatrix} \dot{\xi}(t) \\ \ddot{\xi}(t) \end{bmatrix}}_{\dot{z}(t)} &= \underbrace{\begin{bmatrix} 0 & \mathcal{F} \\ -\mathcal{K} & -\mathcal{D} \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} \xi(t) \\ \dot{\xi}(t) \end{bmatrix}}_{z(t)} + \underbrace{\begin{bmatrix} 0 \\ \mathcal{H} \end{bmatrix}}_{B} u(t), \\ y(t) &= \underbrace{\begin{bmatrix} \mathcal{L}_1 & \mathcal{L}_2 \end{bmatrix}}_{C} \begin{bmatrix} \xi(t) \\ \dot{\xi}(t) \end{bmatrix} + \mathcal{D}_a u(t), \end{aligned} \tag{2.22}$$

where $\mathcal{F}$ is any nonsingular matrix with appropriate size. For simplicity, one can consider $\mathcal{F} = I$. If the matrices $\mathcal{M}$, $\mathcal{D}$ and $\mathcal{K}$ are all symmetric, perhaps one of the suitable first order representations of the system (2.21) can be

$$\begin{aligned} \underbrace{\begin{bmatrix} 0 & \mathcal{F} \\ \mathcal{M} & \mathcal{D} \end{bmatrix}}_{E} \underbrace{\begin{bmatrix} \ddot{\xi}(t) \\ \dot{\xi}(t) \end{bmatrix}}_{\dot{z}(t)} &= \underbrace{\begin{bmatrix} \mathcal{F} & 0 \\ 0 & -\mathcal{K} \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} \dot{\xi}(t) \\ \xi(t) \end{bmatrix}}_{z(t)} + \underbrace{\begin{bmatrix} 0 \\ \mathcal{H} \end{bmatrix}}_{B} u(t), \\ y(t) &= \underbrace{\begin{bmatrix} \mathcal{L}_2 & \mathcal{L}_1 \end{bmatrix}}_{C} \begin{bmatrix} \dot{\xi}(t) \\ \xi(t) \end{bmatrix} + \mathcal{D}_a u(t). \end{aligned} \tag{2.23}$$

In this formulation the system matrices E and A become symmetric if $\mathcal{F} = \mathcal{M}$. We can exploit this property in practical implementations. For both representations in (2.22) and (2.23) of the second order system (2.21), one can show that

$$C(sE - A)^{-1}B = (\mathcal{L}_1 + \mathcal{L}_2 s)(\mathcal{M}s^2 + \mathcal{D}s + \mathcal{K})^{-1}\mathcal{H}, \tag{2.24}$$

where $s \in \mathbb{C}$. Therefore, the transfer function for the second order model (2.21) can directly be defined from (2.8) as

$$\mathrm{G}_s(s) = (\mathcal{L}_1 + \mathcal{L}_2 s)(\mathcal{M}s^2 + \mathcal{D}s + \mathcal{K})^{-1}\mathcal{H} + \mathcal{D}_a. \tag{2.25}$$

The Gramians for the second order systems can be defined from an energy interpretation perspective as mentioned above for the first order systems. Let us consider that P is the controllability Gramian of the system (2.22). By defining $J(u) = \int_{-\infty}^{0} u^*(t)u(t)dt$, it can be shown that

$$z_0^* P^{-1} z_0 \tag{2.26}$$

is the solution of the problem

$$\begin{aligned} &\min_{u} \ J(u) \\ \text{s. t.} \quad &E\dot{z}(t) = Az(t) + Bu(t), \quad z(0) = z_0, \end{aligned} \tag{2.27}$$

with $z_0 = \begin{bmatrix} \xi_0 \\ \dot{\xi}_0 \end{bmatrix}$. Equation (2.26) in fact represents the required minimal energy to reach to the state z_0 from $t = -\infty$ at time $t = 0$. Now consider the two optimization problems

$$\begin{aligned} &\min_{\xi_0} \min_{u} \ J(u) \\ \text{s. t.} \quad &\mathcal{M}\ddot{\xi}(t) + \mathcal{D}\dot{\xi}(t) + \mathcal{K}\xi(t) = \mathcal{H}u(t), \quad \xi(0) = \xi_0, \end{aligned} \tag{2.28}$$

and

$$\begin{aligned} &\min_{\dot{\xi}_0} \min_{u} \ J(u) \\ \text{s. t.} \quad &\mathcal{M}\ddot{\xi}(t) + \mathcal{D}\dot{\xi}(t) + \mathcal{K}\xi(t) = \mathcal{H}u(t), \quad \dot{\xi}(0) = \dot{\xi}_0. \end{aligned} \tag{2.29}$$

Due to the structure, the controllability Gramian P of the system (2.22) can be compatibly partitioned as

$$P = \begin{bmatrix} P_p & P_o \\ P_o^T & P_v \end{bmatrix}.$$

The authors in [88] (see also [38]), show the optimal solution to the problem (2.28) is $\xi_0 P_p^{-1} \xi_0$, which is the minimal energy required to reach the given position ξ_0 over all past inputs and initial values. And the problem (2.29) is $\dot{\xi}_0 P_v^{-1} \dot{\xi}_0$, which is the minimal energy required to reach the given velocity $\dot{\xi}_0$ over all past inputs and initial values. Therefore, P_p and P_v are called the second order controllability position Gramian and the velocity Gramian, respectively. Similarly, partitioning the observability Gramian Q of the the systems (2.22) as

$$Q = \begin{bmatrix} Q_p & Q_o \\ Q_o^T & Q_v \end{bmatrix},$$

we can denote Q_p and Q_v as the second order observability velocity and position Gramians, respectively.

2.2 Model reduction

The aim of model reduction is to replace the system (2.1) by a substantially lower dimensional system

$$\begin{aligned} \hat{\mathcal{E}}\dot{\hat{x}}(t) &= \hat{\mathcal{A}}\hat{x}(t) + \hat{\mathcal{B}}u(t), \\ \hat{y}(t) &= \hat{\mathcal{C}}\hat{x}(t) + \hat{\mathcal{D}}_a u(t), \end{aligned} \tag{2.30}$$

where $\hat{\mathcal{E}}$, $\hat{\mathcal{A}} \in \mathbb{R}^{r \times r}$, $\hat{\mathcal{B}} \in \mathbb{R}^{r \times m}$, $\hat{\mathcal{C}} \in \mathbb{R}^{m \times p}$, $\hat{\mathcal{D}}_a := \mathcal{D}_a$. Here the goal is to ensure that the approximation error $\|y - \hat{y}\|$, ($\|.\|$ denotes a suitable norm) must be sufficiently small. Analogous to (2.5), applying the Laplace transformations to the system (2.30) we get

$$\hat{Y}(s) = \hat{\mathrm{G}}(s)U(s), \tag{2.31}$$

where

$$\hat{\mathrm{G}}(s) = \hat{\mathcal{C}}(s\hat{\mathcal{E}} - \hat{\mathcal{A}})^{-1}\hat{\mathcal{B}} + \hat{\mathcal{D}}_a \tag{2.32}$$

is the transfer function for the reduced model. We know that

$$\|Y - \hat{Y}\|_{\mathcal{L}_2} = \|\mathrm{G}U - \hat{\mathrm{G}}U\|_{\mathcal{L}_2} \leq \|\mathrm{G} - \hat{\mathrm{G}}\|_{\mathcal{H}_\infty}\|U\|_{\mathcal{L}_2}, \tag{2.33}$$

where $\|.\|_{\mathcal{H}_2}$ and $\|.\|_{\mathcal{H}_\infty}$ are respectively, the $\mathcal{H}_2$ norm and $\mathcal{H}_\infty$ norm of a complex matrix-valued function.

Definition 2.5. *For a stable (SISO) system (2.1) the $\mathcal{H}_2$ norm is defined by*

$$\|G\|_{\mathcal{H}_2} = \sqrt{\left(\frac{1}{2\pi}\int_{-\infty}^{\infty} |G(j\omega)|^2 d\omega\right)}. \tag{2.34}$$

Lemma 2.2. *If $\mathcal{P}$ and $\mathcal{Q}$ are the solutions of the controllability and observability Lyapunov equations defined in (2.12) and (2.13), then the $\mathcal{H}_2$ norm can be computed from*

$$\|G\|_{\mathcal{H}_2} = \sqrt{\mathcal{B}^T\mathcal{Q}\mathcal{B}} = \sqrt{\mathcal{C}\mathcal{P}\mathcal{C}^T}. \tag{2.35}$$

Definition 2.6. *The $\mathcal{H}_\infty$ norm of the stable system (2.1) is defined by*

$$\|G\|_{\mathcal{H}_\infty} = \sup_{\omega\in\mathbb{R}} \sigma_{max}(G(j\omega)), \tag{2.36}$$

where σ_{max} denotes the maximum singular value of $G(j\omega)$.

From (2.33) it is clear that, in the frequency domain, for the same input, the difference between two output responses can be bounded by $\|\mathrm{G} - \hat{\mathrm{G}}\|_{\mathcal{H}_\infty}$. By minimizing $\|\mathrm{G} - \hat{\mathrm{G}}\|_{\mathcal{H}_\infty}$ we can guarantee that $\|Y - \hat{Y}\|_{\mathcal{H}_2}$ is minimized. Hence in model reduction, the approximation error between original and reduced model can be shown by computing the $\mathcal{H}_\infty$ norm of the difference of two transfer functions, i.e., $\|\mathrm{G} - \hat{\mathrm{G}}\|_{\mathcal{H}_\infty}$ in a certain range of the frequency domain. In most cases, the MOR methods for a dynamical system are performed by projecting the system onto a lower dimensional subspace.

Definition 2.7. *A matrix $\Pi \in \mathbb{R}^{n\times n}$ which satisfies $\Pi^2 = \Pi$ is called projector or projection matrix. If $\mathcal{S}_1 = \mathrm{Range}\,(\Pi)$, then Π is the projector onto $\mathcal{S}_1$. Let $V = [v_1, \cdots, v_r]$, and $\mathcal{S}_1 = \mathrm{Range}\,(V)$, then $\Pi = V(V^TV)V^T$ is the projector onto $\mathcal{S}_1$.*

Lemma 2.3. *Suppose Π is a projector. The following are then true for Π:*

1. *The matrix $I - \Pi$ is also a projector, called complementary projector.*
2. *The projector Π is orthogonal if $\Pi = \Pi^T$, otherwise it is an oblique projector.*
3. *Let $\mathcal{S}_2$ be another r dimensional subspace and $\mathcal{S}_2 = \mathrm{Range}\,(W)$, where $W = [w_1, \cdots, w_r]$, then $\Pi = V(W^TV)^{-1}W^T$ is called an oblique projector.*

Recalling the system (2.1), let us assume that the state vector $x(t)$ is contained in a lower dimensional subspace S_1. Thus, we can project $x(t)$ onto $\mathcal{S}_1$ along $\mathcal{S}_2$ by applying an orthogonal or an oblique projector. To achieve this goal, construct $V = [v_1, \cdots, v_r]$ and $W = [w_1, \cdots, w_r]$ such that

$$V = \text{Range}\,(\mathcal{S}_1) \quad \text{and} \quad W = \text{Range}\,(\mathcal{S}_2)\,. \tag{2.37}$$

Now approximating $x(t)$ by $VW^T x(t)$ in (2.1) and defining $\hat{x}(t) = W^T x(t)$ we obtain

$$\mathcal{E}V\dot{\hat{x}}(t) \approx \mathcal{A}V\hat{x}(t) + \mathcal{B}u(t), \tag{2.38a}$$

$$y(t) \approx \mathcal{C}V\hat{x}(t) + \mathcal{D}_a u(t). \tag{2.38b}$$

Since there exists an error $e = \mathcal{E}V\dot{\hat{x}}(t) - \mathcal{A}Vx(t) - \mathcal{B}u(t)$ in the state equation we write " $\approx$ " instead of " $=$ ". This error is called residual. By construction, each column of W is perpendicular to e, i.e., $W^T e = 0$. Thus, (2.38) becomes

$$\begin{aligned} W^T\mathcal{E}V\dot{\hat{x}}(t) &= W^T\mathcal{A}V\hat{x}(t) + W^T\mathcal{B}u(t), \\ \hat{y}(t) &= \mathcal{C}V\hat{x}(t) + \mathcal{D}_a u(t), \end{aligned} \tag{2.39}$$

which is exactly the reduced model as in (2.30) with

$$\hat{\mathcal{E}} = W^T\mathcal{E}V \quad \hat{\mathcal{A}} = W^T\mathcal{A}V, \quad \hat{\mathcal{B}} = W^T\mathcal{B} \;\text{ and }\; \hat{\mathcal{C}} = \mathcal{C}V. \tag{2.40}$$

At a glance, in the projection based model reduction methods to compute the reduced models (2.30), one needs to compute the reduced coefficient matrices (2.40) by applying thin rectangular matrices V and W which are called the right and left transformation matrices, respectively. In this method the basic task is to construct the transformations by using the bases vectors of the subspaces $\mathcal{S}_1$ and $\mathcal{S}_2$. However, the choice of the basis for $\mathcal{S}_1$ and $\mathcal{S}_2$ is not unique. Therefore, different types of model reduction methods are available in the literature based on the different choices of the basis for these subspaces. In the following, we discuss some prominent MOR methods which compute the transformation matrices V and W in different ways.

2.2.1 Balanced truncation

A good motivation of balanced truncation can be found in [5]. The fundamental idea of balanced truncation is to truncate the *less-important states* from the systems. A less-important state is a state that is difficult to control and observe. Those states essentially correspond to the smallest HSVs. In reality, the states which are difficult to control may not be difficult to observe and vice versa. This implicates if we eliminate the states that are hard to be controlled directly from the original system, then we may also eliminate some states that are easy to observe. However, in an application, the easily observable states are essential to be preserved. The same contradiction might appear for those states that are difficult to be observed but easily controlled. This problem can be resolved by transforming the system into a balanced form. In a balanced, system the degree of controllability and the degree of observability of each state are the same. A balanced system can also be defined as follows.

Definition 2.8. *A stable and minimal LTI system is called balanced if the controllability Gramian and the observability Gramian of the system are equal and diagonal. The diagonal elements are the system's HSVs.*

Now if we eliminate those states of the balanced system that are hard to be controlled, we have eliminated the hard to observe states at the same time. The systems can be balanced via a *balancing transformation*.

Definition 2.9. *A state space transformation $\mathcal{T}$ as defined in (2.19) is called balancing transformation if it causes*

$$\mathcal{T}\mathcal{P}\mathcal{T}^{*} = \mathcal{T}^{-*}\mathcal{Q}\mathcal{T}^{-1} = \begin{bmatrix} \Sigma_1 & \\ & \Sigma_2 \end{bmatrix}, \tag{2.41}$$

where $\mathcal{P}$ and $\mathcal{Q}$ are the controllability and observability Gramians, $\Sigma_1 = \operatorname{diag}(\sigma_1, \cdots, \sigma_r)$, $\Sigma_2 = \operatorname{diag}(\sigma_{r+1}, \cdots, \sigma_n)$, and $\{\sigma_i\}_{i=1}^{n}$ are the system's HSVs.

Under the balancing transformations, according to the Gramians in (2.41), the state space realization is transformed into

$$\begin{aligned}(\mathcal{E},\ \mathcal{A},\ \mathcal{B},\ \mathcal{C},\ \mathcal{D}_a) &\mapsto (T\mathcal{E}T^{-1},\ T\mathcal{A}T^{-1},\ T\mathcal{B},\ \mathcal{C}T^{-1},\ \mathcal{D}_a)\\ &= \left(\begin{bmatrix} \mathcal{E}_{11} & \mathcal{E}_{12} \\ \mathcal{E}_{21} & \mathcal{E}_{22} \end{bmatrix}, \begin{bmatrix} \mathcal{A}_{11} & \mathcal{A}_{12} \\ \mathcal{A}_{21} & \mathcal{A}_{22} \end{bmatrix}, \begin{bmatrix} \mathcal{B}_1 \\ \mathcal{B}_2 \end{bmatrix}, \begin{bmatrix} \mathcal{C}_1 & \mathcal{C}_2 \end{bmatrix}, \mathcal{D}_a\right).\end{aligned}$$

Now picking up the block matrices $\mathcal{E}_{11}, \mathcal{A}_{11}, \mathcal{B}_1, \mathcal{C}_1$ one can form the ROM (2.30), where $(\hat{\mathcal{E}}, \hat{\mathcal{A}}, \hat{\mathcal{B}}, \hat{\mathcal{C}}) = (\mathcal{E}_{11}, \mathcal{A}_{11}, \mathcal{B}_1, \mathcal{C}_1)$.

From the above discussion we can conclude that in the balancing based model reduction one must first compute the balancing transformation ($\mathcal{T}$) to convert the system into a balanced form. Then the truncation is performed on the balanced system. For a large-scale system, balancing the whole system before truncation is infeasible. Hence, for such systems, usually the balancing and truncation are carried out simultaneously, by using the so-called *balancing and truncating* transformations.

The author of [5] review, several approaches to compute the balancing and truncating transformations, among which we will focus on the *square-root method* (SRM), originally defined in [117]. To perform this method, compute the Gramian factors $\mathcal{R}_c$ and $\mathcal{L}_c$ as defined in (2.18). Then the balancing transformation can be formed using the SVD

$$\mathcal{R}_c^T \mathcal{E} \mathcal{L}_c = \mathcal{U}\Sigma\mathcal{V}^T = \begin{bmatrix} \mathcal{U}_1 & \mathcal{U}_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & \\ & \Sigma_2 \end{bmatrix} \begin{bmatrix} \mathcal{V}_1^T \\ \mathcal{V}_2^T \end{bmatrix},$$

Algorithm 1: LR-SRM.

```
Input : $\mathcal{E}$, $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, $\mathcal{D}_a$.
Output: $\hat{\mathcal{E}}$, $\hat{\mathcal{A}}$, $\hat{\mathcal{B}}$, $\hat{\mathcal{C}}$, $\hat{\mathcal{D}}_a := \mathcal{D}_a$.
Compute $\mathcal{R}$ and $\mathcal{L}$ as defined in (2.18) by solving (2.10) and (2.11).
Compute SVD $\mathcal{R}^T\mathcal{E}\mathcal{L} = \mathcal{U}\Sigma\mathcal{V}^T = \begin{bmatrix}\mathcal{U}_1 & \mathcal{U}_2\end{bmatrix}\begin{bmatrix}\Sigma_1 & \\ & \Sigma_2\end{bmatrix}\begin{bmatrix}\mathcal{V}_1^T \\ \mathcal{V}_2^T\end{bmatrix}$.
Construct $V := \mathcal{L}\mathcal{V}_1\Sigma_1^{-\frac{1}{2}}, \quad W := \mathcal{R}\mathcal{U}_1\Sigma_1^{-\frac{1}{2}}$.
Form $\hat{\mathcal{E}} = W^T\mathcal{E}V, \quad \hat{\mathcal{A}} = W^T\mathcal{A}V, \quad \hat{\mathcal{B}} = W^T\mathcal{B}$ and $\hat{\mathcal{C}} = \mathcal{C}V$.
```

and defining

$$V := \mathcal{L}_c\mathcal{V}_1\Sigma_1^{-\frac{1}{2}}, \quad W := \mathcal{R}_c\mathcal{U}_1\Sigma_1^{-\frac{1}{2}}, \tag{2.42}$$

where $\mathcal{U}_1$ and $\mathcal{V}_1$ are composed of the leading k columns of $\mathcal{U}$ and $\mathcal{V}$, respectively, Σ_1 is the first $k \times k$ block of the matrix $\Sigma = \operatorname{diag}(\sigma_1, \sigma_2, \ldots, \sigma_k, \ldots, \sigma_n)$. Finally, by applying the balancing transformations (2.42) to the system (2.1), one can derive the ROM (2.30).

The Gramian factors $\mathcal{R}_c$ and $\mathcal{L}_c$ are obtained by using the Cholesky decompositions of the Gramians $\mathcal{P}$ and $\mathcal{Q}$. The Gramians can be computed by solving the corresponding Lyapunov equations. There exist direct solvers [14, 69] as well as iterative solvers [126, 72, 25] to compute $\mathcal{P}$ and $\mathcal{Q}$ by solving the Lyapunov equations (2.12-2.13). All these methods are applicable for a small dense system. If the number of inputs and outputs are much smaller than the dimension of the system, then the Gramians $\mathcal{P}$ and $\mathcal{Q}$ can usually be approximated by low-rank factors, i.e,

$$\mathcal{P} \approx \mathcal{R}\mathcal{R}^T \quad \text{and} \quad \mathcal{Q} \approx \mathcal{L}\mathcal{L}^T. \tag{2.43}$$

Here $\mathcal{R}$ and $\mathcal{L}$ are thin rectangular matrices. Therefore, instead of computing the full Gramian factors, one can compute low-rank factors of the Gramians. During the last few decades, several iterative methods were proposed, e.g., LRCF-ADI (low-rank Cholesky factor - alternating direction implicit) iterations [81, 22], cyclic low-rank Smith methods [93, 67], projection methods [99, 46, 72, 73, 108], and sign function methods [26, 28, 15]. Although most of the methods are shown to be applicable for large scale dynamical systems, the LRCF-ADI iteration is more attractive in the context of Gramian based model reduction for large sparse systems with few inputs and outputs. A motivation of this prominent method can be found in [30]. The next section contributes the LRCF-ADI iteration and related issues for solving the large sparse continuous-time algebraic Lyapunov equations.

Using the low-rank Gramian factors $\mathcal{R}$ and $\mathcal{L}$, the square root method is summarized in Algorithm 1.

The reduced systems obtained by balanced truncation satisfy [5, 59] the global error bound

$$\|\mathrm{G} - \hat{\mathrm{G}}\|_{\mathcal{H}_\infty} \leq 2 \sum_{i=k+1}^{n} \sigma_i, \tag{2.44}$$

where $\hat{\mathrm{G}}$ is the transfer function of the reduced model. The relation (2.44) is an *a priori* error bound. Thus, for a given error bound (tolerance) one can use it to fix the required dimension of the reduced system.

2.2.2 Interpolatory projections

Interpolatory projection methods seek a ROM (2.30) by constructing the matrices V and W in such way that the reduced transfer function (2.32) interpolates the original transfer function (2.6) at a predefined set of interpolation points. That is find $\hat{\mathrm{G}}(s)$ such that

$$\begin{aligned} \mathrm{G}(\alpha_i) &= \hat{\mathrm{G}}(\alpha_i), \\ \mathcal{C}(\alpha_i\mathcal{E} - \mathcal{A})^{-1}\mathcal{B} &= \hat{\mathcal{C}}(\alpha_i\hat{\mathcal{E}} - \hat{\mathcal{A}})^{-1}\hat{\mathcal{B}}, \qquad \text{for } i = 1, \cdots, r, \end{aligned} \tag{2.45}$$

where $\alpha_i \in \mathbb{C}$ are the interpolation points. Often, in addition to the above conditions, we are interested in matching more quantities, that is

$$\begin{aligned} \mathrm{G}^{(j)}(\alpha_i) &= \hat{\mathrm{G}}^{(j)}(\alpha_i), \\ \mathcal{C}[(\alpha_i\mathcal{E} - \mathcal{A})^{-1}\mathcal{E}]^j(\alpha_i\mathcal{E} - \mathcal{A})^{-1}\mathcal{B} &= \hat{\mathcal{C}}[(\alpha_i\hat{\mathcal{E}} - \hat{\mathcal{A}})^{-1}\hat{\mathcal{E}}]^j(\alpha_i\hat{\mathcal{E}} - \hat{\mathcal{A}})^{-1}\hat{\mathcal{B}}, \end{aligned} \tag{2.46}$$

for $j = 0, 1, \cdots, q$, where $\mathcal{C}[-(\alpha_i\mathcal{E} - \mathcal{A})^{-1}\mathcal{E}]^j(\alpha_i\mathcal{E} - \mathcal{A})^{-1}\mathcal{B}$ is called the j-th moment of $\mathrm{G}(s)$ at α_i, and represents the j-th derivative of $\mathrm{G}(s)$ evaluated at σ_i. Note that for $j = 0$, these conditions reduce to (2.45). In this thesis, we restrict ourselves to simple Hermite interpolation, where $j = 0$ and $j = 1$. In the following, we discuss how projection can ensure a reduced interpolating approximation by carefully selecting the matrices V and W.

The concept of projection for interpolatory model reduction was initially introduced in [124], later, Grimme in [62] modified the approach by utilizing the rational Krylov method [98]. Since Krylov based methods can achieve moment matching without explicitly computing moments (explicit computation of moments is known to be ill-conditioned [51]), they are extremely useful for model reduction of large scale systems.

The following result suggests a choice of V and W that ensure Hermite interpolation with the use of a rational Krylov subspace.

Lemma 2.4 ([64])**.** *Consider two sets of distinct interpolation points,* $\{\alpha_i\}_{i=1}^r \subset \mathbb{C}$ *and* $\{\beta_i\}_{i=1}^r \subset \mathbb{C}$*, which are closed under conjugation (i.e., the points are either real* *or appear in conjugate pairs). Suppose V and W satisfy*

$$\operatorname{Range}(V) = \operatorname{span}\left\{(\alpha_1\mathcal{E}-\mathcal{A})^{-1}\mathcal{B}, \cdots, (\alpha_r\mathcal{E}-\mathcal{A})^{-1}\mathcal{B}\right\}, \tag{2.47a}$$

$$\operatorname{Range}(W) = \operatorname{span}\left\{(\beta_1\mathcal{E}^T-\mathcal{A}^T)^{-1}\mathcal{C}^T, \cdots, (\beta_r\mathcal{E}^T-\mathcal{A}^T)^{-1}\mathcal{C}^T\right\}. \tag{2.47b}$$

Then V and W can be chosen real and $\hat{G}(s) = \hat{\mathcal{C}}(s\hat{\mathcal{E}}-\hat{\mathcal{A}})^{-1}\hat{\mathcal{C}}$, where $\hat{\mathcal{E}}$, $\hat{\mathcal{A}}$, $\hat{\mathcal{B}}$ and $\hat{\mathcal{C}}$ define the ROM (2.30). The ROM satisfies the interpolation conditions

$$\begin{aligned} G(\alpha_i) = \hat{G}(\alpha_i), &\quad G(\beta_i) = \hat{G}(\beta_i), \quad \textit{and} \\ G'(\alpha_i) = \hat{G}'(\alpha_i) &\quad \textit{when } \alpha_i = \beta_i, \end{aligned}$$

for $i = 1, \cdots, r$.

The subspace in (2.47a), that is, the span of the column vectors $(\alpha_i\mathcal{E} - A)^{-1}\mathcal{B}$ for $i = 1, \cdots, r$, can be considered as the union of shifted rational Krylov subspaces. For a given shift frequency $\alpha \in \mathbb{C}$, the rational Krylov subspace $\mathcal{K}_q((\alpha\mathcal{E}-\mathcal{A})^{-1}, (\alpha\mathcal{E}-\mathcal{A})^{-1}\mathcal{B})$ is defined as

$$\mathcal{K}_q((\alpha\mathcal{E}-\mathcal{A})^{-1}, (\alpha\mathcal{E}-\mathcal{A})^{-1}\mathcal{B}) := \operatorname{span}\left\{(\alpha\mathcal{E}-\mathcal{A})^{-1}\mathcal{B}, \cdots, (\alpha\mathcal{E}-\mathcal{A})^{-q}\mathcal{B}\right\}.$$

If $q = 1$ for each $\alpha_i, i = 1, \cdots, r$, then the union of such shifted rational Krylov subspaces is equivalent to the subspace in (2.47a). Analogously, the subspace in (2.47b) can also be defined as the union of shifted rational Krylov subspaces given above. To summarize, rational Krylov based model reduction requires a suitable choice of interpolation points, the construction of V and W as in Lemma 2.4, and the use of Petrov-Galerkin conditions.

The quality of the reduced model is highly dependent on the choice of interpolation points and therefore various techniques [124] have been developed for the selection of interpolation points. Recently in [64], the issue of selecting a good choice of interpolation points is linked to the problem of $\mathcal{H}_2$-optimal model reduction.

Definition 2.10. *A ROM (2.30) is called $\mathcal{H}_2$ optimal if it satisfies*

$$\|G\|_{\mathcal{H}_2} = \min_{\dim(\hat{G})=r} \|G-\hat{G}\|_{\mathcal{H}_2}. \tag{2.48}$$

IRKA is proposed in [64]. upon convergence, it identifies a choice of interpolation points that guarantees the $\mathcal{H}_2$-optimality conditions for the reduced system. Starting from an initial set of interpolation points, the IRKA iteration updates the interpolation points until they converge to a fixed value. Until now we have considered that (2.1) is a SISO system. A complete procedure of IRKA for a SISO system is given in [64, Algorithm 4.1].

For model reduction of MIMO dynamical systems, rational tangential interpolation has been developed by Gallivan et. al. [57]. The problem of rational tangential

Algorithm 2: IRKA for MIMO systems.

```
Input  : $\mathcal{E}, \mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}_a$.
Output: $\hat{\mathcal{E}},\ \hat{\mathcal{A}},\ \hat{\mathcal{B}},\ \hat{\mathcal{C}},\ \hat{\mathcal{D}}_a := \mathcal{D}_a$.
Make an initial selection of the interpolation points $\{\alpha_i\}_{i=1}^r$ and the
tangential directions $\{b_i\}_{i=1}^r$ and $\{c_i\}_{i=1}^r$.
$V = \Big[(\alpha_1\mathcal{E} - \mathcal{A})^{-1}\mathcal{B}b_1, \cdots, (\alpha_r\mathcal{E} - \mathcal{A})^{-1}\mathcal{B}b_r\Big]$,
$W = \Big[(\alpha_1\mathcal{E} - \mathcal{A})^{-1}\mathcal{C}^T c_1, \cdots, (\alpha_r\mathcal{E} - \mathcal{A})^{-1}\mathcal{C}c_r\Big]$.
while (not converged) do
    $\hat{\mathcal{E}} = W^T\mathcal{E}V$, $\hat{\mathcal{A}} = W^T\mathcal{A}V$, $\hat{\mathcal{B}} = W^T\mathcal{B}$, $\hat{\mathcal{C}} = \mathcal{C}V$.
    Compute $\hat{\mathcal{A}}z_i = \hat{\lambda}_i\hat{\mathcal{E}}z_i$ and $y_i^*\hat{\mathcal{A}} = \hat{\lambda}_i y_i^*\hat{\mathcal{E}}$.
    $\alpha_i \leftarrow -\lambda_i$, $b_i^* \leftarrow -y_i^*\hat{\mathcal{B}}$ and $c_i \leftarrow \hat{\mathcal{C}}z_i$, for $i = 1, \cdots, r$.
    $V = \Big[(\alpha_1\mathcal{E} - \mathcal{A})^{-1}\mathcal{B}b_1, \cdots, (\alpha_r\mathcal{E} - \mathcal{A})^{-1}\mathcal{B}b_r\Big]$,
    $W = \Big[(\alpha_1\mathcal{E}^T - \mathcal{A}^T)^{-1}\mathcal{C}^T c_1, \cdots, (\alpha_r\mathcal{E}^T - \mathcal{A}^T)^{-1}\mathcal{C}c_r\Big]$.
    $i = i + 1$
end while
$\hat{\mathcal{E}} = W^T\mathcal{E}V$,   $\hat{\mathcal{A}} = W^T\mathcal{A}V$,   $\hat{\mathcal{B}} = W^T\mathcal{B}$   and   $\hat{\mathcal{C}} = \mathcal{C}V$.
```

interpolation is to construct V and W such that the reduced transfer function $\hat{\mathrm{G}}(s)$ tangentially interpolates the original transfer function $\mathrm{G}(s)$ at a predefined set of interpolation points and some fixed tangent directions. That is

$$\begin{aligned} &\mathrm{G}(\alpha_i)b_i = \hat{\mathrm{G}}(\alpha_i)b_i, \quad c_i^T\mathrm{G}(\alpha_i) = c_i^T\hat{\mathrm{G}}(\alpha_i), \quad \text{and} \\ &c_i^T\mathrm{G}(\alpha_i)b_i = c_i^T\hat{\mathrm{G}}(\alpha_i)b_i, \quad \text{for } i = 1, \cdots, r, \end{aligned}$$

where $b_i \in \mathbb{C}^m$ and $c_i \in \mathbb{C}^p$ are the right and left tangential directions, respectively, and correspond to the interpolation points α_i. With these quantities, the rational tangential interpolation can be achieved. The IRKA based interpolatory projection methods for MIMO systems have been discussed in [64, 36], where the algorithm updates interpolation points as well as tangential directions until the reduced system satisfies the necessary condition for $\mathcal{H}_2$-optimality. We have summarized a complete procedure for a MIMO system in Algorithm 2.

2.2.3 Model reduction of a second order system

In the previous section we have introduced second order ODE systems. A classical approach to find a reduced order model (ROM) of second order systems is first to rewrite the systems into first-order form, then apply model order reduction techniques to find reduced state space systems [110, 37, 29, 20]. Now the question arises: can we return to the second order form from the reduced systems? In general, the answer is negative since the structure of the original model is destroyed in the reduced order form. Sometimes, the preservation of second order structure in the reduced systems is essential to perform the simulation, optimization and controller design if the software tools are specially designed for second order systems. Moreover, structure preserving reduced models allow meaningful physical interpretation and provide more accurate approximation which we will see later. Recently, structure preserving model reduction of second order systems received much attention. See e.g., [76, 11, 104, 105, 16, 96, 29, 66, 20] and the references therein. The structure preserving model reduction of second order systems using balanced truncation was first discussed by Meyer and Srinivasan in [88] based on the second order Gramians defined above. Next in [96] four types of balancing criteria are shown to obtain four types of reduced models of second order systems based on the second order Gramians. Following [96], the authors in [20] show an efficient technique for model reduction of symmetric second order systems. The technique uses the low-rank factors of the second order Gramians to construct the balancing and truncation transformations.

The controllability Gramian $P \in R^{2n_\xi \times 2n_\xi}$ and the observability Gramian $Q \in R^{2n_\xi \times 2n_\xi}$ for the system (2.21) are the solutions of the Lyapunov equations

$$APE^T + EPA^T = -BB^T \quad \text{and} \quad A^TQE + E^TQA = -C^TC, \tag{2.49}$$

where $E,\ A,\ B,\ C$ are defined either as in (2.22) or (2.23). We consider R as a low-rank controllability Gramian factor such that $W_c \approx RR^T$. The structure of the first order system allows us to split R as

$$R = \begin{bmatrix} R_v^T & R_p^T \end{bmatrix}^T. \tag{2.50}$$

Therefore, the controllability Gramian can be written [29] as

$$P = \begin{bmatrix} P_v & P_o \\ P_o^T & P_p \end{bmatrix} \approx RR^T = \begin{bmatrix} R_v \\ R_p \end{bmatrix} \begin{bmatrix} R_v^T & R_p^T \end{bmatrix} = \begin{bmatrix} R_vR_v^T & R_vR_p^T \\ R_pR_v^T & R_pR_p^T \end{bmatrix}.$$

Hence we have

$$P_v \approx R_vR_v^T \qquad \text{and} \qquad P_p \approx R_pR_p^T.$$

Similarly, considering $Q \approx LL^T$ we have

$$Q_v \approx L_vL_v^T \qquad \text{and} \qquad Q_p \approx L_pL_p^T,$$

where $L = \begin{bmatrix} L_v^T & L_p^T \end{bmatrix}^T$. Apparently, R_v and R_p are obtained from the first n_ξ rows and the lower n_ξ rows of R, respectively. Analogously, L_v and L_p can be obtained from the first n_ξ rows and the lower n_ξ rows of the low-rank observability Gramian factor L. Once we have R_α and L_β ($\alpha \in \{v, p\}$, $\beta \in \{v, p\}$), the balancing transformation can be formed [96, 20] using the SVD

$$R_\alpha^T M L_\beta = U_{\alpha\beta}\Sigma_{\alpha\beta}V_{\alpha\beta}^T = \begin{bmatrix} U_{\alpha\beta,1} & U_{\alpha\beta,2} \end{bmatrix} \begin{bmatrix} \Sigma_{\alpha\beta,1} & \\ & \Sigma_{\alpha\beta,2} \end{bmatrix} \begin{bmatrix} V_{\alpha\beta,1}^T \\ V_{\alpha\beta,2}^T \end{bmatrix}, \tag{2.51}$$

and defining

$$W_s := L_\beta U_{\alpha\beta,1}\Sigma_{\alpha\beta,1}^{-\frac{1}{2}}, \quad V_s := R_\alpha V_{\alpha\beta,1}\Sigma_{\alpha\beta,1}^{-\frac{1}{2}}, \tag{2.52}$$

where $U_{\alpha\beta,1}$ and $V_{\alpha\beta,1}$ are composed of the leading k columns of $U_{\alpha\beta}$ and $V_{\alpha\beta}$, respectively, $\Sigma_{\alpha\beta,1}$ is the first $k \times k$ block of the matrix $\Sigma_{\alpha\beta}$. Applying $W_s,\ V_s \in \mathbb{R}^{n_\xi \times k}$ with $k \ll n_\xi$ in (2.21), we obtain the reduced models

$$\begin{aligned} \hat{\mathcal{M}}\ddot{\hat{\xi}}(t) + \hat{\mathcal{D}}\dot{\hat{\xi}}(t) + \hat{\mathcal{K}}\hat{\xi}(t) &= \hat{\mathcal{H}}u(t), \\ \hat{y}(t) &= \hat{\mathcal{L}}x(t) + \hat{D}_s u(t), \end{aligned} \tag{2.53}$$

where

$$\begin{aligned} &\hat{\mathcal{M}} = W_s^T \mathcal{M} V_s, \ \ \hat{\mathcal{D}} = W_s^T \mathcal{D} V_s, \ \ \hat{\mathcal{K}} = W_s^T \mathcal{K} V_s, \\ &\hat{\mathcal{H}} = W_s^T \mathcal{H}, \ \ \hat{\mathcal{L}} = \mathcal{L} V_s, \ \ \hat{D}_s := D_s. \end{aligned} \tag{2.54}$$

When $\alpha = \beta = v$, the balancing technique by the above procedure is called velocity-velocity (VV) balancing. Likewise position-position (PP) balancing is obtained if $\alpha = \beta = p$, velocity-position (VP) balancing is obtained if $\alpha = v, \beta = p$, and position-velocity (PV) balancing is obtained if $\alpha = p, \beta = v$.

2.3 The LRCF-ADI iteration and related issues

In the previous section, we have already seen that to implement the balancing based MOR for the large sparse dynamical systems with few inputs and outputs (see Algorithm 1), the key tools are the low-rank controllability Gramian factor $\mathcal{R}$ and the observability Gramian factor $\mathcal{L}$. During the recent decades, several methods have been developed [93, 81, 108, 22] that allow to exploit the fact that often all coefficient matrices are sparse and the number of inputs and outputs is very small compared to the number of DoFs. The LRCF-ADI iteration [81, 22] is one of such efficient methods. This prominent method is derived from the *ADI* (alternating direction implicit) *iteration* introduced in [84]. Details on the derivation of the LRCF-ADI iteration can be found in, e.g., [79]. This iterative approach is also extended by Stykel in [112] to compute the low-rank Gramian factors by solving the generalized *projected Lyapunov equations* for descriptor systems. We will focus on these issues in the later chapters because it is one of the main interests of this thesis.

Much research has been done in the development of the LRCF-ADI iteration over the last two decades. The most recent developments were performed by Benner et. al. in [19, 20]. Usually, the computed low-rank Gramian factors via the LRCF-ADI method are complex due to the effect of complex ADI shift parameters. In [19], the authors show an efficient technique to compute real low-rank Gramian factors by cleverly handling the complex shift parameters. On the other hand, a computationally cheap approach, a low-rank residual based stopping criterion of

Algorithm 3: G-LRCF-ADI iteration.

```
Input  : $\mathcal{E}$, $\mathcal{A}$, $\mathcal{B}$, $\{\mu_i\}_{i=1}^{J}$.
Output: $\mathcal{R} = Z_i$, such such that $\mathcal{P} \approx \mathcal{R}\mathcal{R}^H$.
$Z_0 = [\,]$.
for $i = 1 : i_{max}$ do
    if $i = 1$ then
        $V_i = (\mathcal{A} + \mu_1 \mathcal{E})^{-1}\mathcal{B}$.
    else
        $V_i = \left[V_{i-1} - (\mu_i + \overline{\mu_{i-1}})\,(\mathcal{A} + \mu_i \mathcal{E})^{-1}\,\mathcal{E}V_{i-1}\right]$.
    end if
    Update $Z_i = \begin{bmatrix} Z_{i-1} & \sqrt{-2\,\mathrm{Re}\,(\mu_i)}V_i \end{bmatrix}$.
end for
```

the LRCF-ADI iteration is introduced in [20]. For convenience, we briefly review both the ideas for the generalized systems as in (2.1) and combine them in a single algorithm.

Recall the generalized (G-)LRCF-ADI iteration in [102, Algorithm 5.1] which is presented again in Algorithm 3. This algorithm successively generates the columns of the low-rank controllability Gramian factor $\mathcal{R}$ by solving the Lyapunov equation (2.12). For the low-rank factor of the observability Gramian, one can follow the same algorithm to solve the observability Lyapunov equation (2.13). In that case, the inputs $\mathcal{E}$, $\mathcal{A}$ and $\mathcal{B}$ are replaced by $\mathcal{E}^T$, $\mathcal{A}^T$ and $\mathcal{C}^T$. In this thesis all the details are given for the low-rank controllability Gramian factor. The low-rank observability Gramian factor can be handled in the same manner.

A set of optimal *ADI shift parameters* or simply *shift parameters* $\{\mu_i\}_{i=1}^{J} \subset \mathbb{C}^-$ are necessary for fast convergence of the algorithm. We will discuss the shift parameter selection criterion later in this section. In this algorithm we also see that if the maximum number of iterations $i_{\max}$ is greater than the number of shifts J, then the shift parameters are used in a cyclic way.

Although in Algorithm 3 all of the input matrices $\mathcal{E}$, $\mathcal{A}$ and $\mathcal{B}$ are real, due to the complex shift parameters in each iteration step, the updated Z_i store complex data, which increases the overall complexity and memory requirements of the method. Moreover, in the balancing based methods using these complex Gramian factors yields complex reduced systems by performing some complex arithmetic operations. This problem is resolved in [19]. In this regard, the important assumption is that the selected ADI shift parameters should be proper.

Definition 2.11. *The ADI shift parameters $\{\mu_i\}_{i=1}^{J} \subset \mathbb{C}^-$ are called proper if μ_i and μ_{i+1} are complex conjugates of each other when one of them is complex.*

In [19] it is shown that at the $(i+1)$-st iteration of the G-LRCF-ADI iteration, V_{i+1}

can be computed by

$$V_{i+1} = \overline{V}_i + 2\delta \operatorname{Im}(V_i), \tag{2.55}$$

where $\delta = \frac{\operatorname{Re}(\mu_i)}{\operatorname{Im}(\mu_i)}$. This identity states that in Algorithm 3, corresponding to $\mu_{i+1} = \overline{\mu}_i$, V_{i+1} can be computed explicitly from V_i, which releases us from solving a shifted linear system with $\mathcal{A} + \overline{\mu}_i\mathcal{E}$. This idea also results in the following theorem.

Theorem 2.1. *Let us assume a set of proper ADI shift parameters. For a pair of complex conjugate shifts $\{\mu_i, \mu_{i+1} := \overline{\mu_i}\}$, the two subsequent block iterates V_i and V_{i+1} of Algorithm 3 satisfy*

$$[V_i, V_{i+1}] = \left[\sqrt{-2\operatorname{Re}(\mu_i)}(\operatorname{Re}(V_i) + \delta \operatorname{Im}(V_i)), \sqrt{-2\operatorname{Re}(\mu_i)}\sqrt{\delta^2+1}\operatorname{Im}(V_i)\right]. \tag{2.56}$$

This theorem reveals that for a pair of complex conjugate shifts at any iteration step in the G-LRCF-ADI iteration, Z_i can be updated by

$$Z_{i+1} = [Z_{i-1}, \sqrt{-2\operatorname{Re}(\mu_i)}(\operatorname{Re}(V_i) + \delta \operatorname{Im}(V_i)), \sqrt{-2\operatorname{Re}(\mu_i)}\sqrt{\delta^2+1}\operatorname{Im}(V_i)]. \tag{2.57}$$

A version of the GLRCF-ADI algorithm is summarized in Algorithm 4, which computes low-rank real Gramian factors.

Additionally, Algorithm 3 can be stopped whenever the norm of the *ADI-residual*

$$\mathcal{F}(Z_i) = \mathcal{A}_i Z_i {Z_i}^T \mathcal{E}^T + \mathcal{E} Z_i {Z_i}^T \mathcal{A}^T + \mathcal{B}\mathcal{B}^T \tag{2.58}$$

becomes very small. But computing $\|\mathcal{F}(Z_i)\|$ in Frobenius-norm or 2-norm in each iteration step is an expensive task, since in each iteration the resulting residual (2.58) is an $n \times n$ matrix. Recently, in [20] the authors show that in the i-th iteration, the ADI-residual in (2.58) can be represented as

$$\mathcal{F}(Z_i) = W_i W_i^H,$$

with

$$W_i = \left(\prod_{j=1}^{i}(\mathcal{A} - \overline{\mu}_j\mathcal{E})(\mathcal{A} + \mu_j\mathcal{E})^{-1}\right)\mathcal{B}. \tag{2.59}$$

And the V_i iterate in the GLRCF-ADI can be expressed as [20]

$$V_i = (\mathcal{A} + \mu_i\mathcal{E})^{-1} W_{i-1}. \tag{2.60}$$

From (2.59) again we obtain

$$\begin{aligned} W_i &= (\mathcal{A} - \overline{\mu}_i\mathcal{E})(\mathcal{A} + \mu_i\mathcal{E})^{-1}\left(\prod_{j=1}^{i-1}(\mathcal{A} - \overline{\mu}_j\mathcal{E})(\mathcal{A} + \mu_j\mathcal{E})^{-1}\right)\mathcal{B} \\ &= (\mathcal{A} - \overline{\mu_i}\mathcal{E})(\mathcal{A} + \mu_i\mathcal{E})^{-1} W_{i-1} \\ &= \left[I - (\mu_i + \overline{\mu_i})\mathcal{E}(\mathcal{A} + \mu_i\mathcal{E})^{-1}\right] W_{i-1} \\ &= W_{i-1} - 2\operatorname{Re}(\mu_i)\mathcal{E}V_i \quad (\text{using } 2.60). \end{aligned} \tag{2.61}$$

Algorithm 4: G-LRCF-ADI iteration (for a real low-rank Gramian factor).

```
Input  : $\mathcal{E}$, $\mathcal{A}$, $\mathcal{B}$, $\{\mu_i\}_{i=1}^{J}$.
Output: $\mathcal{R} = Z_i$, such that $\mathcal{P} \approx \mathcal{R}\mathcal{R}^T$.
$Z_0 = []$.
for $i = 1 : i_{max}$ do
    if $i = 1$ then
        $V_i = (\mathcal{A} + \mu_1 \mathcal{E})^{-1}\mathcal{B}$.
    else
        $V_i = \left[V_{i-1} - (\mu_i + \overline{\mu_{i-1}})\,(\mathcal{A} + \mu_i \mathcal{E})^{-1}\,\mathcal{E}V_{i-1}\right]$.
    end if
    if $\operatorname{Im}(\mu_i) = 0$ then
        $Z_i = \begin{bmatrix} Z_{i-1} & \sqrt{-2\mu_i}V_i \end{bmatrix}$.
    else
        $\gamma = 2\sqrt{-\operatorname{Re}(\mu_i)}, \quad \delta = \frac{\operatorname{Re}(\mu_i)}{\operatorname{Im}(\mu_i)}$,
        $Z_{i+1} = \begin{bmatrix} Z_{i-1} & \sqrt{2}\gamma(\operatorname{Re}(V_i) + \delta \operatorname{Im}(V_i)) & \sqrt{2}\gamma\sqrt{(\delta^2+1)}.\operatorname{Im}(V_i) \end{bmatrix}$,
        $V_{i+1} = \overline{V} + \delta \operatorname{Im}(V_i)$.
        $i = i + 1$
    end if
end for
```

In the case of real setting when we consider $\mu_{i+1} := \overline{\mu_i}$, one must compute

$$\begin{aligned} W_{i+1} &= W_i - 2\operatorname{Re}(\mu_i)\mathcal{E}V_{i+1} \\ &= W_{i-1} - 2\operatorname{Re}(\mu_i)\mathcal{E}V_i - 2\operatorname{Re}(\mu_i)\mathcal{E}V_{i+1} \\ &= W_{i-1} - 2\operatorname{Re}(\mu_i)\mathcal{E}\left(V_i + \overline{V_i} + 2\delta\operatorname{Im}(V_i)\right) \quad \text{(using} \quad (2.55)) \\ &= W_{i-1} - 4\operatorname{Re}(\mu_i)\mathcal{E}\left(\operatorname{Re}(V_i) + \delta\operatorname{Im}(V_i)\right), \end{aligned} \tag{2.62}$$

where δ is defined in (2.55). The rank of W_i is at most m, i.e., the number of columns of $\mathcal{B}$. Therefore, the computation of the Frobenius-norm or 2-norm of $\|W_i W_i^T\| = \|W_i^T W_i\|$ in each iteration is extremely cheap. Applying these strategies (computation of real Gramian factors and low-rank residual based stopping techniques), the updated GLRCF-ADI is rewritten in Algorithm 5.

ADI shift parameter selection

The convergence speed of the LRCF-ADI algorithms presented above heavily depends on a set of ADI shift parameters. The ADI shift parameters were originally introduced by Wachspress in [126] to solve Lyapunov equations using the ADI methods. The author shows that a set of optimal ADI shift parameters $\{\mu_i\}_{i=1}^{J}$ for

Algorithm 5: G-LRCF-ADI iteration (updated).

```
Input  : $\mathcal{E},\ \mathcal{A},\ \mathcal{B},\ \{\mu_i\}_{i=1}^{J}$.
Output: $\mathcal{R} = Z_i$, such that $\mathcal{P} \approx \mathcal{R}\mathcal{R}^T$.
$W_0 = \mathcal{B}, \quad Z_0 = [\,], \quad i = 1.$
while $\|W_{i-1}^T W_{i-1}\| \geq tol$ or $i \leq i_{max}$ do
    Compute $V_i = (\mathcal{A} + \mu_i \mathcal{E})^{-1} W_{i-1}$.
    if $\mathrm{Im}\,(\mu_i) = 0$ then
        $Z_i = \begin{bmatrix} Z_{i-1} & \sqrt{-2\mu_i} V_i \end{bmatrix}$.
        $W_i = W_{i-1} - 2\mu_i \mathcal{E} V_i$.
    else
        $\gamma = -2\,\mathrm{Re}\,(\mu_i), \quad \delta = \frac{\mathrm{Re}\,(\mu_i)}{\mathrm{Im}\,(\mu_i)}$,
        $Z_{i+1} = \begin{bmatrix} Z_{i-1} & \sqrt{2\gamma}(\mathrm{Re}\,(V_i) + \delta\,\mathrm{Im}\,(V_i)) & \sqrt{2\gamma}\sqrt{(\delta^2+1)}\,\mathrm{Im}\,(V_i) \end{bmatrix}$,
        $W_{i+1} = W_{i-1} + 2\gamma\mathcal{E}\,(\mathrm{Re}\,(V_i) + \delta\,\mathrm{Im}\,(V_i))$.
        $i = i + 1$
    end if
    $i = i + 1$
end while
```

the system (2.1) can be computed by solving the so called *ADI min-max* problem [125, 127]

$$\min_{\mu_1,\cdots,\mu_j \subset \mathbb{C}^-} \left(\max_{1\leq l\leq n} \left| \prod_{i=1}^{J} \frac{\overline{\mu_i} - \lambda_l}{\mu_i + \lambda_l} \right| \right), \quad \lambda_l \in \Lambda(\mathcal{A},\mathcal{E}), \tag{2.63}$$

where $\Lambda(\mathcal{A},\mathcal{E})$ denotes the spectrum of the matrix pencil (2.2). For a large-scale system, determining the entire spectrum of $(\mathcal{A},\mathcal{E})$ is almost impossible. Therefore, in the literature several techniques are proposed, see, e.g., [93, 79, 24, 102] to solve the min-max problem (2.63) using a much smaller part of the spectrum. Currently, one such commonly used technique is Penzl's *heuristic approach* introduced in [93], where k_+ *Ritz values* (see, e.g., [60]) and k_- ($k_+, k_- \ll n$) reciprocal Ritz values with respect to $\mathcal{E}^{-1}\mathcal{A}$ and $\mathcal{A}^{-1}\mathcal{E}$, respectively, are employed. A complete procedure of the heuristic approach can be found in [93, Algorithm 5.1]. Although computing the Ritz values is computationally expensive, this approach is applicable to a large-scale standard or generalized (where $\mathcal{E}$ is invertible) state space systems. However, for large-scale descriptor systems, computing the Ritz values with respect to $\mathcal{E}^{-1}\mathcal{A}$ is a challenging task since $\mathcal{E}$ is then not invertible. This thesis will discuss the solution of this problem for a large-scale descriptor system in the next chapters.

Another promising ADI shift selection criterion that we focus on is the *adaptive approach* introduced in [21]. This approach is reported to be superior to the heuristic approach, especially for the descriptor systems discussed regarding the computational issues. In this approach, the ADI shift parameters are generated and updated automatically by the LRCF-ADI algorithm itself. There, the computed k shifts are the eigenvalues of the projected matrix pencil

$$\lambda U^T \mathcal{E} U - U^T \mathcal{A} U, \quad \lambda \in \mathbb{C}, \tag{2.64}$$

where $U \in \mathbb{R}^{n \times k}$ $(k \ll n)$. For a set of initial shifts, U in (2.64) is the span of W_0. Then, whenever all shifts in the current set have been used, the matrix pencil is projected by using U as the span of the current V_i and the eigenvalues are used as the next set of shifts. In this procedure, specially, for a SISO system or a system with few inputs and outputs, sometimes the projected pencil may become unstable. In this situation, it is suggested in [21] to use the previous set of shift parameters for the next cycle of the iterations. Sometimes, in this procedure, the convergence rate of the LRCF-ADI iteration is not as good as desired. To resolve this problem, we propose slightly different initialization and also updating criterion for the adaptive ADI shift parameters approach. We will discuss this issue in Chapter 3 (Section 3.4), Chapter 4 (Section 4.4) and Chapter 5 (Section 5.4)

2.4 Model reduction of descriptor systems

In the above we have discussed the background theory only for the non-descriptor generalized systems. This is important since our approach for model reduction of descriptor systems is based on the transformation of descriptor systems into equivalent ODE systems. As mentioned before this ODE formulation is not required explicitly in our computations. This idea was introduced in [53, 70, 68] for first order structured differential-algebraic systems. In the following, we briefly discuss the structure of the descriptor systems and review the reduction techniques from the literature.

2.4.1 Descriptor systems

A descriptor system is a special form of a generalized state space system. Systems (2.1) with singular matrix $\mathcal{E}$, i.e., $\det(\mathcal{E}) = 0$, are often called *descriptor systems*. In some literature, they are also known as *singular systems* or *differential-algebraic equations (DAEs)* (see [112, 58, 75]). A descriptor system is solvable if the corresponding matrix pencil, defined in (2.2), is regular, i.e., $\det(\mathcal{P}_c) \neq 0$. In the case of regular pencils, there exist invariable matrices S_L and $\mathcal{S}_R$, so that $\mathcal{E}$ and $\mathcal{A}$ have the following Weierstrass canonical representations [75]:

$$\mathcal{E} = \mathcal{S}_L \begin{bmatrix} \mathcal{I}_{n_f} & 0 \\ 0 & \mathcal{N} \end{bmatrix} \mathcal{S}_R \quad \text{and} \quad \mathcal{A} = \mathcal{S}_L \begin{bmatrix} \mathcal{A}_1 & 0 \\ 0 & \mathcal{I}_{n_\infty} \end{bmatrix} \mathcal{S}_R, \tag{2.65}$$

where $\mathcal{N}$ is nilpotent with nil-potency ν, i.e., $N^{\nu-1} \neq 0$ but $N^\nu = 0$. Usually the nilpotency ν indicates the index of the descriptor system. In the literature this is known as *algebraic index*. However, there are other types of indices for descriptor systems, such as the *differentiation index* [9], the *tractability index* [86], and so on. The most commonly used concept is differentiation index, which is defined by the number of derivatives that take place in a DAE system to convert it into an equivalent ODE system.

Definition 2.12. *The differentiation index of a DAE system is the minimum number of times that all or part of the system must be differentiated with respect to t in order to find explicit ODE systems.*

Note that for an LTI system, the algebraic index and the differential index coincide.

This thesis is concerned with special structured descriptor systems considering their applications in different fields. The descriptor systems that we focus on have the following form

$$\begin{bmatrix} E_{11} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u(t), \tag{2.66a}$$

$$y(t) = \begin{bmatrix} C_1 & C_2 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + D_a u(t), \tag{2.66b}$$

where $x(t) \in \mathbb{R}^n$ ($n = n_1 + n_2$, with $x_1(t) \in \mathbb{R}^{n_1}$ and $x_2(t) \in \mathbb{R}^{n_2}$) as generalized states, where E_{11} and A_{11} have full rank. The descriptor system (2.66) is called

- index 1 if $\det(A_{22}) \neq 0$,
- index 2 if $A_{22} = 0$ and $\det(A_{21}A_{12}) \neq 0$, and
- index 3 if $A_{22} = 0$ and $\det(A_{21}A_{12}) = 0$.

Using the Weierstrass canonical representation defined in (2.65), the transfer function ($\mathrm{G}(s)$) of a descriptor systems can be written as [112]

$$\mathrm{G}(s) = \mathrm{G}_{sp}(s) + \mathrm{G}_p(s), \tag{2.67}$$

where $\mathrm{G}_{sp}(s)$ and $\mathrm{G}_p(s)$ respectively denote the strictly proper and polynomial parts of $\mathrm{G}(s)$.

Definition 2.13. *The transfer function $G(s)$ is called proper if $\lim_{s\to\infty} G(s) < \infty$. Otherwise, it is called improper. If $\lim_{s\to\infty} G(s) = 0$, then $G(s)$ is called strictly proper.*

2.4.2 MOR of structured descriptor systems

The idea of model reduction for large-scale descriptor systems was first introduced by Stykel in [111, 112]. The author discusses a general framework for the BT method applied to descriptor systems. In general, the method is based on explicit computation of the spectral projectors onto the left and right deflating subspaces of the matrix pencil corresponding to the finite and infinite eigenvalues. Although these projectors are available for particular systems, their computation is expensive. In this thesis we focus on a method to avoid computing such kind of projectors explicitly. An efficient balancing based model reduction method for structured first order index 1 DAEs is discussed in [53]. The authors show that due to the non singularity of block matrix A_{22}, the index 1 system (2.66) can be converted into an ODE systems. Although the original system is sparse, the converted ODE system is typically dense, which in general causes undesired computational complexity. Therefore, in the practical implementation, the explicit computation of ODE system is avoided. We generalize this strategy for balancing based model reduction of second order index 1 systems. We will discuss this in Chapter 4.

Heinkenschloss et. al. in [70] discuss an efficient balancing based method for structured first order index 2 DAEs. The main task is to convert the DAE into an equivalent projected ODE by applying an appropriate projector. The projector can be constructed by exploiting the knowledge of the structure of the system. Note that this projector is also a spectral projector as in [112], since the projected system preserves all the finite eigenvalues of the original systems. We will discuss these issues in Chapter 5. Due to some properties of the projector in [70], it is shown that to implement the BT based model reduction of such a system, explicit computation of the projector is not required. This avoidance of projectors is followed by [68] while implementing the interpolatory technique via IRKA, for model reduction of such structured first order index 2 DAEs. We exploit the idea of [70] for model reduction of first order index 2 *unstable* DAEs with small numbers of unstable poles. This approach is discussed in Chapter 3. To show the model reduction of second order index 3 descriptor systems presented in Chapter 5, we also depend on the strategies in [70] and [68] for the balancing and interpolatory methods, respectively. We leave this section to discuss more details in the relevant chapters.

Chapter 3

First Order Index 2 Unstable Descriptor Systems

In this chapter we study model reduction of a class of structured index 2 descriptor systems of the form

$$\underbrace{\begin{bmatrix} E_1 & 0 \\ 0 & 0 \end{bmatrix}}_{\check{E}} \begin{bmatrix} \dot{v}(t) \\ \dot{p}(t) \end{bmatrix} = \underbrace{\begin{bmatrix} A_1 & A_2 \\ A_2^T & 0 \end{bmatrix}}_{\check{A}} \begin{bmatrix} v(t) \\ p(t) \end{bmatrix} + \underbrace{\begin{bmatrix} B_1 \\ 0 \end{bmatrix}}_{\check{B}} u(t), \tag{3.1a}$$

$$y(t) = \underbrace{\begin{bmatrix} C_1 & 0 \end{bmatrix}}_{\check{C}} \begin{bmatrix} v(t) \\ p(t) \end{bmatrix}, \tag{3.1b}$$

where $v(t) \in \mathbb{R}^{n_1}$, $p(t) \in \mathbb{R}^{n_2}$ $(n_1 > n_2)$ are the states, $u \in \mathbb{R}^m$ are the inputs, and $y \in \mathbb{R}^p$ are the outputs and in which $\check{E}$, $\check{A}$, $\check{B}$ and $\check{C}$ are all sparse matrices with appropriate dimensions. We assume that some of the eigenvalues of the matrix pencil, $\lambda\check{E} - \check{A}$ lie in $\mathbb{C}^+$, which makes the system (3.1) unstable. Such models arise, for instance, from a spatial discretization of Navier-Stokes equations with moderate Reynolds number ($Re \geq 300$) using the finite element method (see, Section 3.1 for details). In this chapter we mainly focus on balancing based model reduction techniques for the system in (3.1). To obtain IRKA based reduced models, we can directly follow the approaches as discussed in [68, Section 6], since IRKA does not rely on the stability of the system. In principle, one can apply the balanced truncation technique to this model by stabilizing the system first using a proper stabilizing feedback matrix (SFM) and then following the approach in [70]. Following the ideas in [135], we apply the balancing and truncating transformations computed with respect to the stabilized system to the original unstable system. To compute the controllability and observability Gramian factors we need to solve two projected algebraic Lyapunov equations of the stabilized system. Again following [135] we employ Bernoulli stabilization to derive the SFM. The main advantage of the Bernoulli stabilization is that it only changes the anti-stable eigenvalues of the system. Thus the required Bernoulli equation can be restricted to these and is of the same dimension as the corresponding eigenspaces (see, Section 3.2).

This chapter also presents an updated version of the LRCF-ADI algorithm to solve the projected Lyapunov equations for the Bernoulli stabilized system. In order to ensure fast convergence of LRCF-ADI, we also discuss and resolve the difficulties in computing shift parameters for the models of flow control considered here. Moreover, here, we show that a Riccati-based boundary feedback stabilization matrix [12] for the original model can be computed using a reduced order model. The proposed method is applied to data for a spatially FEM semi-discretized linearized Navier-Stokes model. Numerical results are discussed for both the BT model reduction, as well as, the reduced order model based stabilization. We also compare the balancing based results with those of the interpolatory based method. The results of this chapter have been published in [32].

3.1 Motivating example

The *linearization principle* as presented in [109], basically states that a general nonlinear model can be stabilized by a linear quadratic regulator (LQR) for a linearization of itself in the vicinity of the linearization point. The basic idea is that if the regulator is working properly, the vicinity where the linearization is a proper approximation of the nonlinear system is never left. This principle has been exploited by the authors in [12] for a Navier-Stokes model for the *von Kármán vortex street*. The linearized Navier-Stokes equations that arose there and that we consider in this chapter are

$$\begin{aligned} \frac{\partial}{\partial t}\vec{v} - \frac{1}{Re}\Delta\vec{v} + (\vec{w}.\nabla)\vec{v} + (\vec{v}.\nabla)\vec{w} + \nabla p = 0, \\ \nabla.\vec{v} = 0, \end{aligned} \tag{3.2}$$

where $\vec{v}, \vec{w}$ denote velocity vectors, p the pressure and Re is the Reynolds number. The vector $\vec{w}$ represents the stationary solution of the incompressible nonlinear Navier-Stokes equations and $\vec{v}$ is the deviation of the original state from the stationary solution. The boundary and initial conditions, as well as the derivation of this model, are given in [12]. There the authors apply a mixed finite element method based on the well known Taylor-Hood finite elements [71] to discretize equation (3.2). The coarsest discretization of the domain for the *von Kármán vortex street* example from [12] is shown in Figure 3.1. This yields the differential-algebraic equations (3.1a), where $v(t)$ denotes the nodal vector of discretized velocity deviations and $p(t)$ the discretized pressure. Additionally, the vertical velocities in the observation nodes depicted in Figure 3.1 in the domain are modeled by the output equation (3.1b). The system (3.1) remains stable, i.e., the finite spectrum of the matrix pencil $\lambda\check{E} - \check{A}$ is located in the negative half plane $\mathbb{C}^-$, as long as the Reynolds number Re is small. However, already for moderate Reynolds numbers

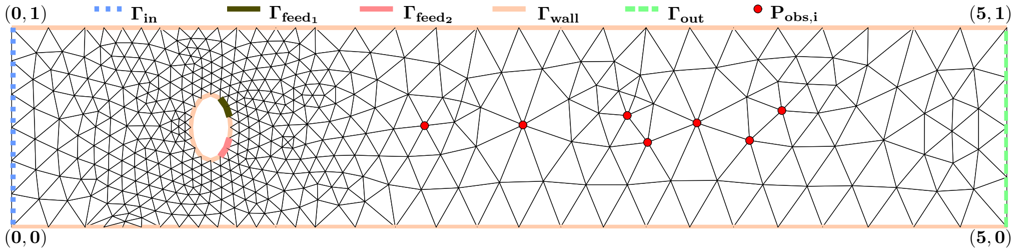


Figure 3.1: Initial discretization of the *von Kármán vortex street* with coordinates, boundary parts and observation points (source [12]).

(e.g., in the configuration of Figure 3.1 $Re \geq 300$) a few finite eigenvalues move to the positive half plane, $\mathbb{C}^+$ [4].

3.2 BT for unstable systems

In Chapter 2 we have not introduced the idea of balanced truncation for an unstable generalized state space system. Therefore, in this section we concentrate on BT for unstable systems

$$\begin{aligned} \mathcal{E}\dot{x}(t) &= \mathcal{A}x(t) + \mathcal{B}u(t), \\ y(t) &= \mathcal{C}x(t), \end{aligned} \tag{3.3}$$

via *Bernoulli stabilization*. All the matrices and vectors are defined in (2.1). The helpful feature of our investigated example is that still the number of anti-stable eigenvalues is very small. This is exactly the property we exploit for fast computation of the ROMs and ROM based feedback matrices. In Chapter 2 we have recalled classical (Lyapunov based) balancing for stable systems. The main ingredients there are the two Gramians (e.g., [5])

$$\begin{aligned} \mathcal{P} &= \int_0^\infty \mathrm{e}^{\mathcal{E}^{-1}\mathcal{A}t}\mathcal{E}^{-1}\mathcal{B}\mathcal{B}^T\mathcal{E}^{-T}\mathrm{e}^{\mathcal{A}^T\mathcal{E}^{-T}t}\,\mathrm{d}t, \\ \mathcal{Q} &= \int_0^\infty \mathrm{e}^{(\mathcal{E}^{-1}\mathcal{A})^T t}\mathcal{E}^{-T}\mathcal{C}^T\mathcal{C}\mathcal{E}^{-1}\mathrm{e}^{\mathcal{E}^{-1}\mathcal{A}t}\,\mathrm{d}t, \end{aligned}$$

which obviously do not exist if the system's unstable poles are controllable, which is in fact the desired case in our motivating example. In [135], the authors use the frequency domain representations of these integrals

$$\begin{aligned} \mathcal{P} &= \frac{1}{2\pi}\int_{-\infty}^{\infty}(\imath\omega\mathcal{E} - \mathcal{A})^{-1}\mathcal{B}\mathcal{B}^T(-\imath\omega\mathcal{E}^T - \mathcal{A}^T)^{-1}\,d\omega \\ \mathcal{Q} &= \frac{1}{2\pi}\mathcal{E}^T\{\int_{-\infty}^{\infty}(\imath\omega\mathcal{E}^T - \mathcal{A}^T)^{-1}\mathcal{C}^T\mathcal{C}(-\imath\omega\mathcal{E} - \mathcal{A})^{-1}\,d\omega\}\mathcal{E} \end{aligned}$$

to extend the definition of the Gramians to systems with no poles on the imaginary axis.

Following the theory in [135], the generalized controllability and observability Gramians $\mathcal{P}_s$, $\mathcal{Q}_s$ for such systems can be computed by solving the algebraic Lyapunov equations

$$\begin{aligned}(\mathcal{A}-\mathcal{B}\mathcal{K}_c^{fm})\mathcal{P}_s\mathcal{E}^T+\mathcal{E}\mathcal{P}_s(\mathcal{A}-\mathcal{B}\mathcal{K}_c^{fm})^T&=-\mathcal{B}\mathcal{B}^T,\\ (\mathcal{A}-\mathcal{K}_o^{fm}\mathcal{C})^T\mathcal{Q}_s\mathcal{E}+\mathcal{E}^T\mathcal{Q}_s(\mathcal{A}-\mathcal{K}_o^{fm}\mathcal{C})&=-\mathcal{C}^T\mathcal{C},\end{aligned} \tag{3.4}$$

where $\mathcal{K}_c^{fm}=\mathcal{B}^T\mathcal{X}_c\mathcal{E}$ and $\mathcal{K}_o^{fm}=\mathcal{E}\mathcal{X}_o\mathcal{C}^T$ are called Bernoulli stabilizing feedback matrices, due to the fact that the matrices $\mathcal{X}_c$ and $\mathcal{X}_o$ are the respective stabilizing solutions of the generalized algebraic Bernoulli equations

$$\begin{aligned}\mathcal{E}^T\mathcal{X}_c\mathcal{A}+\mathcal{A}^T\mathcal{X}_c\mathcal{E}&=\mathcal{E}^T\mathcal{X}_c\mathcal{B}\mathcal{B}^T\mathcal{X}_c\mathcal{E},\\ \mathcal{A}\mathcal{X}_o\mathcal{E}^T+\mathcal{E}\mathcal{X}_o\mathcal{A}^T&=\mathcal{E}\mathcal{X}_o\mathcal{C}^T\mathcal{C}\mathcal{X}_o\mathcal{E}^T.\end{aligned} \tag{3.5}$$

Now, since the Bernoulli stabilization only mirrors the anti-stable eigenvalues across the imaginary axis, it is sufficient to solve these Bernoulli equations only on the invariant subspaces corresponding to those eigenvalues. That is, for orthogonal matrices $T_c,\ T_o\in\mathbb{R}^{n\times l}$ spanning the left and right eigenspaces corresponding to the anti-stable eigenvalues, respectively, we define the Petrov-Galerkin projected system $(\check{\mathcal{E}},\check{\mathcal{A}},\check{\mathcal{B}},\check{\mathcal{C}})$ by

$$\check{\mathcal{E}}:=T_o^T\mathcal{E}T_c,\quad \check{\mathcal{A}}:=T_o^T\mathcal{A}T_c,\quad \check{\mathcal{B}}:=T_o^T\mathcal{B},\quad \check{\mathcal{C}}:=\mathcal{C}T_c,$$

where $\check{\mathcal{E}},\ \check{\mathcal{A}}\in\mathbb{R}^{l\times l}$, $\check{\mathcal{B}}\in\mathbb{R}^{l\times m}$, and $\check{\mathcal{C}}\in\mathbb{R}^{p\times l}$. The size of these projected matrices is very small, since we have considered a few anti-stable eigenvalues. Therefore, we solve very small sized projected Bernoulli equations

$$\begin{aligned}\check{\mathcal{E}}^T\check{\mathcal{X}}_c\check{\mathcal{A}}+\check{\mathcal{A}}^T\check{\mathcal{X}}_c\check{\mathcal{E}}&=\check{\mathcal{E}}^T\check{\mathcal{X}}_c\check{\mathcal{B}}\check{\mathcal{B}}^T\check{\mathcal{X}}_c\check{\mathcal{E}},\\ \check{\mathcal{A}}\check{\mathcal{X}}_o\check{\mathcal{E}}^T+\check{\mathcal{E}}\check{\mathcal{X}}_o\check{\mathcal{A}}^T&=\check{\mathcal{E}}\check{\mathcal{X}}_o\check{\mathcal{C}}^T\check{\mathcal{C}}\check{\mathcal{X}}_o\check{\mathcal{E}}^T,\end{aligned} \tag{3.6}$$

and construct $\mathcal{K}_c^{fm}=\mathcal{B}^TT_c\check{\mathcal{X}}_cT_c^T\mathcal{E}$ and $\mathcal{K}_o^{fm}=\mathcal{E}T_o^T\check{\mathcal{X}}_oT_o\mathcal{C}^T$. The projected Bernoulli equations in (3.6) can be solved by the *Matrix Sign Function* method presented in [13].

The low-rank factors of $\mathcal{P}_s$ and $\mathcal{Q}_s$ can also be computed by solving (3.4) using Algorithm 5, but avoiding to form the closed loop matrices stays crucial. We discuss this issue in more detail in Section 3.4. Now using these Gramian factors from the balancing and truncating transformations and applying them to the original unstable system we can compute an unstable ROM that satisfies the error bound as in (2.44), but with the $\mathcal{H}_\infty$-norm replaced by the $\mathcal{L}_\infty$-norm. Therefore, this error bound can not be translated to a global time domain error bound, as in the classic setting, due to the lack of a Parseval-identity-like result. In fact in the numerical experiments we observed that the error may very well grow over time.

3.3 BT for index 2 unstable descriptor systems

To apply the balancing based model reduction to the system (3.1), first we convert the system into an equivalent ODE system in order to make it fit into the framework for BT based model order reduction as discussed in the previous section. Recalling the strategy as in [70, Section 3], let us consider a projector of the form

$$\Pi_2 = I_{n_1} - A_2(A_2^T E_1^{-1} A_2)^{-1} A_2^T E_1^{-1}, \tag{3.7}$$

which satisfies $\operatorname{Null}(\Pi_2) = \operatorname{Range}(A_2)$, $\operatorname{Range}(\Pi_2) = \operatorname{Null}\left(A_2^T E_1^{-1}\right)$ and $\Pi_2 E_1 = E_1 \Pi_2^T$. These properties imply

$$A_2^T Y = 0 \quad \text{if and only if} \quad \Pi_2^T Y = Y, \tag{3.8}$$

i.e., the image of Π_2^T is exactly the subspace where the algebraic condition of the DAEs is satisfied. Now applying the projector to (3.1) and exploiting the property (3.8) we obtain the following projected system

$$\begin{aligned} \Pi_2 E_1 \Pi_2^T \dot{v}(t) &= \Pi_2 A_1 \Pi_2^T v(t) + \Pi_2 B_1 u(t), && \text{(3.9a)} \\ y(t) &= C_1 \Pi_2^T v(t). && \text{(3.9b)} \end{aligned}$$

The system dynamics of (3.9) are projected onto the $m_1 := n_1 - n_2$ dimensional subspace $\operatorname{Range}\left(\Pi_2^T\right)$ [70]. This subspace is, however, still represented in the coordinates of the n_1 dimensional space. The m_1 dimensional representation can be made explicit by employing the *thin* singular value decomposition (SVD)

$$\Pi_2 = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} S_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} = U_1 \Sigma_1 V_1^T = \Theta_{2,l} \Theta_{2,r}^T, \tag{3.10}$$

where $\Theta_{2,l} = U_1$ and $\Theta_{2,r} = V_1$ and in which $U_1,\ V_1 \in \mathbb{R}^{n_1 \times n_m}$ consist of the corresponding leading m_1 columns of U, $V \in \mathbb{R}^{n_1 \times n_1}$. Moreover, $\Theta_{2,l}, \Theta_{2,r}$ satisfy

$$\Theta_{2,l}^T \Theta_{2,r} = I_{m_1}. \tag{3.11}$$

This representation is always possible since Π_2 is a projector and therefore, $S_1 = I_{m_1}$. Inserting the decomposition of Π_2 as in (3.10) into (3.9) and considering $\tilde{v}(t) = \Theta_{2,l}^T v(t)$, we get

$$\begin{aligned} \Theta_{2,r}^T E_1 \Theta_{2,r} \dot{\tilde{v}}(t) &= \Theta_{2,r}^T A_1 \Theta_{2,r} \tilde{v}(t) + \Theta_{2,r}^T B_1 u(t), \\ y(t) &= C_1 \Theta_{2,r} \tilde{v}(t). \end{aligned} \tag{3.12}$$

System (3.12) practically is system (3.3) with the redundant equations removed by the $\Theta_{2,r}$ projection. We observe that the dynamical systems (3.1), (3.9) and (3.12) are equivalent in the sense that their finite spectrum is the same [48, Theorem 2.7.3] and the input-output relations are the same, i.e., they realize the same transfer function. In the following we will discuss how to avoid forming (3.12) explicitly to perform the model reduction.

Suppose that we want to apply balanced truncation to the system (3.12). To this end, we need to solve the Lyapunov equations

$$\begin{aligned}\Theta_{2,r}^T A_c \Theta_{2,r} \tilde{P} \Theta_{2,r}^T E_1^T \Theta_{2,r} + \Theta_{2,r}^T E_1 \Theta_{2,r} \tilde{P} \Theta_{2,r}^T A_c^T \Theta_{2,r} &= -\Theta_{2,r}^T B_1 B_1^T \Theta_{2,r},\\ \Theta_{2,r}^T A_o^T \Theta_{2,r} \tilde{Q} \Theta_{2,r}^T E_1 \Theta_{2,r} + \Theta_{2,r}^T E_1^T \Theta_{2,r} \tilde{Q} \Theta_{2,r}^T A_o \Theta_{2,r} &= -\Theta_{2,r}^T C_1^T C_1 \Theta_{2,r},\end{aligned} \tag{3.13}$$

where $A_c = A_1 - B_1 K_c^{fm}$, $A_o = A_1 - K_o^{fm} C_1$ and $\tilde{P} \in \mathbb{R}^{m_1 \times m_1}$, $\tilde{Q} \in \mathbb{R}^{m_1 \times m_1}$ are the corresponding projected controllability and observability Gramians. Again, K_c^{fm} and K_o^{fm} are the Bernoulli stabilizing feedback matrices and can be computed as described in Section 3.2. The solutions $\tilde{P}$, $\tilde{Q}$ of (3.13) are unique since we are assured that the respective dynamical system is asymptotically stable and symmetric positive (semi-)definite since the right hand side is semi-definite.

Now multiplying (3.13) by $\Theta_{2,l}$ from the left and $\Theta_{2,l}^T$ from the right and exploiting that $\Theta_{2,r} = \Pi_2^T \Theta_{2,r}$ (e.g., due to (3.10), (3.11)) we obtain

$$\begin{aligned}\Pi_2 A_c \Pi_2^T P \Pi_2 E_1^T \Pi_2^T + \Pi_2 E_1 \Pi_2^T P \Pi_2 A_c^T \Pi_2^T &= -\Pi_2 B_1 B_1^T \Pi_2^T,\\ \Pi_2 A_o^T \Pi_2^T Q \Pi_2 E_1 \Pi_2^T + \Pi_2 E_1^T \Pi_2^T Q \Pi_2 A_o \Pi_2^T &= -\Pi_2 C_1^T C_1 \Pi_2^T,\end{aligned} \tag{3.14}$$

where $P = \Theta_{2,r} \tilde{P} \Theta_{2,r}^T$ and $Q = \Theta_{2,r} \tilde{Q} \Theta_{2,r}^T$. These are the respective controllability and observability Lyapunov equations for the realization (3.9) and the solutions P, $Q \in \mathbb{R}^{n_1 \times n_1}$ are the corresponding controllability and observability Gramians. The system (3.14) is singular due to the fact that Π_2 is a projection. Uniqueness of solutions is reestablished by the condition that the solutions satisfy $P = \Pi_2^T P \Pi_2$ and $Q = \Pi_2^T Q \Pi_2$.

It is also an easy exercise to go back to (3.13) from (3.14). Let us consider $P \approx RR^T$, $Q \approx LL^T$ and $\tilde{P} \approx \tilde{R}\tilde{R}^T$, $\tilde{Q} \approx \tilde{L}\tilde{L}^T$. Then R, L, $\tilde{R}$ and $\tilde{L}$ are called approximate low-rank Cholesky factors. They fulfill the relation

$$R = \Theta_{2,r} \tilde{R} \qquad \text{and} \qquad L = \Theta_{2,r} \tilde{L}.$$

For large-scale problems, however, computing $\Theta_{2,r}$ is usually impossible due to memory limitations. Therefore, R and L are computed by solving (3.14). The balancing truncating transformations for (3.12) are

$$\tilde{W} = \tilde{R} U_k \Sigma_k^{-\frac{1}{2}}, \quad \tilde{V} = \tilde{L} V_k \Sigma_k^{-\frac{1}{2}},$$

where U_k, $V_k, \in \mathbb{R}^{n_m \times k}$ consist of the corresponding leading k columns of U, $V \in \mathbb{R}^{n_m \times n_m}$, and $\Sigma_k \in \mathbb{R}^{k \times k}$ is the upper left $k \times k$ block of Σ in the SVD

$$\tilde{R}^T \Theta_r^T E_1 \Theta_r \tilde{L} = U \Sigma V^T.$$

Observing further that $R^T \Pi_2 E_1 \Pi_2^T L = \tilde{R}^T \Theta_{2,r}^T E_1 \Theta_{2,r} \tilde{L} = U \Sigma V^T$, the projection matrices for the system (3.9) can be formed as

$$W = R U_k \Sigma_k^{-\frac{1}{2}} \quad \text{and} \quad V = L V_k \Sigma_k^{-\frac{1}{2}}. \tag{3.15}$$

Algorithm 6: LR-SRM for unstable index 2 DAEs.

```
Input  : $E_1,\ A_1,\ B_1,\ C_1$ from (3.1).
Output: $\hat{E},\ \hat{A},\ \hat{B},\ \hat{C}$ in (3.17).
Compute $R$ and $L$ by solving the projected Lyapunov equations (3.14).
Construct $W$ and $V$ as in (3.16)
Form $I = \hat{E} = W^T E_1 V,\ \ \hat{A} = W^T A_1 V,\ \ \hat{B} = W^T B_1$ and $\hat{C} = C_1 V.$
```

As in [70] we find that

$$\begin{aligned} W &= RU_k\Sigma_k^{-\frac{1}{2}} = \Theta_{2,r}\tilde{R}U_k\Sigma_k^{-\frac{1}{2}} = \Theta_{2,r}\tilde{W} = \Theta_{2,r}\Theta_{2,l}^T\Theta_{2,r}\tilde{W} = \Pi_2^T W, \\ V &= LV_k\Sigma_k^{-\frac{1}{2}} = \Theta_{2,r}\tilde{L}V_k\Sigma_k^{-\frac{1}{2}} = \Theta_{2,r}\tilde{V} = \Theta_{2,r}\Theta_{2,l}^T\Theta_{2,r}\tilde{V} = \Pi_2^T V. \end{aligned} \tag{3.16}$$

Now we apply the transformations W and V in (3.9) to find the reduced order model as

$$\begin{aligned} \hat{E}\dot{\hat{v}}(t) &= \hat{A}\hat{v}(t) + \hat{B}u(t) \\ \hat{y}(t) &= \hat{C}\hat{v}(t), \end{aligned} \tag{3.17}$$

where

$$\hat{E} = W^T\Pi_2E_1\Pi_2^TV,\ \ \hat{A} = W^T\Pi_2A_1\Pi_2^TV,\ \ \hat{B} = W^T\Pi_2B_1 \ \text{ and } \ \hat{C} = C_1\Pi_2^TV.$$

Due to (3.16) we can avoid the explicit usage of Π_2 and find

$$I = \hat{E} = W^TE_1V,\ \ \hat{A} = W^TA_1V,\ \ \hat{B} = W^TB_1 \ \text{ and } \ \hat{C} = C_1V.$$

Eventually, we see that the reduced order model (3.17) is obtained without forming the projected system (3.9). In the next section we will show how to compute R and L using a tailored version of the LRCF-ADI iteration without using Π_2 explicitly. The above procedure to compute the ROM for the unstable index 2 DAEs is summarized in Algorithm 6.

3.4 Solution of the projected Lyapunov equations

In order to apply the aforementioned balancing based MOR we need to solve the projected Lyapunov equations (3.14). We have seen above that the Π_2^T invariant solution factors enable us to compute the corresponding truncating transformations. The approach here is different from the spectral projection based work by Stykel in that we are applying the E_1-orthogonal projection to the hidden manifold, where Stykel uses the orthogonal projection (in the euclidean sense) onto the eigenspaces corresponding to the finite poles of the system. In fact both methods project to the same subspace considering orthogonality in different inner products. Here we are concerned with two main issues. First, we discuss the reformulation of the basic low-rank ADI Algorithm for the projected Lyapunov equation that ensures the invariance of the solution factor and the computation of the correct corresponding residual factors. We are lifting the ideas of [70] to the reformulation of the LR-ADI in Algorithm 5. For the spectral projection methods, the analogue procedure has been discussed in [34]. In the second part we address the important issue of ADI shift parameter computation. There the main issue in the DAE setting is to avoid the subspaces corresponding to infinite eigenvalues in order to correctly compute the large magnitude Ritz values involved in many parameter choices. The crucial point in both parts is to provide methods that use the projection Π_2^T at most implicitly and never form the projected system (3.9).

3.4.1 GS-LRCF-ADI for index 2 unstable systems

Here, we are concerned with the efficient solution of the Lyapunov equations in (3.14) to compute the low-rank Gramian factors using LRCF-ADI as discussed in Chapter 2. First, we consider the projected controllability equation elaborately. The observability equation can be handled analogously. For convenience we rewrite the Lyapunov equations (3.14) as

$$\begin{aligned} \tilde{A}\tilde{P}\tilde{E}^T + \tilde{E}\tilde{P}\tilde{A}^T &= -\tilde{B}\tilde{B}^T, \\ \tilde{A}^T\tilde{Q}\tilde{E} + \tilde{E}^T\tilde{Q}\tilde{A} &= -\tilde{C}^T\tilde{C}, \end{aligned} \tag{3.18}$$

with $\tilde{E} = \Pi_2 E_1 \Pi_2^T$, $\tilde{A} = \Pi_2 A_c \Pi_2^T$, $\tilde{B} = \Pi_2 B_1$ and $\tilde{C} = C_1 \Pi_2^T$.

In the i-th iteration step of the ADI the residual of the controllability Lyapunov equation (3.18) can be written as

$$\tilde{\mathcal{F}}(\tilde{P}_i) = \tilde{A}\tilde{P}_i\tilde{E}^T + \tilde{E}\tilde{P}_i\tilde{A}^T + \tilde{B}\tilde{B}^T = \tilde{W}_i\tilde{W}_i^T,$$

where

$$\tilde{W}_i = \prod_{k=1}^{i}(\tilde{A} - \overline{\mu}_i\tilde{E})(\tilde{A} + \mu_i\tilde{E})^{-1}\tilde{B}.$$

To compute the low-rank controllability Gramian factor $\tilde{R}$ we follow Algorithm 5. In the $i-$th iteration step, V_i is computed from

$$(\tilde{A} + \mu_i\tilde{E})V_i = \tilde{W}_{i-1}, \tag{3.19}$$

which enables us to update the residual factor according to

$$\tilde{W}_i = (\tilde{A} - \mu^*\tilde{E})V_i = \tilde{W}_{i-1} - 2\,\mathrm{Re}\,(\mu_i)\tilde{E}V_i. \tag{3.20}$$

In complete analogy to [70, Lemma 5.2], we observe that instead of solving (3.19), one can compute V_i by solving

$$\begin{bmatrix} A_c + \mu_i E_1 & A_2 \\ A_2^T & 0 \end{bmatrix} \begin{bmatrix} V_i \\ \star \end{bmatrix} = \begin{bmatrix} \tilde{W}_{i-1} \\ 0 \end{bmatrix}, \tag{3.21}$$

where the special case $i = 1$, here especially the computation of the initial residual factor $\tilde{W}_0$ is discussed in detail below.

Inserting $A_c = A_1 - B_1 K_c^{fm}$ in (3.21),

$$\begin{bmatrix} A_1 + \mu_i E_1 - B_1 K_c^{fm} & A_2 \\ A_2^T & 0 \end{bmatrix} \begin{bmatrix} V_i \\ \star \end{bmatrix} = \begin{bmatrix} \tilde{W}_{i-1} \\ 0 \end{bmatrix},$$

implies

$$\left(\underbrace{\begin{bmatrix} A_1 + \mu_i E_1 & A_2 \\ A_2^T & 0 \end{bmatrix}}_{\underline{A}} - \underbrace{\begin{bmatrix} B_1 \\ 0 \end{bmatrix}}_{\underline{B}} \underbrace{\begin{bmatrix} K_c^{fm} & 0 \end{bmatrix}}_{\underline{K}} \right) \begin{bmatrix} V_i \\ \star \end{bmatrix} = \begin{bmatrix} \tilde{W}_{i-1} \\ 0 \end{bmatrix}. \tag{3.22}$$

In this equation the inversion of $(\underline{A} - \underline{B}\,\underline{K})$ should in practice be computed using the *Sherman-Morrison-Woodbury formula* (see, e.g. [60]):

$$(\underline{A} - \underline{B}\,\underline{K})^{-1} = \underline{A}^{-1} + \underline{A}^{-1}\underline{B}(I - \underline{K}\,\underline{A}^{-1}\underline{B})^{-1}\underline{K}\,\underline{A}^{-1}, \tag{3.23}$$

to avoid explicit formulation of the (usually dense) matrix $\underline{A} - \underline{B}\,\underline{K}$. In accordance with [70, Lemma 5.2], again the computed V_i in (3.21) satisfies $V_i = \Pi_2^T V_i$. Therefore, the correct projected residual factor in (3.20) can be obtained by

$$\tilde{W}_i = \tilde{W}_{i-1} - 2\,\mathrm{Re}\,(\mu_i) E_1 V_i, \tag{3.24}$$

since we have $\Pi_2 E_1 = E_1 \Pi_2^T$.

In order to really compute the correct residual, the initial residual must be computed as $\tilde{W}_0 = \Pi_2 B_1$ to ensure $\tilde{W}_0 = \Pi_2 \tilde{W}_0$. This can be performed cheaply using the following lemma.

Lemma 3.1. *The matrix* Ξ *satisfies* $\Xi = \Pi_2^T \Xi$ *and* $E_1 \Xi = \Pi_2 B_1 \Leftrightarrow$

$$\begin{bmatrix} E_1 & A_2 \\ A_2^T & 0 \end{bmatrix} \begin{bmatrix} \Xi \\ \Lambda \end{bmatrix} = \begin{bmatrix} B_1 \\ 0 \end{bmatrix}. \tag{3.25}$$

Proof. If $\Xi = \Pi_2^T \Xi$, then $E_1 \Xi = \Pi_2 B_1$ implies $\Pi_2(E_1 \Xi - B_1) = 0$. Since $\mathrm{Null}\,(\Pi_2) = \mathrm{Range}\,(A_2)$, there exists Λ such that $E_1 \Xi - B_1 = -A_2 \Lambda$, or $E_1 \Xi + A_2 \Lambda = B_1$. Again applying the properties in (3.8), we have $A_2^T \Xi = 0$. These two relations give (3.25). Conversely, we assume (3.25) holds. From the first block row of (3.25) we get

$$\Xi = E_1^{-1} B_1 - E_1^{-1} A_2 \Lambda,$$

and thus from the second row it follows that

$$0 = A_2^T \Xi = A_2^T E_1^{-1} B_1 - A_2^T E_1^{-1} A_2 \Lambda,$$

such that

$$\Lambda = (A_2^T E_1^{-1} A_2)^{-1} A_2^T E_1^{-1} B_1.$$

Inserting this in the first block row we get as desired

$$E_1 \Xi = B_1 - (A_2^T E_1^{-1} A_2)^{-1} A_2^T E_1^{-1} B_1 = \Pi_2 B_1.$$

□

This especially ensures $\Xi = \Pi_2^T \Xi$, since

$$E_1 \Xi = \Pi_2 B_1 = \Pi_2 B_1 = \Pi_2 E_1 \Xi = E_1 \Pi_2^T \Xi,$$

and thus using $\tilde{W}_0 = E_1 \Xi$, we get the desired invariance $\tilde{W}_0 = \Pi_2 \tilde{W}_0$.

The above findings on the residual factor can be summarized as the following lemma.

Lemma 3.2. *The residual factor in every step of Algorithm 7 fulfills the relation*

$$\tilde{W}_i = \Pi_2 \tilde{W}_i.$$

The whole procedure of computing the low-rank factor of the controllability Gramian $\tilde{R}$ is summarized in Algorithm 7. Analogous to the derivation in [70], our algorithm computes the correct solution factor. In contrast to the version presented there, we guarantee to compute a real solution factor even if the shifts occur in complex conjugate pairs and we have the low-rank residual factors in hand to evaluate stopping criteria cheaply. Still one issue remains open that has not been tackled in the original paper [70]. The shifts that guarantee fast convergence of the algorithm are closely related to the spectrum of the original pencil. The question how these can be computed is answered in the next section.

3.4.2 ADI shift parameter selection

The appropriate shift parameter selection is one of the crucial tasks for fast convergence of the GS-LRCF-ADI iteration. Recently, most of the papers followed the *heuristic* procedure introduced by Penzl [93] to compute sub-optimal ADI shift parameters μ_i, $i = 1, 2, \ldots, J$, for a large-scale problem. Very recently, new shift computation ideas considering adaptive and automatic computation of shifts during the iteration [21, 130] have come up. We present the basic ideas to adapt both the classic and the new methods to our framework in the following two paragraphs.

Heuristic shift selection. The main ingredient of the heuristic method is the computation of a number of large and small magnitude Ritz values. In the case of DAE systems, the computation of Ritz values of large magnitude is causing difficulties

Algorithm 7: GS-LRCF-ADI for unstable index 2 DAEs.

```
Input  : $E_1$, $A_1$, $A_2$, $B_1$, $K_c^{fm}$, $\{\mu_i\}_{i=1}^{J}$.
Output: $\tilde{R} = Z_i$, such that $\tilde{P} \approx \tilde{R}\tilde{R}^T$.
Set $Z_0 = [\,]$.
Solve the linear system (3.25) for $\Xi$ and compute $\tilde{W}_0 = E_1\Xi$
$i = 1$
while $\|\tilde{W}_{i-1}^T \tilde{W}_{i-1}\| \geq tol$ or $i \leq i_{max}$ do
    Solve the linear system (3.22) for $V_i$.
    if $\mathrm{Im}\,(\mu_i) = 0$ then
        $Z_i = \begin{bmatrix} Z_{i-1} & \sqrt{-2\mu_i}V_i \end{bmatrix}$,
        $\tilde{W}_i = \tilde{W}_{i-1} - 2\mu_i E_1 V_i$
    else
        $\gamma = -2\,\mathrm{Re}\,(\mu_i)$, $\delta = \frac{\mathrm{Re}\,(\mu_i)}{\mathrm{Im}\,(\mu_i)}$,
        $Z_{i+1} = \begin{bmatrix} Z_{i-1} & \sqrt{2\gamma}(\mathrm{Re}\,(V_i) + \delta\,\mathrm{Im}\,(V_i)) & \sqrt{2\gamma}\sqrt{(\delta^2+1)}\,\mathrm{Im}\,(V_i) \end{bmatrix}$,
        $\tilde{W}_i = \tilde{W}_{i-1} + 2\gamma E_1(\overline{V_i} + 2\delta\,\mathrm{Im}\,(V_i))$.
        $i = i + 1$
    end if
    $i = i + 1$
end while
```

due to the existence of infinite eigenvalues. We need to make sure that the infinite eigenvalues are avoided. This can be achieved by the following fact that is a direct consequence of [39, Theorem 3.1].

Corollary 3.1. *The matrix pencil*

$$\mathcal{P}_\delta(\lambda) = \lambda \begin{bmatrix} E_1 & \delta A_2 \\ \delta A_2^T & 0 \end{bmatrix} - \begin{bmatrix} A_1 & A_2 \\ A_2^T & 0 \end{bmatrix} \tag{3.26}$$

transforms all infinite eigenvalues of the pencil $\lambda\check{E} - \check{A}$ to $\frac{1}{\delta}$ while at the same time preserving its finite eigenvalues.

Thus from the pencil $\mathcal{P}_\delta$ we can compute the desired Ritz values of large magnitude via an Arnoldi iteration [100]. The parameter δ can easily be determined after the small Ritz values β_i have been computed with respect to the original pencil. In order to ensure that $\frac{1}{\delta}$ is avoided by the Arnoldi process for the large magnitude Ritz values, and thus only finite eigenvalues of the original pencil are approximated, one could, e.g., set $\delta = \frac{1}{\min\limits_i \mathrm{Re}\,(\beta_i)}$. For the unstable case the corollary obviously has to be applied with A_1 replaced by A_c.

Adaptive shift selection. A second shift computation strategy we use in the numerical experiments follows the lines of the adaptive shift strategy proposed in [21].

There, the shifts are initialized by the eigenvalues of the pencil projected to the span of W_0. Then, whenever all shifts in the current set have been used, the pencil is projected, e.g., to the span of the current V_i and the eigenvalues are used as the next set of shifts. Here, we use the same initialization. For the update step, however, we extend the subspace to all the V_i generated with the current set of shifts and then choose the next shifts following Penzl's heuristic with the Ritz values replaced by the projected eigenvalues computed with respect to this larger subspace. Note that in lack of the conditions for Bendixon's theorem, we cannot guarantee that the projected eigenvalues will be contained in $\mathbb{C}_-$. Should any of them end up in the wrong half-plane, we neglect them. Note further that the problem with the infinite eigenvalues does not exist in this case. Since we have $\Pi_2^T Z = Z$, for any orthogonal basis U of a set of columns of Z, we also have $\Pi_2^T U = U$. As an immediate result of this fact, the projected pencil $(U^T A_1 U - \lambda U^T E_1 U)$ automatically resides on the hidden manifold and can thus only has finite eigenvalues.

3.5 Riccati-based feedback stabilization from ROM

Stabilization of the non-stationary incompressible Navier-Stokes equations around a stationary solution using a Riccati-based feedback has received considerable attention regarding theory as well as numerical methods during the recent years. In the Riccati-based boundary feedback stabilization technique [12], the most challenging task is to solve the corresponding algebraic Riccati equation (ARE) for the full dimensional model. The key problem in the LQR approach for the model under investigation is to compute the boundary feedback stabilization matrix K_f (see e.g., [12]), such that the stabilized system has the following form:

$$\begin{aligned} E_1\dot{v}(t) &= (A_1 - B_1K_f)v(t) + A_2p(t) + B_1u(t), \\ A_2^T v(t) &= 0. \end{aligned} \tag{3.27}$$

The consequence of the feedback stabilization matrix K_f for Navier-Stokes equations with Re 300 is shown in Figure 3.2 from [12]. In this figure the vertical component of the velocity is shown by red, as maximal value downwards, and white, as maximal value upwards. In the top picture of this figure no feedback stabilization is imposed. The occurring vortexes are shown by the red and white areas that move away from the obstacle in an alternating order. The middle picture shows the consequence of the initial feedback. And the third picture shows that when the optimized feedback matrix K_f is inserted the vertical components vanish very soon. The authors in [12] presented an algorithm (see [12, Algorithm 2]) to compute K_f which is based on the standard linear-quadratic regulator approach [109, 42] for a projected semidiscretized state-space system. The most challenging part in this algorithm is to solve the usually very large, generalized, projected algebraic Riccati equation (GARE) based on the full order semidiscretized model. We employ the reduced-order model (3.17) to compute an approximation to the optimal LQR

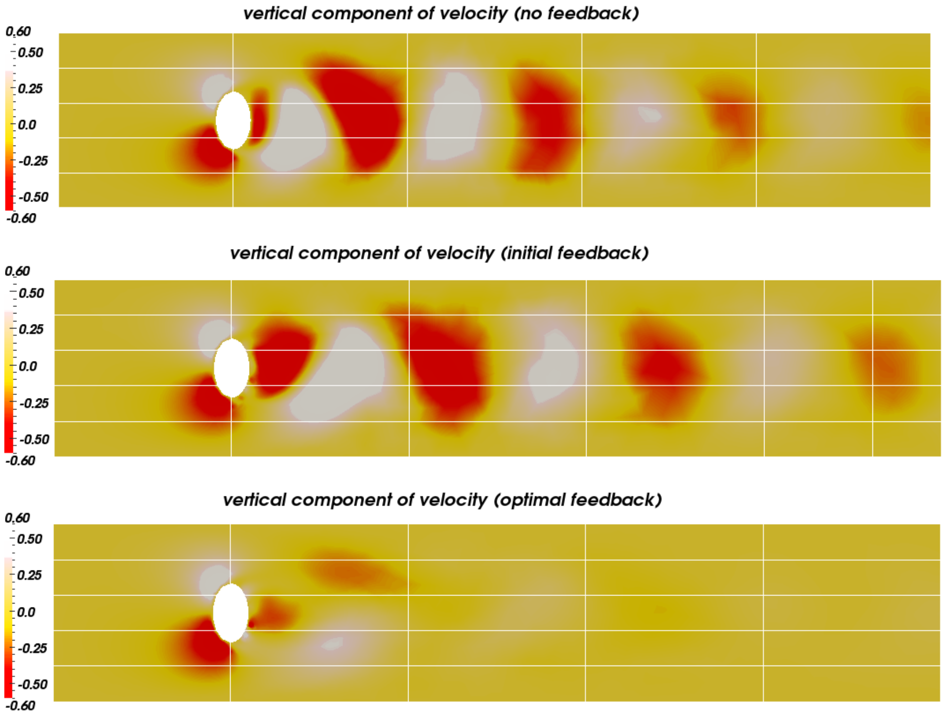


Figure 3.2: Flow field for Re=300 (source [12]).

feedback matrix of the full system. The main advantage of this approach is that we only need to solve two projected algebraic Lyapunov equations in order to derive the reduced-order model instead of one Lyapunov equation per Newton step in the solver for the GARE, which are usually many more [34].

Based on the reduced model (3.17) the GARE

$$\hat{A}^T\hat{X} + \hat{X}\hat{A} - \hat{X}\hat{B}\hat{B}^T\hat{X} = -\hat{C}^T\hat{C} \tag{3.28}$$

is now much smaller in dimension. It can thus easily be solved for $\hat{X}$ using classical solvers as, e.g., the MATLAB `care` command. The stabilizing feedback matrix for the reduced model (3.17) then is

$$\hat{K}_f = \hat{B}^T\hat{X}.$$

The ROM-based approximation to the SFM for the full order model can now be retrieved as

$$K_f = \hat{B}^T\hat{X}W^TE_1 = \hat{K}_fW^TE_1 \tag{3.29}$$

where W is the left balancing and truncating transformation (see Section 3.3) used to compute the reduced-order model.

Name of model	n_1	n_2
Mod-1	3 142	453
Mod-2	8 268	1 123
Mod-3	19 770	2 615
Mod-4	44 744	5 783
Mod-5	98 054	12 566

Table 3.1: The number of differential and algebraic variables of different discretization levels of the model.

model	heuristic shift			adaptive shift		
	iterations		time (sec)	iterations		time (sec)
	$\tilde{R}$	$\tilde{L}$	$\tilde{R}+\tilde{L}+\mu$	$\tilde{R}$	$\tilde{L}$	$\tilde{R}+\tilde{L}$
Mod-1	240	210	67	116	88	25
Mod-2	170	133	165	106	77	87
Mod-3	257	182	625	114	99	305
Mod-4	307	196	1 922	146	111	1 063
Mod-5	368	238	5 839	147	120	2 551

Table 3.2: The performances of the heuristic and adaptive shifts in the GS-LRCF-ADI iteration.

3.6 Numerical results

3.6.1 Test examples and hardware

To assess the performance of the techniques, this section discusses some numerical tests. The method is applied to a set of linearized semi-discretized Navier-Stokes equations as described in Section 3.1. All the computations were carried out using MATLAB® 7.11.0 (R2010b) on a board with 2 Intel® Xeon® X5650 CPUs with a 2.67-GHz clock speed.

The authors of [12] generate different sized models using Reynolds number $R_e = 500$. Table 3.1 shows the different sizes of the model and distinguishes the dimensions n_1 of the velocity vector (differential variable) and n_2 of the pressure vector (algebraic variable). In all the sets, $B_1 \in \mathbb{R}^{n_1 \times 2}$ and $C_1 \in \mathbb{R}^{7 \times n_1}$. For Reynolds numbers of 400 and more the described linearized model is unstable. Thus, especially the Reynolds number 500 case discussed here is unstable. The Bernoulli stabilizing feedback matrices K_c^{fm} and K_o^{fm} for all models are computed applying the procedure from [4] and [12, Section 2]. It uses 2 calls of the MATLAB `eigs` implementation of the Arnoldi method to compute the rightmost eigenvalues together with their left and right eigenvectors.

Models	tolerance	system dimension	
		full	reduced
Mod-1		3 595	145
Mod-2		9 391	147
Mod-3	10^{-5}	22 385	163
Mod-4		50 527	178
Mod-5		110 620	184

Table 3.3: Dimensions of original and reduced systems of the different sizes models for a fixed balanced truncation tolerance.

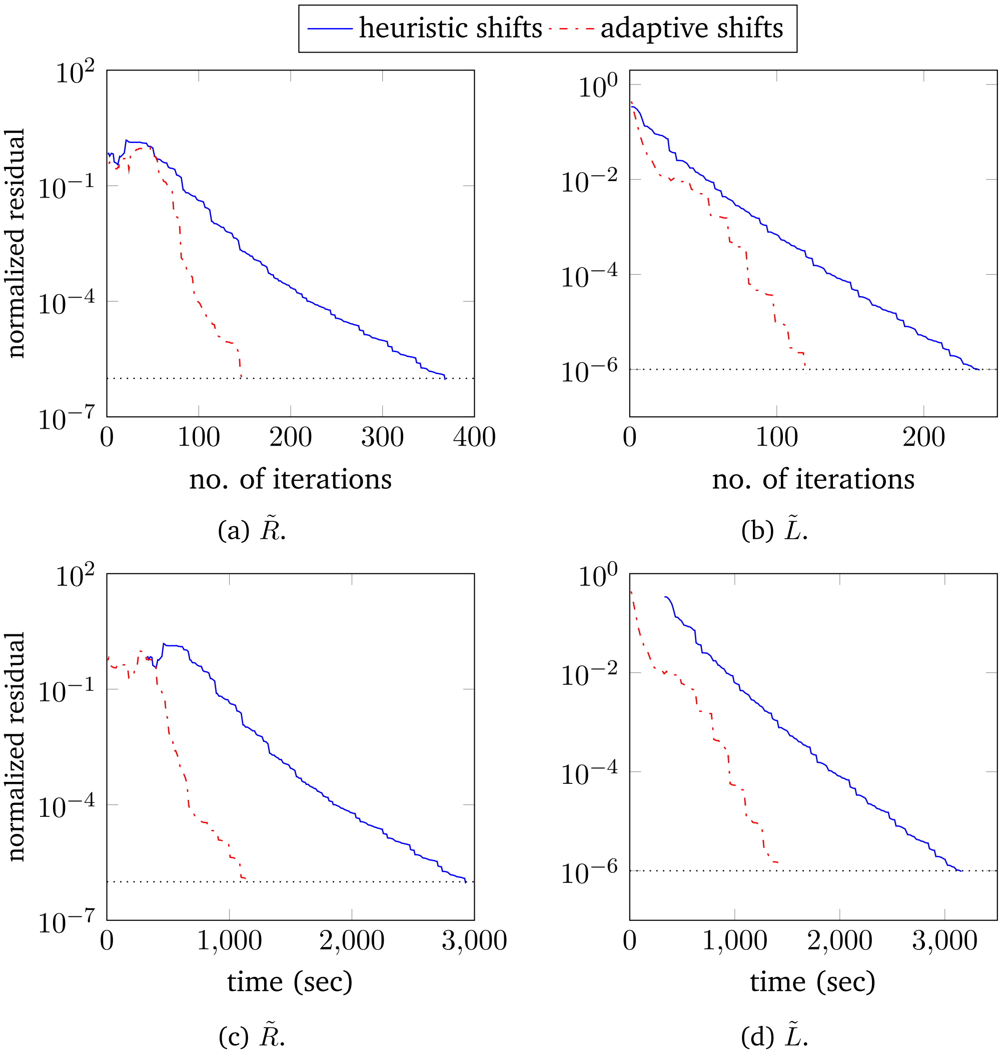
Name of model	tolerance	dimension of ROM
Mod-5	10^{-4}	161
	10^{-3}	138
	10^{-2}	115
	10^{-1}	93

Table 3.4: Balanced truncation tolerances and dimensions of reduced models.

3.6.2 GS-LRCF-ADI and balancing based MOR

We apply the GS-LRCF-ADI iteration (Algorithm 7) to all aforementioned models to compute the low-rank factors $\tilde{R}$ and $\tilde{L}$ considering the tolerance 10^{-6}. We investigate the performances of both the *heuristic* and *adaptive* shifts to implement this algorithm. The results are shown in Table 3.2. For all models we chose 30 optimal heuristic shifts out of 10 large and 80 small magnitude Ritz-values. In the case of the adaptive shifts, in each cycle, 10 proper shift parameters are selected following the procedure discussed above. For computing the initial shifts, first we project the pencil $(A_1 - \lambda E_1)$, onto the column space of a $n_1 \times 100$ random matrix. For all the models the performance of the adaptive shifts is much better than the heuristic shifts. The performances of the heuristic and adaptive shifts are also depicted in Figure 3.3 for the largest dimensional system Mod-5. This figure shows the convergence of the norms of the low-rank controllability and observability Gramian factors with respect to iterations (Figures 3.3a, 3.3b) and time (Figures 3.3c, 3.3d). In both cases the convergence for the adaptive shifts is much faster than for the heuristic shifts. Note that we use the Frobenius norm to compute the residual norm.

We apply Algorithm 6 for all data sets and compute their reduced order models. The dimensions of the original and reduced models are shown in Table 3.3. If nothing else is stated, the truncation tolerance is set to 10^{-5}. The dimension of the ROM can however be decreased by increasing the tolerance if desired or required. This is shown in Table 3.4. Since the numerical results are all comparable we exemplary present only selected plots.



(a) $\tilde{R}$.

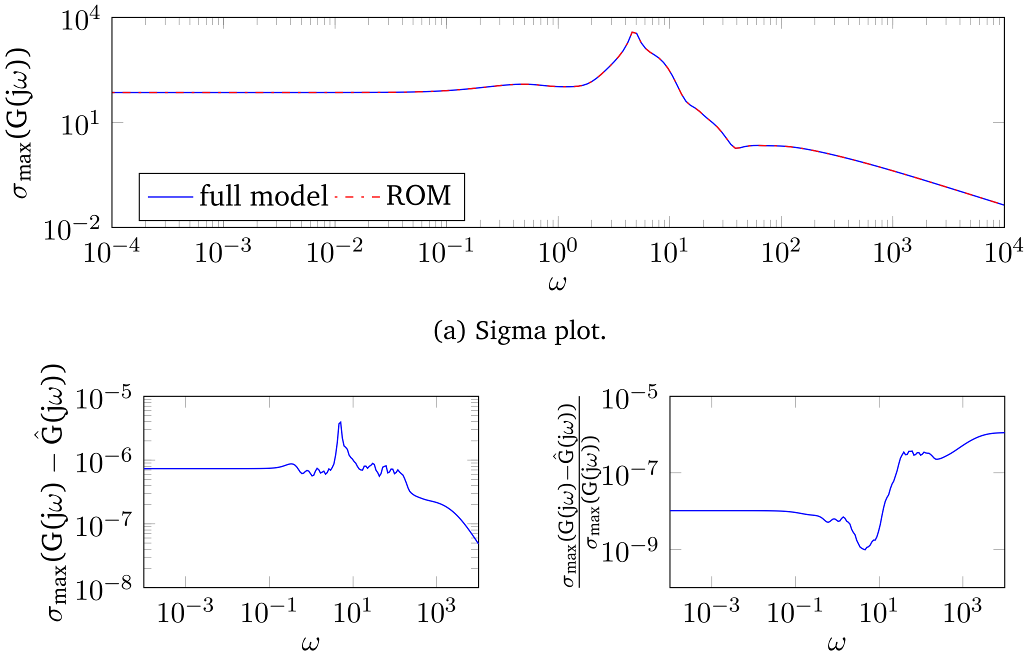
(b) $\tilde{L}$.

(c) $\tilde{R}$.

(d) $\tilde{L}$.

Figure 3.3: Comparisons of the heuristic and adaptive shifts in computing the low-rank Gramian factors using the GS-LRCF-ADI iteration.

Model reduction of the unstable system: Here we review the numerical experiments for the unstable case. We present both frequency and time domain error analyses. The frequency domain error analysis is shown in Figure 3.4. In Figure 3.4a we see the frequency responses of the full and 184 dimensional reduced-order models for Mod-5 with a nice match in the eyeball norm. The absolute and relative deviations between full and reduced-order models are shown in Figures 3.4b and 3.4c. Here, we can see that the absolute error is bounded by the prescribed truncation tolerance of 10^{-5}. For higher frequencies the relative error is slightly increasing since the frequency response is decreasing more rapidly than the absolute error can. Figure 3.5 depicts time domain simulation of full and reduced-order models for Mod-5. This figure shows the step responses from Input 1 to Output 1 together with their absolute deviations. To compute the step response we use an implicit



(a) Sigma plot.

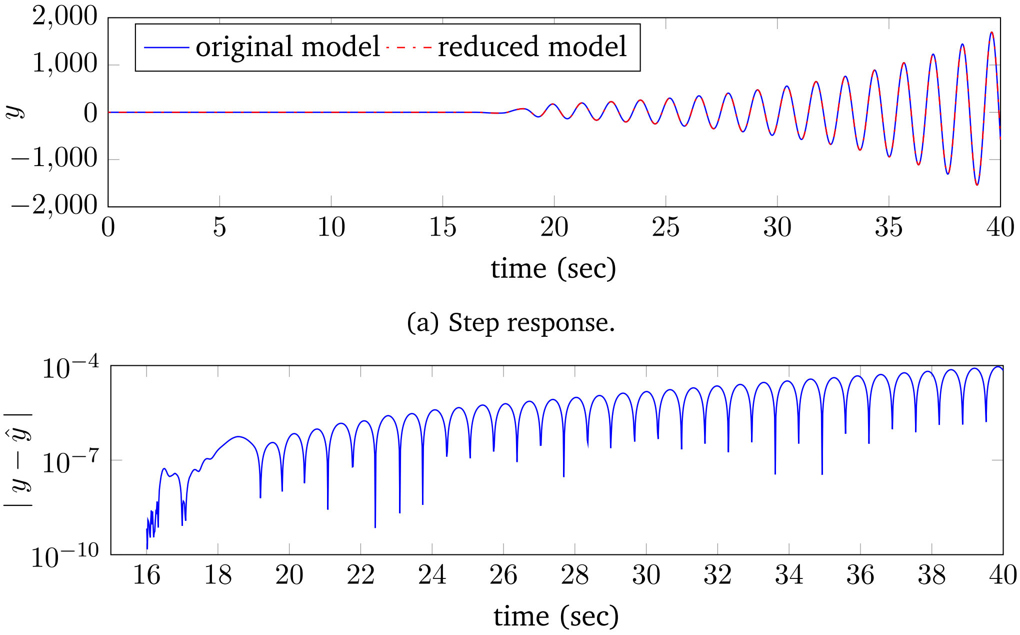
(b) Absolute error.

(c) Relative error.

Figure 3.4: Comparison of the full and reduced models in frequency domain.

Euler method with fixed time step size 10^{-2}. Initially, the imposed control is kept inactive, therefore the responses for both (full and reduced) models are constant within the range 0 to 15s. Switching the control to constant unit actuation on Input 1, the responses are oscillating with increasing amplitude in the higher time domain caused by the instability of the model. Here we also see the issue with the balanced truncation error bound for unstable systems since the absolute error is increasing gradually with increasing time.

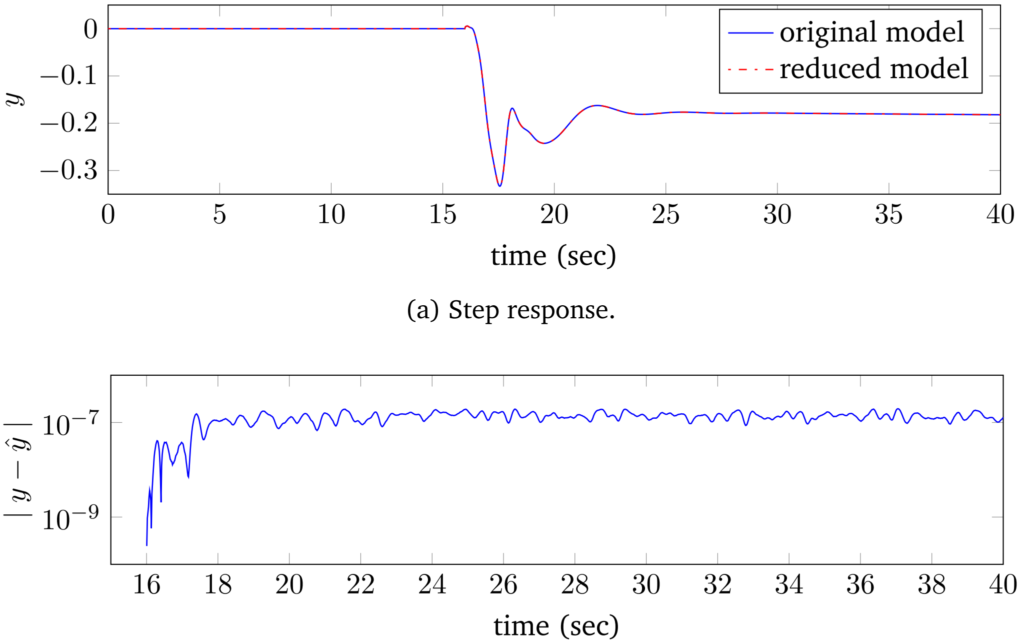
Numerical Experiments for the stabilized system: In Section 3.5 we mentioned that the stabilizing feedback matrix for the full model can be computed from the reduced order model. To this end, we solve the corresponding algebraic Riccati equation for the reduced order model (3.17) arising from the linear quadratic regulator approach using the MATLAB `care` command and compute the optimal stabilizing feedback matrix $\hat{K}_f$ [32]. The ROM based approximation to the stabilizing feedback matrix for the full order model (3.1) is then computed by (3.29). Figure 3.6 shows the step response (from 1st input to 1st output) of closed loop full and reduced order models and their absolute error. For the generation of the step response, the same procedure has been followed as for the unstable case above. Note that for a stabilizing feedback, the step response system has to be viewed as that of an asymptotically stable system with a constant source term. Thus the outputs stabilize at constant nonzero values.



(a) Step response.

(b) Absolute error.

Figure 3.5: Step responses of 1st input to 1st output of full and reduced-order models and respective absolute deviations.



(a) Step response.

(b) Absolute error.

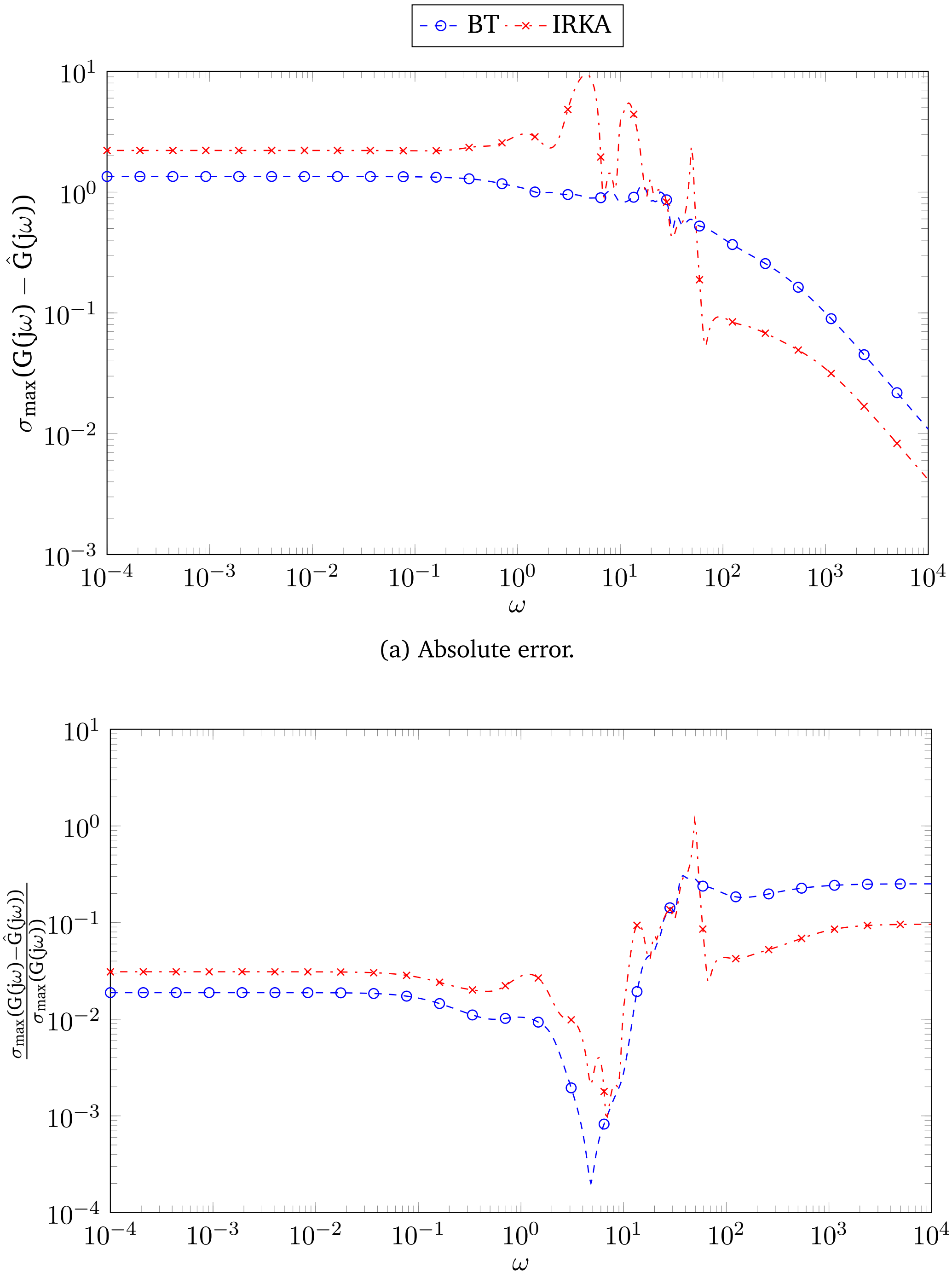
Figure 3.6: Step responses of 1st input to 1st output of stabilized full and reduced-order models and respective absolute deviations.

model	CPU time (sec)		relative error H_∞ norm	
	BT	IRKA	BT	IRKA
Mod-1	23.90	41.57	1.71×10^{-1}	1.88×10^{-1}
Mod-2	75.35	150.79	1.51×10^{-1}	4.20×10^{-1}
Mod-3	280.10	411.35	1.04×10^{-1}	2.30×10^{-1}
Mod-4	996.92	1 221.22	3.38×10^{-1}	4.09×10^{-1}
Mod-5	2 311.28	3 083.79	3.03×10^{-1}	8.45×10^{-1}

Table 3.5: Comparisons of balanced truncation and IRKA for different sized models and their 50 dimensional reduced models.

3.6.3 Comparison of BT with interpolatory technique

Table 3.5 describes the comparison of the BT and interpolatory based model reduction methods. Here we compute 50 dimensional reduced models for all the model examples mentioned above, applying both balanced truncation and the interpolatory method via IRKA. For the interpolatory based approach we exactly follow [68, Algorithm 4.1]. As we can see in this table, for all model examples the balancing based method performs better than IRKA considering both relative error and computational time. To find the relative error, we divide the norm of the error system the corresponding norm of the full order system. If we consider a ROM that is larger than 50, then IRKA becomes even more expensive in contrast to balanced truncation. This is due to the fact, that in the BT the only expensive part is computing the low-rank Gramian factors. Once they are computed we can construct a reduced model of any dimension. Note that, for computing the low-rank Gramian factors, Algorithm 7 is stopped by the tolerance 10^{-5}. The quality of the IRKA based reduced model also depends on the number of cycles (i.e., how many times the interpolation points are updated). Although taking more cycles ensures a better reduced model, it gets more expensive. Here to construct the ROMs, for all model examples we restricted to 10 cycles. Figure 3.7 shows the error comparisons of the sigma plots (as in Figure 3.4a) between full system (Mod-5) and 50 dimensional reduced systems. From this figure, one can notice that in the higher frequency range, the interpolatory method performs better than balanced truncation.



(a) Absolute error.

(b) Relative error.

Figure 3.7: Errors between the full and 50 dimensional reduced systems computed by BT and IRKA using the system Mod-5.

Chapter 4

Second Order Index 1 Descriptor Systems

In this chapter we consider second order index 1 descriptor systems of the form

$$\underbrace{\begin{bmatrix} M_{11} & 0 \\ 0 & 0 \end{bmatrix}}_{\check{M}} \begin{bmatrix} \ddot{\xi}(t) \\ \ddot{\varphi}(t) \end{bmatrix} + \underbrace{\begin{bmatrix} D_{11} & 0 \\ 0 & 0 \end{bmatrix}}_{\check{D}} \begin{bmatrix} \dot{\xi}(t) \\ \dot{\varphi}(t) \end{bmatrix} + \underbrace{\begin{bmatrix} K_{11} & K_{12} \\ K_{12}^T & K_{22} \end{bmatrix}}_{\check{K}} \begin{bmatrix} \xi(t) \\ \varphi(t) \end{bmatrix} = \underbrace{\begin{bmatrix} H_1 \\ H_2 \end{bmatrix}}_{\check{H}} u(t), \tag{4.1a}$$

$$\underbrace{\begin{bmatrix} H_1^T & H_2^T \end{bmatrix}}_{\check{H}^T} \begin{bmatrix} \xi(t) \\ \varphi(t) \end{bmatrix} = y(t), \tag{4.1b}$$

where $\xi(t) \in \mathbb{R}^{n_\xi}$, $\varphi(t) \in \mathbb{R}^{n_\varphi}$ are the states, $n_\xi > n_\varphi$, $u(t) \in \mathbb{R}^m$ are control inputs and the measurement outputs are $y(t) \in \mathbb{R}^p$, and the matrices $\check{M}$, $\check{D}$, $\check{K}$ are sparse. We assume the block matrix K_{22} to be nonsingular. We call (4.1) an index 1 system due to the analogy to first order index 1 (see, e.g., section 4.2) linear time-invariant (LTI) systems [121]. Such dynamical systems usually arise in different branches of engineering such as mechanics [48], where an extra constraint is imposed in order to control the dynamic behavior of the systems, or mechatronics where mechanical and electrical components are coupled with each other. In the specific case of the model example we use in the numerical experiments, the index 1 character results from the multiphysics application with very different timescales (see, e.g., Section 4.1). This allows to treat one variable by a stationary analysis, while the other is covered fully dynamic.

Since the block matrix K_{22} is invertible, from the second line of (4.1a) we obtain

$$\varphi(t) = -K_{22}^{-1} K_{12}^T \xi(t) + K_{22}^{-1} H_2 u(t).$$

Insert this identity into the first line of (4.1a) and (4.1b). The index 1 system (4.1) is then transformed into an equivalent ODE system

$$\begin{aligned} M\ddot{\xi}(t)+D\dot{\xi}(t)+K\xi(t)&=Hu(t),\\ y(t)&=H^Tx(t)+D_au(t), \end{aligned} \tag{4.2}$$

where

$$\begin{aligned} &M= \;\; M_{11},\\ &D=D_{11},\;\; K=K_{11}-K_{12}K_{22}^{-1}K_{12}^T,\\ &H=H_1-K_{12}K_{22}^{-1}H_2,\;\; D_a=H_2^TK_{22}^{-1}H_2. \end{aligned} \tag{4.3}$$

In principle, we can apply the model reduction techniques to the system (4.2) following the approaches discussed in Chapter 2. In this case the matrix K is usually dense and causes infeasible computational complexity. Moreover, for a large-scale system with a large K_{22} block (e.g., the system that we consider for the numerical experiments), due to memory restriction forming (4.2) is simply impossible. This chapter discusses how to perform the model reduction for the DAEs (4.1) without forming the ODEs (4.2) explicitly. Here we show both second order index 1 to first order and second-order-to-second-order reduction techniques. In our earlier work (see, e.g., [120]) we have developed a balancing based algorithm to obtain a first order reduced system from the second order index 1 DAE system. In contrast to that work here we present a more efficient algorithm by exploiting the symmetry properties of the system and using all recent updates in the low-rank version of the ADI method. In addition, we develop the interpolatory model reduction method via IRKA for such systems and compare the results with the balancing based method. One of the major contributions of this chapter is the structure preserving model reduction for the second order index 1 descriptor system (4). In this case, first we also discuss the balancing based method. Besides this, we show that a second order reduced model can be obtain efficiently via projecting the systems onto the dominant eigenspace of the *second order systems Gramians*. Here this technique is called PDEG method. This method was originated in [80, 79, 94] for the model reduction of a standard state space system. The PDEG method is computationally cheaper than the balanced truncation. Moreover, in general, this method preserves some important properties such as stability, symmetry etc., of the original system. For the BT and PDEG based reduction methods, the main expensive part is to compute the low-rank Gramian factors by solving the Lyapunov equations. This chapter discusses the efficient techniques to solve Lyapunov equations for the model (4.1) using the LRCF-ADI method. To ensure fast convergence of the LRCF-ADI iteration, we show the automatic shift generation technique inside the algorithm. The proposed techniques are applied to a piezo-actuated structural FEM model of a certain building block of a parallel kinematic machine tool. Numerical results illustrate the efficiency of the methods.

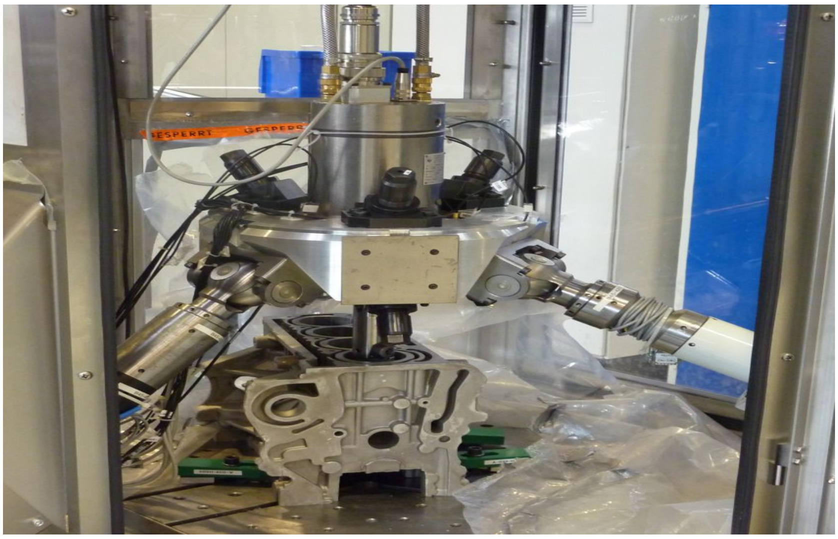


Figure 4.1: Piezo-actuator based mechanical system

4.1 Motivating example

Piezo-actuator based adaptive spindle support (ASS) is an important component [45, 90] of the mechanical system shown in Figure 4.2. The ASS is designed as an independent sub-component of the test machine. First the purpose of the ASS was to gain additional positioning freedom during machining operations (see, [45] for details). Now the concept is enhanced and specialized for non-circular drilling and microstructuring of surfaces. Based on the engineering design with a differential setup of the piezo stack actuators, the suitability for a special application is mainly defined by the applied control concept. Before implementation into the real machine, system simulation is needed to design and test the control concept.

Applying the finite element method (FEM) to the ASS as shown in Figure 4.3, a mathematical model is formed which has the following form:

$$\begin{aligned} \check{M}\ddot{\check{z}}(t) + \check{D}\dot{\check{z}}(t) + \check{K}\check{z}(t) &= \check{H}u(t), \\ y(t) &= \check{H}^T\check{z}(t), \end{aligned} \tag{4.4}$$

where $\check{z}(t)$ consists of the mechanical displacements $\xi(t)$ and the electric potentials $\varphi(t)$. Separating the mechanical and electrical parts and defining $\check{z}(t) = \begin{bmatrix}\xi(t)^T & \varphi(t)^T\end{bmatrix}^T$, (4.4) results in (4.1). The block matrices M_{11}, D_{11} and K_{11} are the mechanical mass, damping and stiffness matrices. The matrix K is composed of the mechanical (K_{11}), electrical (K_{22}) and coupling (K_{12}) terms. Selected general force quantities (mechanical forces and electrical charges) are chosen as the input quantities u, and the corresponding general displacements (mechanical displacements and electrical potential) are the output quantities y. The total mass matrix

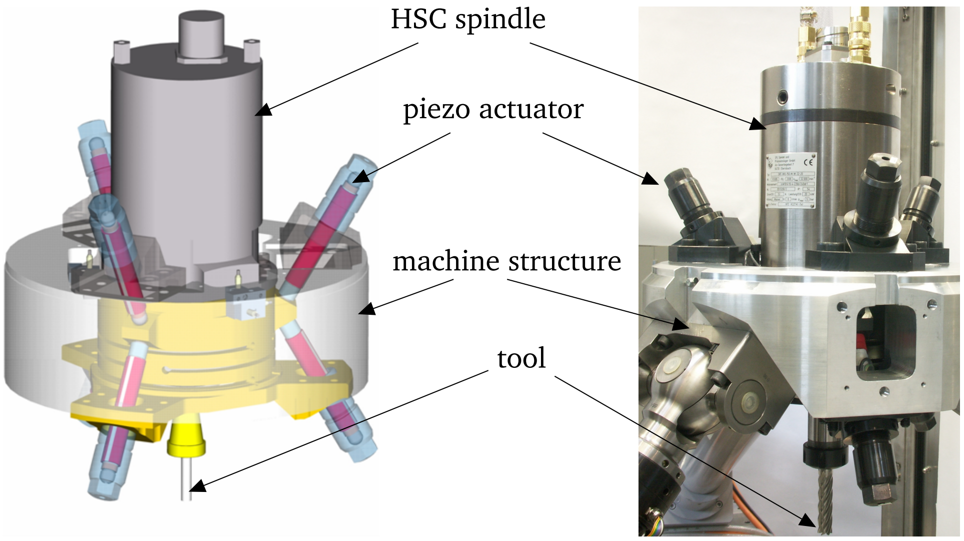


Figure 4.2: The adaptive spindle support: CAD-model (left) and real component mounted on the test bench (right).

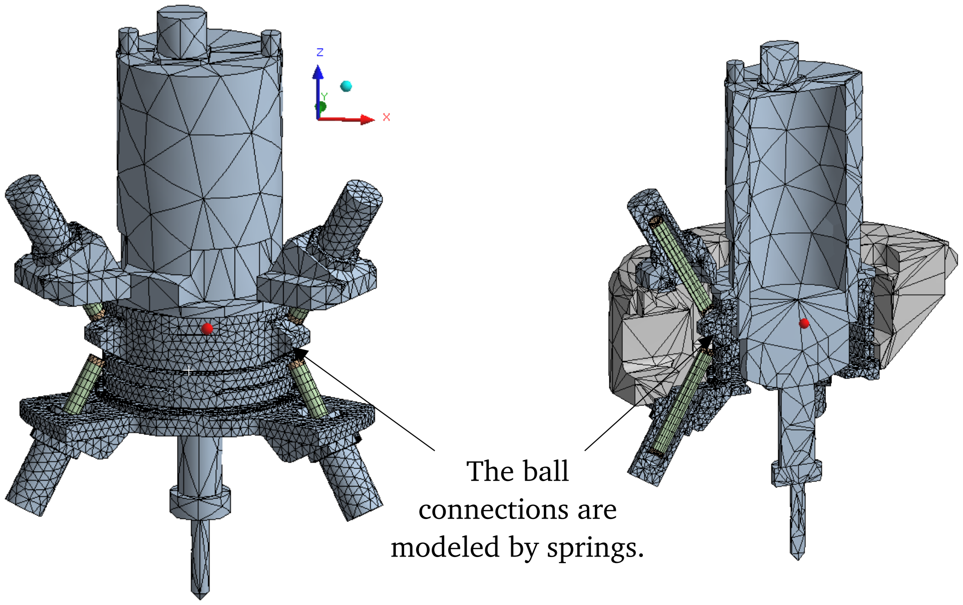


Figure 4.3: Detail of the finite element mesh of the adaptive spindle support.

contains zeros at the locations of the electrical potential. More precisely, the electrical potential of the system (degrees of freedom (DoF) for the electrical part) is not associated with an inertia. The equation of motion of the system in (4.1) can be found in [91]. This equation results from a finite element discretization of the balance equations. For piezo-mechanical systems, these are the mechanical balance of momentum (with inertia term) and the electro-static balance. From this, the electrical potential without inertia term is obtained. Thus, for the whole system (mechanical and electrical DoFs) the mass matrix has rank deficiency.

The simulation with the full finite element model is not suitable due to the large number of degrees of freedom. Therefore our first aim is to obtain a considerably reduced state space model in order to facilitate fast simulation using MATLAB-Simulink ® . Second, we want to obtain a reduced second order model that will reflect the physical structure of the original model to perform the simulation work using special software, e.g., in flexible multibody simulation (if it is necessary).

4.2 Second-order-to-first-order reduction techniques

Although there exists a variety of transformations of (4.1) into first order form the formulation

$$\begin{aligned}\begin{bmatrix} 0 & M_{11} & 0 \\ M_{11} & D_{11} & 0 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} \ddot{\xi}(t) \\ \dot{\xi}(t) \\ \dot{\varphi}(t) \end{bmatrix} &= \begin{bmatrix} M_{11} & 0 & 0 \\ 0 & -K_{11} & -K_{12} \\ 0 & -K_{12}^T & -K_{22} \end{bmatrix}\begin{bmatrix} \dot{\xi}(t) \\ \xi(t) \\ \varphi(t) \end{bmatrix} + \begin{bmatrix} 0 \\ H_1 \\ H_2 \end{bmatrix} u(t), \\ y(t) = \begin{bmatrix} 0 & H_1^T & H_2^T \end{bmatrix} & \begin{bmatrix} \dot{\xi}(t) \\ \xi(t) \\ \varphi(t) \end{bmatrix},\end{aligned} \tag{4.5}$$

is ideally suited, since this representation has only symmetric matrices and the output matrix is the transpose of the input matrix. These are exactly the properties we exploit for efficient computations in the context of model reduction. The system (4.5) is now in first order index 1 form (as defined in Chapter 2), which can be again written as

$$\begin{aligned}\underbrace{\begin{bmatrix} E_1 & 0 \\ 0 & 0 \end{bmatrix}}_{\check{E}}\begin{bmatrix} \dot{z}(t) \\ \dot{\varphi}(t) \end{bmatrix} &= \underbrace{\begin{bmatrix} A_1 & A_2 \\ A_2^T & A_4 \end{bmatrix}}_{\check{A}}\begin{bmatrix} z(t) \\ \varphi(t) \end{bmatrix}\underbrace{\begin{bmatrix} B_1 \\ B_2 \end{bmatrix}}_{\check{B}} u(t), \\ y(t) &= \underbrace{\begin{bmatrix} B_1^T & B_2^T \end{bmatrix}}_{\check{C}}\begin{bmatrix} z(t) \\ \varphi(t) \end{bmatrix},\end{aligned} \tag{4.6}$$

where

$$\begin{aligned} E_1 &:= \begin{bmatrix} 0 & M_{11} \\ M_{11} & D_{11} \end{bmatrix}, \quad A_1 := \begin{bmatrix} M_{11} & 0 \\ 0 & -K_{11} \end{bmatrix}, \quad A_2 := \begin{bmatrix} 0 \\ -K_{12} \end{bmatrix}, \\ A_4 &= -K_{22}, \quad B_1 := \begin{bmatrix} 0 \\ H_1 \end{bmatrix}, \quad B_2 := H_2, \quad z(t) := \begin{bmatrix} \dot{\xi}(t) \\ \xi(t)) \end{bmatrix}. \end{aligned} \tag{4.7}$$

The authors in [53] show a balancing based model reduction method for first order structured index 1 descriptor systems. Following the approaches in [53], since the sub-matrix A_4 is nonsingular, we can put the system (4.6) into a compact form

$$E\dot{z}(t) = Az(t) + Bu(t), \quad y(t) = B^T z(t) + D_a u(t), \tag{4.8}$$

with

$$E = \ E_1, \quad A = A_1 - A_2 A_4^{-1} A_2^T, \quad B = B_1 - A_2 A_4^{-1} B_2,$$

where $E,\ A \in \mathbb{R}^{2n_\xi \times 2n_\xi}$, $B \in \mathbb{R}^{2n_\xi \times m}$. The algebraic part of the system (4.6) has been removed in (4.8). Hence, one can apply the standard model reduction techniques (e.g., balanced truncation and interpolatory methods) to the system (4.8). Again note that the matrix A is typically dense which increases the computational cost and memory requirements in the implementation. Therefore, we are forbidden to convert the system (4.6) into (4.8) explicitly. In the following we discuss efficient BT and interpolatory methods for the model reduction of the system (4.6) avoiding the explicit formulation of the system (4.8).

4.2.1 Balancing based method

In Chapter 2 we already have discussed that to perform the balancing based model reduction (e.g., using Algorithm 1) one has to compute the controllability and observability Gramian factors by solving the controllability and observability Lyapunov equations as in (2.12) and (2.13). We know that solving the Lyapunov equations is the most expensive task in balanced truncation. If we consider the system (4.8) due to the symmetric form (i.e., $E = E^T$ and $A = A^T$) and the input-output matrices are the transpose of each other, the controllability and observability Lyapunov equations coincide. That means the systems controllability and the observability Gramians are identical and hence, we need to solve only one Lyapunov equation

$$APE + EPA = -BB^T, \tag{4.9}$$

where $P \in \mathbb{R}^{2n_\xi \times 2n_\xi}$ denotes either controllability or observability Gramian of the system. By applying the LRCF-ADI iteration discussed in Chapter 2 we can compute the low-rank approximate (controllability or observability) Gramian factor Z, which satisfies

$$ZZ^T \approx P. \tag{4.10}$$

Algorithm 8: LR-SRM for second order index 1 systems.

```
Input  : $M_{11},\ D_{11},\ K_{11},\ K_{12},\ K_{22},\ H_1,\ H_2.$
Output: $\hat{E},\ \hat{A},\ \hat{B},\ \hat{D}_a := D_a.$
1 Form $E_1,\ A_1,\ A_2,\ A_4,\ B_1,\ B_2$ as in (4.7) and $D_a = -B_2^T A_4^{-1} B_2$.
2 Compute $Z$ by solving the Lyapunov equation (4.9).
3 Construct $V$ by performing (4.11) - (4.12).
4 Form the reduced matrices $\hat{E},\ \hat{A}$ and $\hat{B}$ as in (4.14)
```

Section 4.4 will detail how to compute the low-rank Gramian factor by solving the Lyapunov equation (4.9) efficiently. It can be shown that if the two Gramians are equal, then the left and right balancing and truncating transformations i.e, V and W as defined in Algorithm 1, are the same. Once we have the Gramian factor Z, the balancing and truncating transformation can be formed by computing the SVD

$$ZE_1Z^T = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & \\ & \Sigma_2 \end{bmatrix} \begin{bmatrix} U_1^T \\ U_2^T \end{bmatrix}, \tag{4.11}$$

and defining

$$V = W := ZU_1\Sigma_1^{-\frac{1}{2}}. \tag{4.12}$$

The reduced system

$$\begin{aligned} \hat{E}\dot{\hat{z}}(t) &= \hat{A}\hat{z}(t) + \hat{B}u(t), \\ \hat{y}(t) &= \hat{B}^T\hat{z}(t) + \hat{D}_a u(t), \end{aligned} \tag{4.13}$$

is obtained by constructing the reduced matrices as

$$\begin{aligned} &\hat{E} = \ V^T E V, \\ &\hat{A}_1 = V^T A_1 V, \quad \hat{A}_2 = V^T A_2, \quad \hat{B}_1 = V^T B_1, \\ &\hat{A} = \hat{A}_1 - \hat{A}_2^T A_4^{-1} \hat{A}_2, \quad \hat{B} = \hat{B}_1 - \hat{A}_2 A_4^{-1} B_2, \quad \hat{D}_a := D_a. \end{aligned} \tag{4.14}$$

The whole procedure to obtain the reduced ODE system (4.13), for a given second order index 1 system (4.1) is shown in Algorithm 8. However, we represent (4.13) in the reduced index 1 DAE setting as

$$\begin{aligned} \begin{bmatrix} \hat{E} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\hat{z}}(t) \\ \dot{\varphi}(t) \end{bmatrix} &= \begin{bmatrix} \hat{A}_1 & \hat{A}_2 \\ \hat{A}_2^T & A_4 \end{bmatrix} \begin{bmatrix} \hat{z}(t) \\ \varphi(t) \end{bmatrix} \begin{bmatrix} \hat{B}_1 \\ B_2 \end{bmatrix} u(t), \\ y(t) &= \begin{bmatrix} \hat{B}_1^T & B_2^T \end{bmatrix} \begin{bmatrix} \dot{\hat{z}}(t) \\ \hat{\varphi}(t) \end{bmatrix}. \end{aligned} \tag{4.15}$$

Note the reduced system (4.15) is not very useful if the block matrix A_4 is large. Because in that case the reduced model is still large.

4.2.2 Interpolatory method

Here we discuss the model reduction technique for the second order index 1 descriptor system (4.1) by applying the interpolatory method via IRKA. Such work has not been done yet to refer here. We can start with the same procedure as it is discussed for the balanced truncation. That means first convert the second order DAEs (4.1) into the first order form (4.6), and then to the generalized state space form (4.8). When the ODE in (4.8) has been formed, we can immediately follow Algorithm 2 to construct the ROM for the second order index 1 system. Again note that explicit formulation of (4.8) is prohibitive due to the reasons mentioned above. In the system (4.8), since $E = E^T$, $A = A^T$ and $B = B^T$, the left transformation (W) and the right transformation (V) in IRKA are equal. Thus, one needs to construct only one transformation, e.g., V of the form

$$V = \left[(\alpha_1 E - A)^{-1}Bb_1, \cdots, (\alpha_r E - A)^{-1}Bb_r\right]. \tag{4.16}$$

Now using the transformation V we construct the ROM as in (4.13), where the reduced matrices are formed following (4.14). This completes the method. Now the question how to construct V in (4.16) efficiently is answered in the following.

In (4.16) each column of V can be computed by solving a shifted linear system like

$$(\alpha E - A)\chi = Bb, \tag{4.17}$$

which implies

$$(\alpha E_1 - A_1 + A_2 A_4^{-1} A_2^T)\chi = (B_1 - A_2 A_4^{-1} B_2)b.$$

Undoing the *Schur complement* [132], this linear system leads to

$$\begin{bmatrix} \alpha E_1 - A_1 & -A_2 \\ -A_2^T & -A_4 \end{bmatrix} \begin{bmatrix} \chi \\ \Gamma \end{bmatrix} = \begin{bmatrix} B_1 b \\ B_2 b \end{bmatrix}. \tag{4.18}$$

Inserting E_1, A_1, A_2, A_4, B_1 and B_2 from (4.7), the linear system (4.18) becomes

$$\begin{bmatrix} -M_{11} & \alpha M_{11} & 0 \\ \alpha M_{11} & \alpha D_{11} + K_{11} & K_{12} \\ 0 & K_{12}^T & K_{22} \end{bmatrix} \begin{bmatrix} \chi_1 \\ \chi_2 \\ \Gamma \end{bmatrix} = \begin{bmatrix} 0 \\ H_1 b \\ H_2 b \end{bmatrix}, \tag{4.19}$$

for $\begin{bmatrix} \chi_1^T & \chi_2^T \end{bmatrix}^T$. Although the matrix in (4.19) has larger dimension $(2n_\xi + n_\varphi)$, it is sparse and can efficiently be solved by suitable direct (e.g., [43, 47]) or iterative (e.g., [123, 101]) solvers. Further, splitting the linear system (4.19) as

$$-M_{11}\chi_1 + \alpha M_{11}\chi_2 = 0, \tag{4.20a}$$

$$\alpha M_{11}\chi_1 + (\alpha D_{11} + K_{11})\chi_2 + K_{12}\Gamma = H_1 b, \tag{4.20b}$$

$$K_{12}^T\chi_2 + K_{22}\Gamma = H_2 b, \tag{4.20c}$$

Algorithm 9: IRKA for second order index 1 systems.

```
Input  : $M_{11},\ D_{11},\ K_{11},\ K_{12},\ K_{22},\ H_1,\ H_2$.
Output: $\hat{E},\ \hat{A},\ \hat{B},\ \hat{D}_a := H_2^T K_4^{-1} H_2$.
Form $E_1,\ A_1,\ A_2,\ A_4,\ B_1,\ B_2$ as in (4.7).
Make an initial selection of the interpolation points $\{\alpha_i\}_{i=1}^r$ and tangential
directions $\{b_i\}_{i=1}^r$.
while (not converged) do
    for $i = 1, 2, \cdots, r$ do
        $\begin{bmatrix} \alpha_i^2 M_{11} + \alpha_i D_{11} + K_{11} & K_{12} \\ K_{12}^T & K_{22} \end{bmatrix} \begin{bmatrix} \chi_2 \\ \Gamma \end{bmatrix} = \begin{bmatrix} H_1 b_i \\ H_2 b_i \end{bmatrix}, \mathrm{v}_i = \begin{bmatrix} \alpha_i \chi_2 \\ \chi_2 \end{bmatrix}$,
        $V = \begin{bmatrix} \mathrm{v}_1, \mathrm{v}_2, \cdots, \mathrm{v}_r \end{bmatrix}$.
    end for
    $\hat{E} = V^T E_1 V$, $\hat{A} = V^T A_1 V - (V^T A_2) A_4^{-1} (A_2^T V)$,
    $\hat{B} = V^T B_1 V - (V^T A_2) A_4^{-1} B_2$.
    Compute $\hat{A} z_i = \hat{\lambda}_i \hat{E} z_i$, where $z_i$ is eigenvector associated with $\hat{\lambda}_i$.
    $\alpha_i \leftarrow -\hat{\lambda}_i$, $b_i \leftarrow -\hat{B}^T z_i$.
end while
Form the reduced matrices $\hat{E},\ \hat{A}$ and $\hat{B}$ as in (4.14).
```

from (4.20a) and (4.20c) we obtain respectively,

$$\chi_1 = \alpha \chi_2 \tag{4.21}$$

and $\Gamma = K_{22}^{-1} H_2 b_i - K_{22} K_{12}^T \chi_2$. Now inserting χ_1 and Γ into (4.20b) yields

$$\alpha^2 M_{11} \chi_2 + (\alpha D_{11} + K_{11}) \chi_2 - K_{12} K_{22}^{-1} K_{12}^T \chi_2 = H_1 b - K_{12} K_{22}^{-1} H_2 b,$$

which is again equivalent to the solution of the linear system

$$\begin{bmatrix} \alpha^2 M_{11} + \alpha D_{11} + K_{11} & K_{12} \\ K_{12}^T & K_{22} \end{bmatrix} \begin{bmatrix} \chi_2 \\ \Gamma \end{bmatrix} = \begin{bmatrix} H_1 b \\ H_2 b \end{bmatrix}, \tag{4.22}$$

for χ_2. Applying this splitting idea to the system (4.19), instead of solving an $2n_\xi + n_\varphi$ dimensional linear system we can solve only an $n_\xi + n_\varphi$ dimensional linear system, which ensures faster computation. We summarize the above idea in Algorithm 9 for computing the ROM (4.13) for the second order index 1 descriptor system (4.1).

4.3 Second-order-to-second-order MOR techniques

A balancing based second-order-to-second-order structure preserving MOR of the second order systems is discussed in Chapter 2 from the literature [88, 96, 20].

Unfortunately, all of those references contribute only for the second order ODE systems. We first introduced the second-order-to-second-order balancing criterion for the large-scale second order index 1 system (4.1) in [31]. This section discusses an efficient balancing based method for MOR of such structural second order index 1 DAEs. Here, we also propose that we can compute the ROM for such large-scale second order index 1 model via projecting the system onto the dominant eigenspace of the *second order system Gramian*, which is called the PDEG method. The results of this section are found in [33].

4.3.1 Balancing based method

Since the block matrix K_{22} is nonsingular, the second order index 1 system (4.1) can be transformed into the standard second order system (4.2). Again note that this transformation is not possible explicitly since there the matrix K is dense. Now converting (4.2) into the first order form in (4.8), we solve only the Lyapunov equation (4.9) for P. Let us recall the Gramians of the standard second order systems as defined in Chapter 2. Due to the structure of the system, the Gramian P can be partitioned as

$$P = \begin{bmatrix} P_v & P_0 \\ P_0^T & P_p \end{bmatrix}, \tag{4.23}$$

where P_v denotes either the controllability or the observability velocity Gramian and P_p denotes either the controllability or the observability position Gramian. The low-rank controllability or observability Gramian factor Z, defined in (4.10) then can be partitioned as $Z = \begin{bmatrix} Z_v^T & Z_p^T \end{bmatrix}^T$, such that

$$P \approx ZZ^T = \begin{bmatrix} Z_v \\ Z_p \end{bmatrix} \begin{bmatrix} Z_v^T & Z_p^T \end{bmatrix} = \begin{bmatrix} Z_v Z_v^T & Z_v Z_p^T \\ Z_p Z_v^T & Z_p Z_p^T \end{bmatrix}, \tag{4.24}$$

where Z_v is called the low-rank factor of the velocity Gramian and Z_p is called the low-rank factor of the position Gramian. Comparing (4.24) with (4.23), the relations

$$P_v \approx Z_v Z_v^T \quad \text{and} \quad P_p \approx Z_p Z_p^T, \tag{4.25}$$

can be obtained. Once the low-rank Gramian factor Z is computed by solving the Lyapunov equation (4.9), then Z_v and Z_p can be obtained by taking upper n_ξ and lower n_ξ rows of Z. Now using these low-rank Gramian factors and following (2.51-2.52), we can compute four types of balancing and truncating transformations which is summarized in Table 4.1. By applying each pair $(W_s,\ V_s)$ of the balancing and truncating transformations from this table we construct the ROM as

$$\begin{aligned} \hat{M}\ddot{\hat{\xi}}(t) + \hat{D}\dot{\hat{\xi}}(t) + \hat{K}\hat{\xi}(t) &= \hat{H}u(t), \\ \hat{y}(t) &= \hat{H}^T\hat{\xi}(t) + \hat{D}_a u(t), \end{aligned} \tag{4.26}$$

type	SVD	left proj. W_s	right proj. V_s
velocity-velocity (VV)	$Z_v^T M_{11} Z_v = U_{vv}\Sigma_{vv}U_{vv}^T$	$Z_v U_{vv,1}\Sigma_{vv,1}^{-\frac{1}{2}}$	$Z_v U_{vv,1}\Sigma_{vv,1}^{-\frac{1}{2}}$
position-position (PP)	$Z_p^T M_{11} Z_p = U_{pp}\Sigma_{pp}U_{pp}^T$	$Z_p U_{pp,1}\Sigma_{pp,1}^{-\frac{1}{2}}$	$Z_p U_{pp,1}\Sigma_{pp,1}^{-\frac{1}{2}}$
velocity-position (VP)	$Z_v^T M_{11} Z_p = U_{vp}\Sigma_{vp}V_{vp}^T$	$Z_p U_{vp,1}\Sigma_{vp,1}^{-\frac{1}{2}}$	$Z_v V_{vp,1}\Sigma_{vp,1}^{-\frac{1}{2}}$
position-velocity (PV)	$Z_p^T M_{11} Z_v = U_{pv}\Sigma_{pv}V_{pv}^T$	$Z_v U_{pv,1}\Sigma_{pv,1}^{-\frac{1}{2}}$	$Z_p V_{pv,1}\Sigma_{pv,1}^{-\frac{1}{2}}$

Table 4.1: Balancing transformations for the second order index 1 descriptor systems.

Algorithm 10: SOLR-SRM for second order index 1 system.

```
Input  : $M_{11}$, $D_{11}$, $K_{11}, K_{12}, K_{22}$ $H_1, H_2$.
Output: $\hat{M}$, $\hat{D}$, $\hat{K}$, $\hat{H}$, $\hat{D}_a := D_a$.
1 Solve the Lyapunov equation (4.9) to compute $Z = \begin{bmatrix} Z_v^T & Z_p^T \end{bmatrix}^T$.
2 Compute one of the four types of transformations following Table 4.1.
3 Construct $\hat{M}$, $\hat{D}$, $\hat{K}$ and $\hat{H}$ following (4.27).
```

where the reduced coefficient matrices are formed as

$$
\begin{aligned}
&\hat{M} = \ W_s^T M_{11} V_s, \quad \hat{D} = \ W_s^T D_{11} V_s, \quad \hat{K}_{11} = W_s^T K_{11} V_s,\\
&\hat{K}_{12} = W_s^T K_{12}, \quad \hat{K}_{21} = K_{12}^T V_s, \quad \hat{B}_1 = W_s^T B_1,\\
&\hat{K} = \hat{K}_{11} - \hat{K}_{12} K_{22}^{-1} \hat{K}_{21}, \quad \hat{H} = \hat{B}_1 - \hat{K}_{12} K_{22}^{-1} B_2, \quad \hat{D}_a = D_a.
\end{aligned}
\tag{4.27}
$$

When we use the pair $(Z_v,\ Z_v)$ to construct the balancing and truncating transformation, the balancing criterion is called *velocity-velocity* (VV) balancing. Analogously, the balancing criteria are called *position-position* (PP), *velocity-position* (VP) and *position-velocity* (PV) balancing if we use the low-rank Gramian factor pairs $(Z_p,\ Z_p)$, $(Z_v,\ Z_p)$ and $(Z_p,\ Z_v)$, respectively. Algorithm 10 summarizes the above procedure to construct the structure preserving ROMs from the second order index 1 system (4.1). From Table 4.1 we can note that in the case of the velocity-velocity and position-position balancing techniques the computed left and right balancing and truncating transformations are equal i.e., $W_s = V_s$. Therefore, in those cases the ROMs may preserve the stability and symmetry since the reduced matrices preserve the definiteness of the original matrices [128, Theorem 4]. It can also be shown that the velocity-position and position-velocity balancing based ROMs are adjoint to each other (see, e.g., [20] and the references therein). Therefore, they essentially show the same frequency responses (see the numerical results).

The ROM (4.26) is an ODE system. However, we can retain the reduced second order index 1 form as

$$\begin{bmatrix} \hat{M} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \ddot{\hat{\xi}}(t) \\ \ddot{\varphi}(t) \end{bmatrix} + \begin{bmatrix} \hat{D} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\hat{\xi}}(t) \\ \dot{\varphi}(t) \end{bmatrix} + \begin{bmatrix} \hat{K}_{11} & \hat{K}_{12} \\ \hat{K}_{21} & K_{22} \end{bmatrix} \begin{bmatrix} \hat{\xi}(t) \\ \varphi(t) \end{bmatrix} = \begin{bmatrix} \hat{H}_1 \\ H_2 \end{bmatrix} u(t), \\ \begin{bmatrix} \hat{H}_1^T & H_2^T \end{bmatrix} \begin{bmatrix} \hat{\xi}(t) \\ \varphi(t) \end{bmatrix} = \hat{y}(t). \tag{4.28}$$

4.3.2 Dominant eigenspace projection of the Gramian

Model reduction via projecting the system onto the dominant eigenspace of the system Gramian is introduced in [80, 79, 94]. However, there the proposed algorithm is for standard state space systems. Here we extend the idea for the structured second order index 1 systems. In the above discussion we already have defined the velocity Gramian P_v and the position Gramian P_p for the underlying system. Since P_v is symmetric positive definite (spd), it has a symmetric decomposition i.e.,

$$P_v = R_v R_v^T. \tag{4.29}$$

The SVD of R_v is

$$R_v = U_v \Sigma_v V_v^T, \tag{4.30}$$

where the diagonal matrix Σ_v consists of the decreasingly ordered singular values σ_{v_i}, $i = 1, 2, \dots, n_\xi$, of R_v. Using this SVD we obviously have

$$P_v = (U_v \Sigma_v V_v^T)(V_v \Sigma_v U_v^T) = U_v \Sigma_v^2 U_v^T. \tag{4.31}$$

This is also an eigenvalue decomposition where Σ_v^2 is a diagonal matrix whose entries are the decreasingly ordered eigenvalues of P_v and U_v is the orthogonal matrix consisting of the eigenvectors corresponding to the eigenvalues. We observe that U_v is the left singular vector matrix of R_v. Hence U_v is obtained by the SVD of R_v. Now identifying the k largest eigenvalues of P_v, construct

$$U_k = \begin{bmatrix} u_1, u_2, \dots, u_k \end{bmatrix}, \tag{4.32}$$

where u_i, $i = 1, 2, \dots, k$ are the eigenvectors corresponding to the eigenvalues σ_i^2. Then we construct the k dimensional reduced order model as in (4.26), by forming the reduced dimensional matrices as in (4.27), where $W_s = V_s = U_k$. Again, if we consider Z_v as a low-rank Gramian factor of the velocity Gramian such that $P_v \approx Z_v Z_v^T$, then we can compute U_k in (4.32) identifying the k largest left singular vectors of the SVD of Z_v.

The above procedure that constructs a k dimensional ROM (4.26) via projecting the system onto the dominant eigenspaces of the velocity Gramian P_v is summarized in Algorithm 11. This algorithm can also be used to obtain a k dimensional ROM via

Algorithm 11: PDEG for second order index 1 system.

```
Input  : $M_{11}, D_{11}, K_{11}, K_{12}, K_{22}\ H_1, H_2$.
Output: $\hat{M}$, $\hat{D}$, $\hat{K}$, $\hat{H}$, $\hat{D}_a := D_a$.
1 Compute $Z_v$ by solving Lyapunov equation.
2 Construct $U_k$ as in (4.32) using the thin SVD of $Z_v$.
3 Form reduced dimensional matrices $\hat{M}$, $\hat{D}$, $\hat{K}$, $\hat{H}$ following (4.27), where
  $W_s = V_s = U_k$.
```

projecting the system onto the eigenspace of the position Gramian P_p. In that case, in Step 2, instead of Z_v we use the low-rank position Gramian factor Z_p, where $P_p \approx Z_p Z_p^T$ to construct the transformation matrix U_k. Note that the pre-assigned order k of the reduced order model should satisfy the inequality

$$k \leq \dim(Z_v), \quad \text{or} \quad k \leq \dim(Z_p).$$

The transformation U_k is called *contra-gredient transformation* [77], since using this transformation we can show that

$$\begin{aligned} U_k^T P_v U_k &= U_k^T U_v \Sigma_v^2 U_v^T U_k \\ &= U_k^T \begin{bmatrix} U_k & U_{n_1-k} \end{bmatrix} \begin{bmatrix} \Sigma_k^2 & 0 \\ 0 & \Sigma_{n_1-k}^2 \end{bmatrix} \begin{bmatrix} U_k^T \\ U_{n_1-k}^T \end{bmatrix} U_k \\ &= \begin{bmatrix} I_k & 0 \end{bmatrix} \begin{bmatrix} \Sigma_k^2 & 0 \\ 0 & \Sigma_{n_1-k}^2 \end{bmatrix} \begin{bmatrix} I_k \\ 0 \end{bmatrix} \\ &= \Sigma_k^2, \end{aligned}$$

i.e., the Gramian of the reduced model is diagonal. This means that U_k is a kind of balancing transformation [65]. It can easily be shown that $\hat{M}$, $\hat{D}$ and $\hat{K}$ are all symmetric and they preserve their original definiteness as well. According to [128, Theorem 4] it can be guaranteed that the reduced model preserves the stability of the original model.

4.4 Efficient solution of the Lyapunov equation

In the previous sections we have seen that to carry out the BT and PDEG methods the main tool is the low-rank Gramian factor Z, which can be obtained by the solution of the Lyapunov equation (4.9). This section concentrates on how to compute this low-rank Gramian factor efficiently using the LRCF-ADI iteration introduced in Chapter 2. As we have mentioned that in contrast to our previous work e.g., [121, 122, 31], here the LRCF-ADI method is updated by computing the real low-rank Gramian factor. Moreover, we use low-rank residual factor based stopping techniques which makes their evaluation much cheaper (see, e.g., Chapter 2 for details). In addition, we show how to partition a large linear system into a small system to accelerate the solution. To ensure the fast convergence of the LRCF-ADI method we propose a novel technique for selecting the shift parameters adaptively. The details of this section are also available in [33].

4.4.1 Generalized sparse (GS)-LRCF-ADI iteration

To compute the low-rank Gramian factor by solving the Lyapunov equation (4.9) efficiently, we can apply Algorithm 5. There we have to replace the input matrices $\mathcal{E}$, $\mathcal{A}$ and $\mathcal{B}$, respectively, by $E,\ A$, and B. The initial guess of the residual is

$$W_0 = B = \begin{bmatrix} 0 \\ H_1 - K_{12}K_{22}^{-1}H_2 \end{bmatrix},$$

which can be partitioned as

$$W_0^{(1)} = 0 \quad \text{and} \quad W_0^{(2)} = H_1 - K_{12}K_{22}^{-1}H_2. \tag{4.33}$$

At the $i-$th step of the LRCF-ADI iteration (see, e.g., Algorithm 5), we need to compute $V_i = (A + \mu_i E)^{-1}W_{i-1}$ by solving the linear system

$$(A + \mu_i E)V_i = W_{i-1}. \tag{4.34}$$

Inserting E and A from (4.2) we obtain

$$\left(\begin{bmatrix} M_{11} & 0 \\ 0 & -(K_{11} - K_{12}K_{22}^{-1}K_{12}^T) \end{bmatrix} + \mu_i \begin{bmatrix} 0 & M_{11} \\ M_{11} & D_{11} \end{bmatrix}\right) \begin{bmatrix} V_i^{(1)} \\ V_i^{(2)} \end{bmatrix} = \begin{bmatrix} W_{i-1}^{(1)} \\ W_{i-1}^{(2)} \end{bmatrix}, \tag{4.35}$$

i.e.,

$$\begin{bmatrix} M_{11} & \mu_i M_{11} \\ \mu_i M_{11} & (\mu_i D_{11} - K_{11}) + K_{12}K_{22}^{-1}K_{12}^T \end{bmatrix} \begin{bmatrix} V_i^{(1)} \\ V_i^{(2)} \end{bmatrix} = \begin{bmatrix} W_{i-1}^{(1)} \\ W_{i-1}^{(2)} \end{bmatrix}. \tag{4.36}$$

It can easily be shown that by reversing the Schur complement instead of solving the linear system (4.36) we can solve the linear system

$$\begin{bmatrix} M_{11} & \mu_i M_{11} & 0 \\ \mu_i M_{11} & \mu_i(D_{11} - K_{11}) & -K_{12} \\ 0 & -K_{12}^T & -K_{22} \end{bmatrix} \begin{bmatrix} V_i^{(1)} \\ V_i^{(2)} \\ \Gamma \end{bmatrix} = \begin{bmatrix} W_{i-1}^{(1)} \\ W_{i-1}^{(2)} \\ 0 \end{bmatrix}, \tag{4.37}$$

for $\begin{bmatrix} V_i^{(1)^T} & V_i^{(2)^T} \end{bmatrix}^T$. Although the dimension of the matrices in (4.37) is higher than that of (4.36), it is sparse and therefore, it can be solved by using a sparse direct solver e.g., [43, Ch. 5], or any suitable iterative solver [101]. To ensure fast solution, we can partition the linear system (4.37) as follows. A simple algebraic manipulation on (4.37) shows that instead of solving the large liner system (4.37), we can compute $V_i^{(2)}$ from

$$\begin{bmatrix} \mu_i^2 M_{11} - \mu_i D_{11} + K_{11} & K_{12} \\ K_{12}^T & K_{22} \end{bmatrix} \begin{bmatrix} V_i^{(2)} \\ \Gamma \end{bmatrix} = \begin{bmatrix} \mu_i W_{i-1}^{(1)} - W_{i-1}^{(2)} \\ 0 \end{bmatrix}, \tag{4.38}$$

and then $V_i^{(1)}$ from $V_i^{(1)} = M_{11}^{-1} W_{i-1}^{(1)} - \mu_i V_i^{(2)}$. That means, as above, the splitting idea reduces the dimension of the linear system from $2n_\xi + n_\varphi$ to $n_\xi + n_\varphi$. Here $W_{i-1}^{(1)}$ and $W_{i-1}^{(2)}$ are already computed from the previous step (from the ADI residual) by following (2.61) as

$$W_i = W_{i-1} - 2\,\mathrm{Re}\,(\mu_i)\mathbb{E}V_i,$$

which implies

$$\begin{aligned} \begin{bmatrix} W_i^{(1)} \\ W_i^{(2)} \end{bmatrix} &= \begin{bmatrix} W_{i-1}^{(1)} \\ W_{i-1}^{(2)} \end{bmatrix} - 2\,\mathrm{Re}\,(\mu_i) \begin{bmatrix} 0 & M_{11} \\ M_{11} & D_{11} \end{bmatrix} \begin{bmatrix} V_i^{(1)} \\ V_i^{(2)} \end{bmatrix} \\ &= \begin{bmatrix} W_{i-1}^{(1)} - 2\,\mathrm{Re}\,(\mu_i) M_{11} V_i^{(2)} \\ W_{i-1}^{(2)} - 2\,\mathrm{Re}\,(\mu_i)(M_{11} V_i^{(1)} + D_{11} V_i^{(2)}) \end{bmatrix}. \end{aligned}$$

From this we get

$$\begin{aligned} W_i^{(1)} &= W_{i-1}^{(1)} - 2\,\mathrm{Re}\,(\mu_i) M_{11} V_i^{(2)}, \\ W_i^{(2)} &= W_{i-1}^{(2)} - 2\,\mathrm{Re}\,(\mu_i)(M_{11} V_i^{(1)} + D_{11} V_i^{(2)}). \end{aligned} \tag{4.39}$$

In case the two consecutive shift parameters are complex conjugates of each other, i.e., $\{\mu_i, \mu_{i+1} := \overline{\mu_i}\}$, recalling (2.62) here we have

$$W_{i+1} = W_{i-1} - 4\,\mathrm{Re}\,(\mu_i)\mathbb{E}\left(\mathrm{Re}\,(V_i) + \delta\,\mathrm{Im}\,(V_i)\right),$$

where $\delta = \frac{\mathrm{Re}\,(\mu_i)}{\mathrm{Im}\,(\mu_i)}$ and which gives

$$\begin{aligned} \begin{bmatrix} W_{i+1}^{(1)} \\ W_{i+1}^{(2)} \end{bmatrix} &= \begin{bmatrix} W_{i-1}^{(1)} \\ W_{i-1}^{(2)} \end{bmatrix} - 4\,\mathrm{Re}\,(\mu_i) \begin{bmatrix} 0 & M_{11} \\ M_{11} & D_{11} \end{bmatrix} \begin{bmatrix} \chi_1 \\ \chi_2 \end{bmatrix} \\ &= \begin{bmatrix} W_{i-1}^{(1)} - 4\,\mathrm{Re}\,(\mu_i) M_{11} \chi_2 \\ W_{i-1}^{(2)} - M_{11}\chi_1 + D_{11}\chi_2 \end{bmatrix}, \end{aligned}$$

where $\chi_1 = \left(\mathrm{Re}\,(V_i^{(1)}) + \delta\,\mathrm{Im}\,(V_i^{(1)})\right)$, $\chi_2 = \left(\mathrm{Re}\,(V_i^{(2)}) + \delta\,\mathrm{Im}\,(V_i^{(2)})\right)$. This results in

$$\begin{aligned} W_{i+1}^{(1)} &= W_{i-1}^{(1)} - 4\,\mathrm{Re}\,(\mu_i) M_{11} \chi_2, \\ W_{i+1}^{(2)} &= W_{i-1}^{(2)} - 4\,\mathrm{Re}\,(\mu_i)(M_{11}\chi_1 + D_{11}\chi_2). \end{aligned} \tag{4.40}$$

The procedure to compute the low-rank Gramian factor for the second order index 1 descriptor system (4.1) is outlined in Algorithm 12.

Algorithm 12: SOGS-LRCF-ADI for the second order index 1 systems.

```
Input  : $M_{11},\ D_{11},\ K_{11},\ K_{12},\ K_{22},\ H_1,\ H_2,\ \{\mu_i\}_{i=1}^{J}$.
Output: $Z = Z_i$, such that $P \approx ZZ^T$.
Set $Z_0 = [\,]$, $i = 1$.
$W_0^{(1)} = 0$ and $W_0^{(2)} = H_1 - K_{12}K_{22}^{-1}H_2$.
while $\|{W_{i-1}^{(1)}}^T W_{i-1}^{(1)} + {W_{i-1}^{(2)}}^T W_{i-1}^{(2)}\| \geq tol$ and $i \leq i_{max}$ do
    Solve $\begin{bmatrix} \mu_i^2 M_{11} - \mu_i D_{11} + K_{11} & K_{12} \\ K_{12}^T & K_{22} \end{bmatrix} \begin{bmatrix} V_i^{(2)} \\ \Gamma \end{bmatrix} = \begin{bmatrix} \mu_i W_{i-1}^{(1)} - W_{i-1}^{(2)} \\ 0 \end{bmatrix}$ for $V_i^{(2)}$.
    Compute $V_i^{(1)} = M_{11}^{-1} W_{i-1}^{(1)} - \mu_i V_i^{(2)}, \quad V_i = \begin{bmatrix} {V_i^{(1)}}^T & {V_i^{(2)}}^T \end{bmatrix}^T$.
    if $\mathrm{Im}\,(\mu_i) = 0$ then
        $Z_i = \begin{bmatrix} Z_{i-1} & \sqrt{2\mu_i}\,\mathrm{Re}\,(V_i) \end{bmatrix}$,
        $W_i^{(1)} = W_{i-1}^{(1)} - 2\mu_i M_{11} V_i^{(2)}$,
        $W_i^{(2)} = W_{i-1}^{(2)} - 2\mu_i (M_{11} V_i^{(1)} + D_{11} V_i^{(2)})$.
    else
        $\gamma = -2\,\mathrm{Re}\,(\mu_i), \quad \delta = \frac{\mathrm{Re}\,(\mu_i)}{\mathrm{Im}\,(\mu_i)}$,
        $Z_{i+1} = \begin{bmatrix} Z_{i-1} & \sqrt{2\gamma}\,(\mathrm{Re}\,(V_i) + \delta\,\mathrm{Im}\,(V_i)) & \sqrt{2\gamma}\sqrt{(\delta^2+1)}.\,\mathrm{Im}\,(V_i) \end{bmatrix}$,
        $W_{i+1}^{(1)} = W_{i-1}^{(1)} + 2\gamma M_{11}\chi_2, \quad W_{i+1}^{(2)} = W_{i-1}^{(2)} + 2\gamma(M_{11}\chi_1 + D_1\chi_2)$,
        where $\chi_1 = \mathrm{Re}\,(V_i^{(1)}) + \delta\,\mathrm{Im}\,(V_i^{(1)})$, $\chi_2 = \mathrm{Re}\,(V_i^{(2)}) + \delta\,\mathrm{Im}\,(V_i^{(2)})$.
        $i = i + 1$
    end if
    $i = i + 1$
end while
```

4.4.2 ADI shift parameter selection

For the fast convergence of Algorithm 12, proper ADI shift selection is necessary. In Chapter 2 we have mentioned that among different kinds of ADI shift parameters proposed in the literature, Penzl's heuristic shifts [93] are more commonly used for large-scale dynamical systems. For this model heuristic shift selection is discussed in [120, Algorithm 4.4], [31]. Besides Penzl's shifts we also investigate in our numerical experiments the adaptive shift [21] selection approach. In [21], the shifts are initialized by the eigenvalues of the pencil $\lambda E - A$ projected to the span of B, where E and A are defined in (4.2). Then whenever all the shifts in the set have been used, the pencil is projected to the span of the current V_i and the eigenvalues are used as the next set of shifts. Here we use the same initialization. For the update step however, we extend the subspace to all the V_i generated with the current set of shifts. Let us assume that U be the basis of the extended subspace. Now from the eigenvalues of $\lambda U^T EU - U^T AU$, select some desired number of optimal shifts by solving the ADI min-max (see Chapter 2) problem like in the heuristic

no. of iterations	normalized residual norm	
	heuristic shifts	adaptive shifts
100	9.88×10^{-1}	1.85×10^{-2}
200	9.99×10^{-1}	8.85×10^{-3}
300	9.78×10^{-1}	5.04×10^{-3}
400	9.69×10^{-1}	3.99×10^{-3}

Table 4.2: Comparison of the normalized residual norms using heuristic and adaptive shifts at different iteration steps in Algorithm 12.

procedure. This approach is repeated while the algorithm has not converged to the given tolerance. Note that our system is dissipative, i.e., all the eigenvalues of $\lambda(E+E^T)-(A+A^T)$ lie in the left complex plane. Therefore, Bendixon's theorem [85] ensures that all the eigenvalues of the projected pencil $\lambda U^TEU - U^TAU$ are stable.

4.5 Numerical results

In this section we illustrate numerical results to asses the accuracy and efficiency of our proposed techniques. The techniques are applied to a set of data for the finite element discretization of an adaptive spindle support (ASS) [74]. The dimension of the original model is $n = 290\,137$, which consists of $n_\xi = 282\,699$ differential equations and $n_\varphi = 7\,438$ algebraic equations. The number of the collocated inputs and outputs is 9.

All results have been obtained using MATLAB 7.11.0 (R2012a) on a board with 4 Intel® Xeon® E7-8837 CPUs with a 2.67-GHz clock speed, 8 Cores each and 1TB of total RAM.

To implement the BT and PDEG based reduced order models, first we compute the low-rank Gramian factor Z, by applying Algorithm 12. Both *heuristic* and *adaptive* shifts are compared to carry out this algorithm. In the case of the heuristic approach, we select 40 optimal shift parameters out of 60 large and 50 small magnitude approximate eigenvalues (see, e.g., [31] for details on the computation of heuristic ADI shift parameters for the ASS model). The algorithm is stopped by the maximum number of iteration steps $i_{\text{max}} = 400$. Next, we apply the adaptive shift computation approach to implement this algorithm. In this case, the algorithm is also stopped by $i_{\text{max}} = 400$. The convergence is compared in Table 4.2 in different iteration steps for both types of shift parameters. As we can see in this table the performances of the adaptive shifts is better than that of the heuristic shifts. Before using the computed low-rank Gramian factor for the model reduction algorithm, we compress the columns of Z applying the technique proposed in [92].

MOR tolerance	ROM dimension
10^{-4}	146
10^{-3}	140
10^{-2}	132
10^{-1}	123
10^{0}	98

Table 4.3: ROMs obtained by using different truncation tolerances.

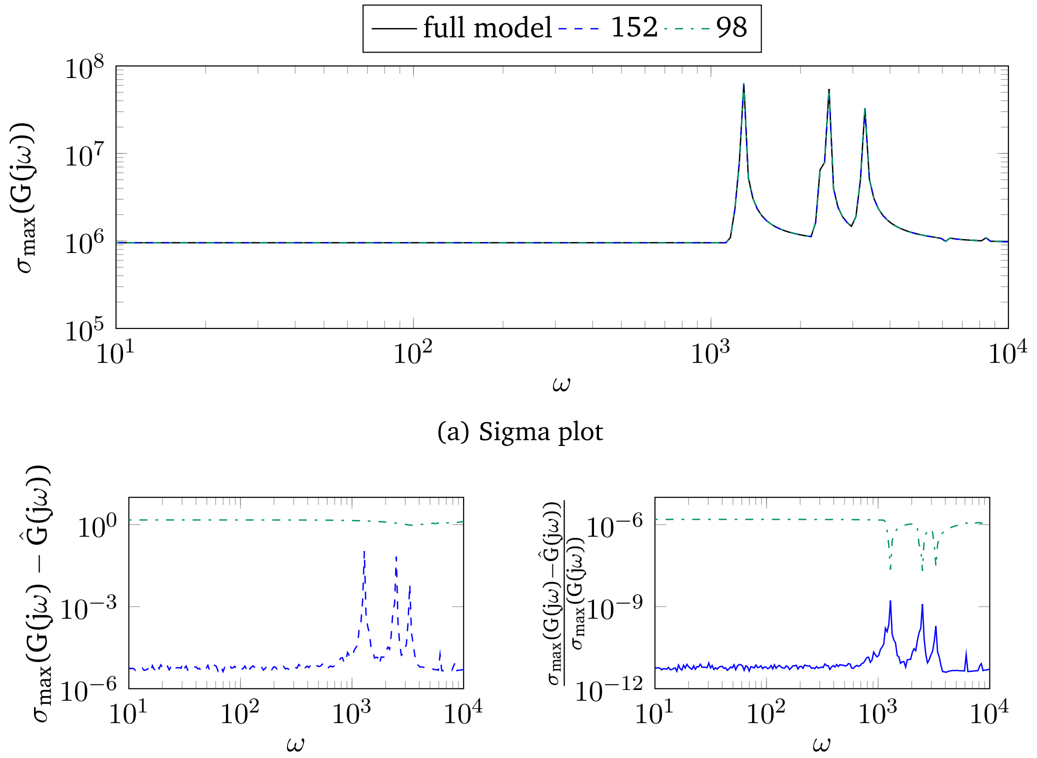
4.5.1 Second-order-to-first-order reduction

Balancing based methods: Algorithm 8 generated an order 152 reduced order model for the tolerance 10^{-5}. However, the dimension of the reduced order model can be reduced further by increasing the error tolerance, which is shown in Table 4.3. Figure 4.7 depicts, that all the ROMs obtained by different truncation tolerances match nicely with the original model keeping the relative error below the error bounds. We can also compute even lower dimensional ROMs if they are required for the controller design. In Figure 4.5 we see that although the approximation quality of the 5 dimensional ROM gets worse, order 50 - 10 dimensional models are satisfactory if an error of no more than 5% is desired. Even an order 10 model model still captures the important features of the original system.

Comparison with IRKA: To compare the balancing based method with IRKA we also compute different dimensional ROMs with Algorithm 9. Figure 4.6 shows the accuracy of the 60 and 10 dimensional BT and IRKA based reduced models. Here IRKA based reduced models show higher relative error. Note that Algorithm 9 is stopped after 50 cycles. That means we have updated the interpolation points and tangential directions 50 times. This number is still large. Perhaps, the quality of the ROMs can be improved further by considering even more cycles. In that case the computation would be more expensive.

4.5.2 Second-order-to-second-order reduction

Balancing based methods. For computing second-order-to-second-order ROMs using balanced truncation we first partition the computed Z as Z_v and Z_p by taking upper and lower n_ξ rows of Z and then applying Algorithm 10. This algorithm computes different dimensional reduced systems for the truncation tolerance 10^{-5} by using different types of balancing labels as shown in Table 4.1. The comparisons of the full and different dimensional reduced systems are shown in Figure 4.7. Figure 4.7a shows the frequency responses of full and reduced systems with good match. The absolute error and the relative error of the frequency responses of full and reduced systems are exhibited in Figure 4.7b and Figure 4.7c, respectively, with



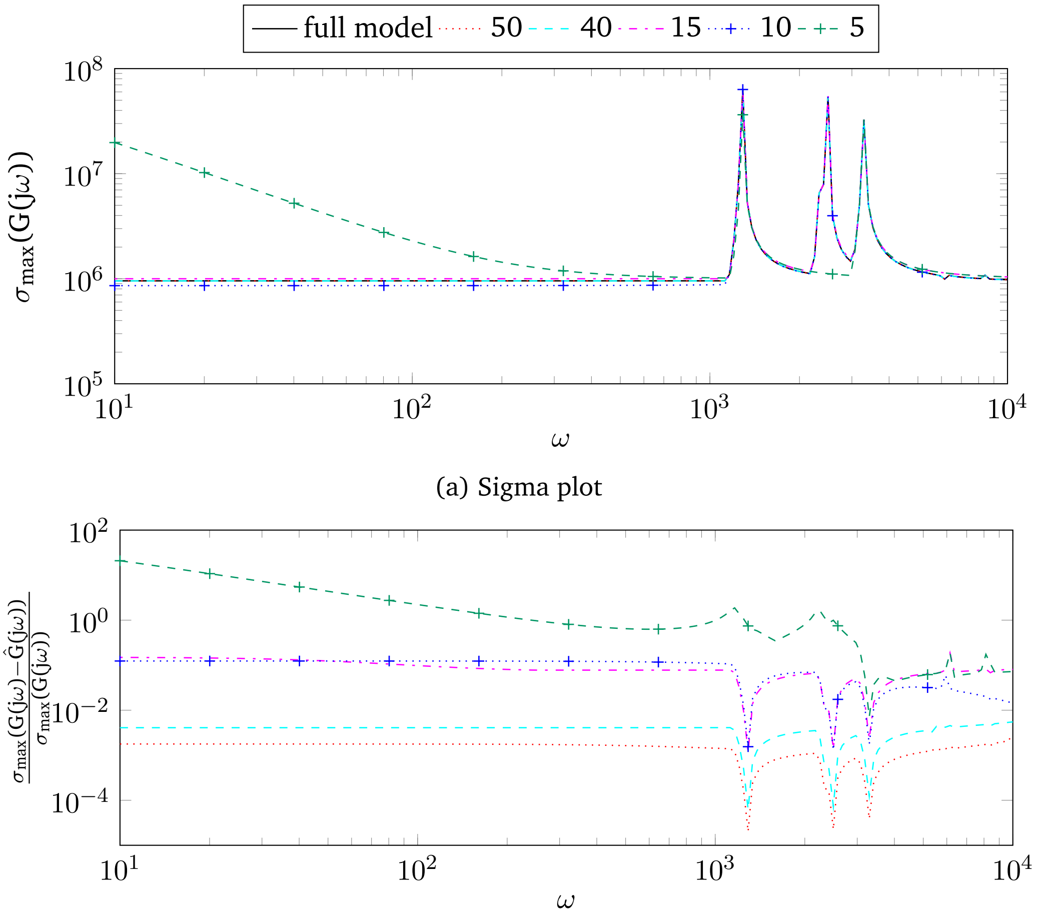
(a) Sigma plot

(b) Absolute error

(c) Relative error

Figure 4.4: Comparison of different dimensional reduced systems obtained by different tolerances.

very good accuracy. As we can see in Figure 4.7c, the relative errors for all reduced systems are far below to the truncation tolerance (10^{-5}). We further compute the $40, 30, 20$ and 10 dimensional reduced order models using the same algorithm via balancing the system on the velocity-velocity and position-position levels. In this case, the frequency responses of the reduced systems also resemble the graph in Figure 4.7a. Figure 4.8 depicts the relative errors between the full and different dimensional reduced order models. Here we observe that the lower the dimension of the reduced models the higher the relative error. But in both the balancing levels, even the very low dimensional models, e.g., a model of dimension 10, preserve the important feature of the original model. Figure 4.9 discusses the SISO relation of full and different dimensional reduced order models computed by position-position balancing. Since in the SISO case we know that the transfer function matrix is just a scalar rational function, here we have computed the absolute values of the transfer function in different frequencies. The relative error between the original and reduced order models of the respective SISO relation are also shown in the same figure. Table 4.4 shows the possible execution time gains that can be expected from the reduced order modeling. Here the computation at one sampling frequency (out of 200 used in the figures) is used as a representative for one eval-



(a) Sigma plot

(b) Relative error

Figure 4.5: Comparison of different dimensional reduced systems with the original system.

uation of the model. This roughly corresponds to the most expensive step in a time step for a simple integrator applied in the transient simulation of the system. Therefore similar speedups can be expected in those simulations.

PDEG based methods. Algorithm 11 is applied again on the ASS model to obtain the reduced systems via projecting the system onto the dominant eigenspace of the Gramian. To execute this algorithm, the computed velocity Gramian factor (Z_v) and and the position Gramian factor (Z_p) are the same as computed for implementing the balancing based method. By predefining the dimension of the ROM, we compute 40, 30, 20 and 10 dimensional models by projecting the system onto the dominant eigenspaces of both velocity and position Gramians. In both cases, the frequency responses of the original and reduced systems are the same as in Figure 4.9. Figure 4.10 shows the relative error between the original and the dif-

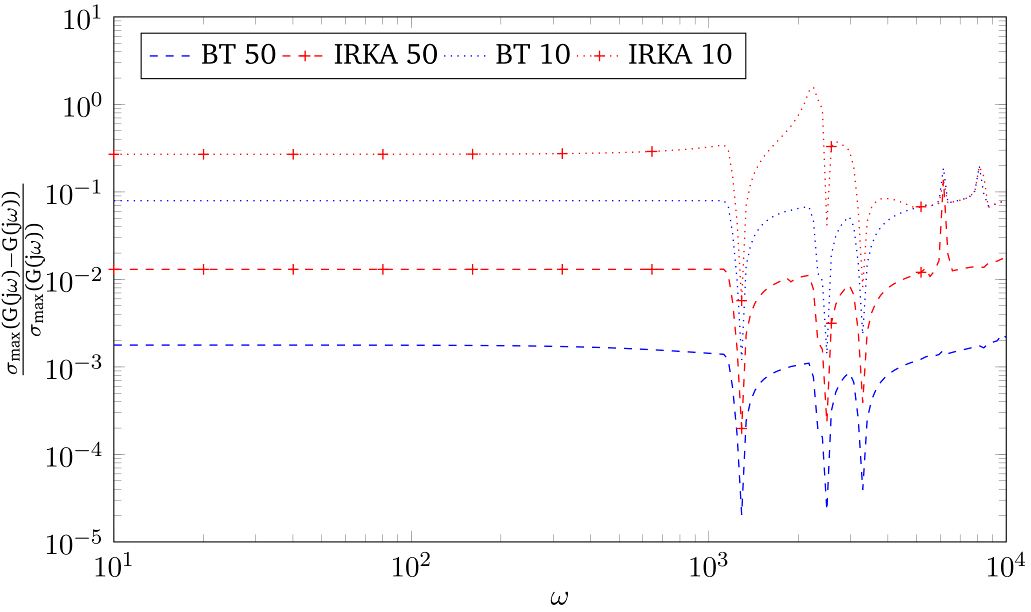
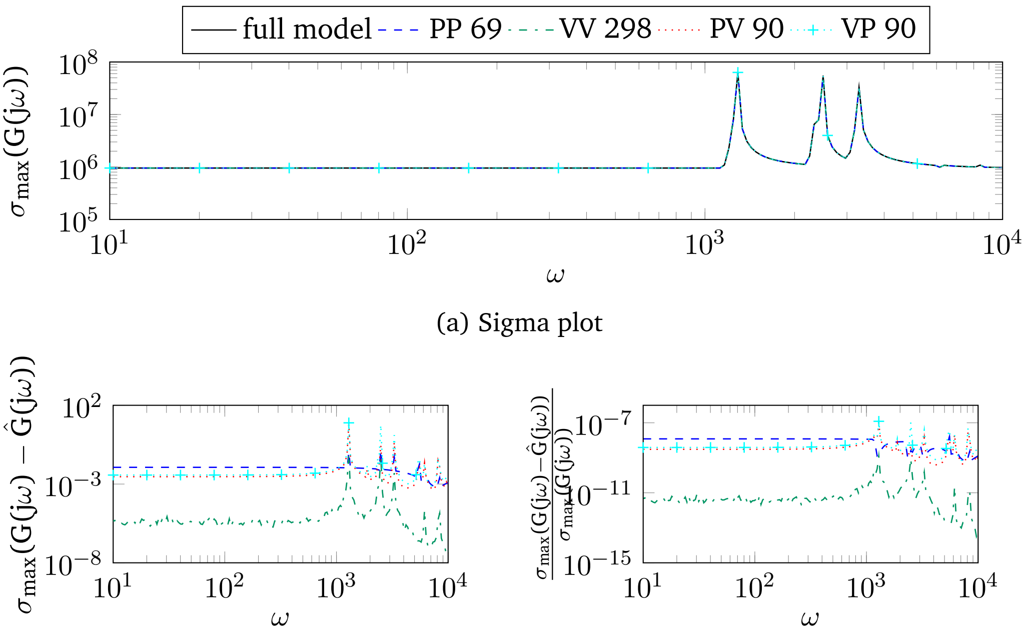


Figure 4.6: Relative error for different dimensional ROMs computed by Algorithms 8 and 9.

system dimension	execution time (sec)	speedup
290 137	90.00	1
50	0.0014	64 285
40	0.0012	75 000
30	0.0009	100 000
20	0.0007	128 571
10	0.0003	300 000

Table 4.4: Average execution time and speedup against full order model for computing the maximum Hankel singular value at a given sampling frequency.

ferent dimensional reduced models when we project the system onto the dominant eigenspace of the velocity Gramian (VG) (Figure 4.10a) and position Gramian (PG) (Figure 4.10b). We observe that the constructed reduced systems of the ASS model by PDEG methods are asymptotically stable which is shown in Figure 4.11. This figure shows that all the eigenvalues of the reduced systems which are obtained via projecting the system onto the dominant eigenspace of the position Gramian lie in the left complex half plane. From this figure one can also see that the successively decreasing dimensional reduced system contains the eigenvalues closer to the imaginary axis.

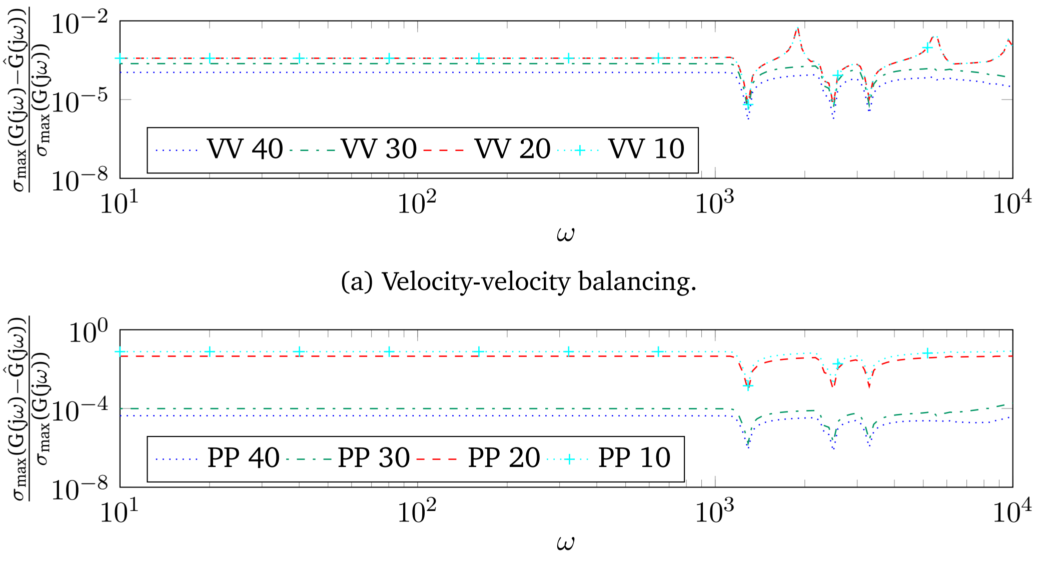


(a) Sigma plot

(b) Absolute error

(c) Relative error

Figure 4.7: Comparison of different dimensional reduced systems obtained by different balancing levels using truncation tolerance 10^{-5}.



(a) Velocity-velocity balancing.

(b) Position-position balancing.

Figure 4.8: Relative error between full and different dimensional reduced models via balanced truncation.

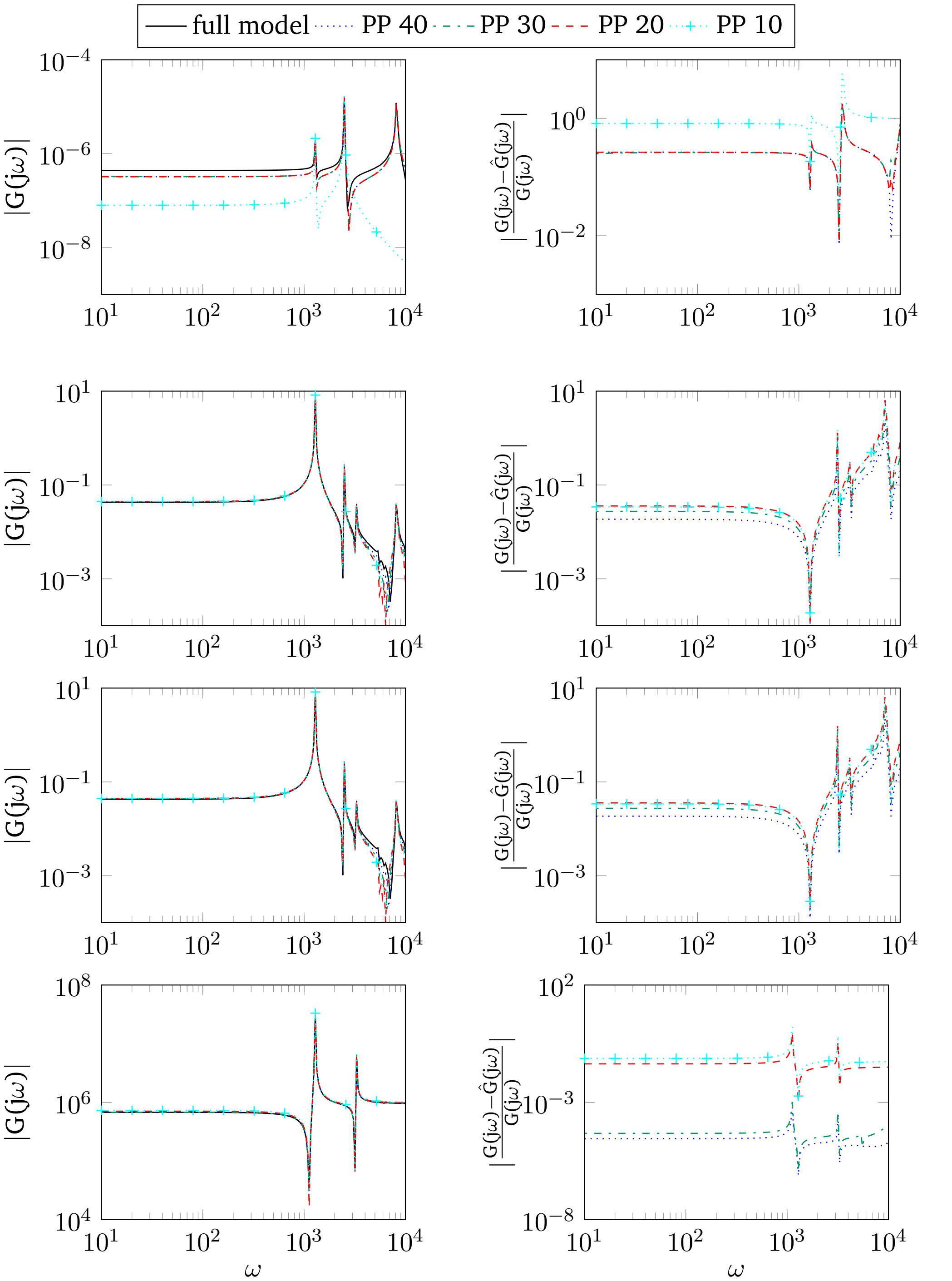
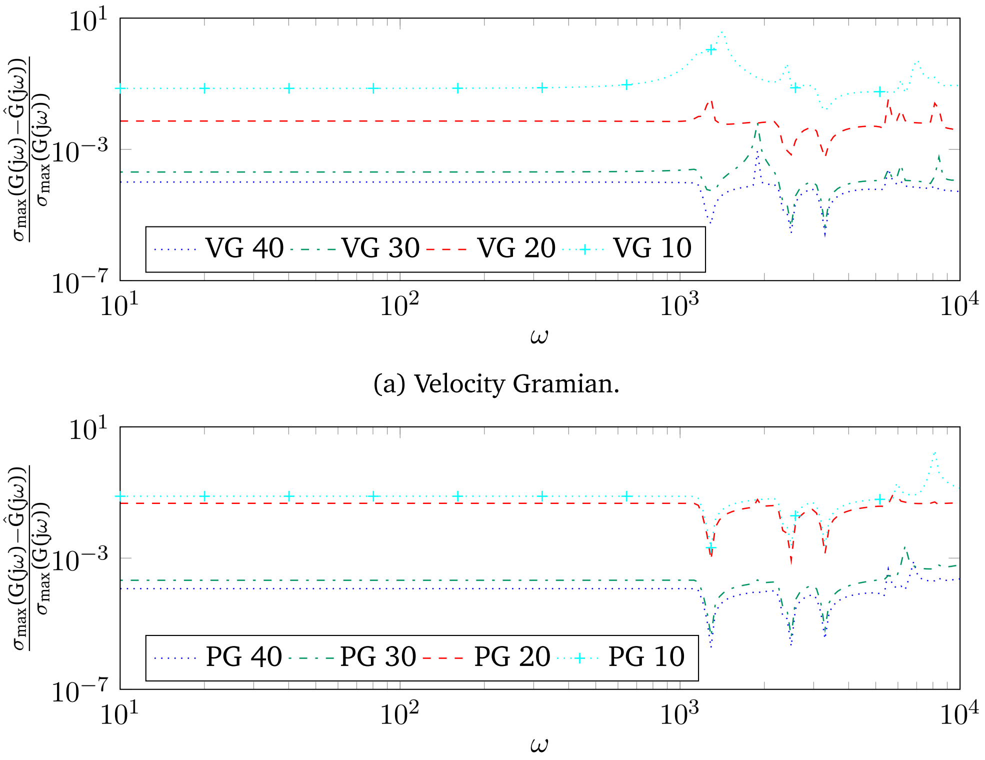


Figure 4.9: 1st, 2nd, 3rd and 4th rows respectively, show the 1st input to 1st output, 9th input to 1st output, 1st input to 9th and 9th input to 9th output relations (left) and the respective relative errors of full and different dimensional reduced systems.



(a) Velocity Gramian.

(b) Position Gramian.

Figure 4.10: Relative error between full and different dimensional reduced models via projecting onto the dominant eigen space of the Gramians.

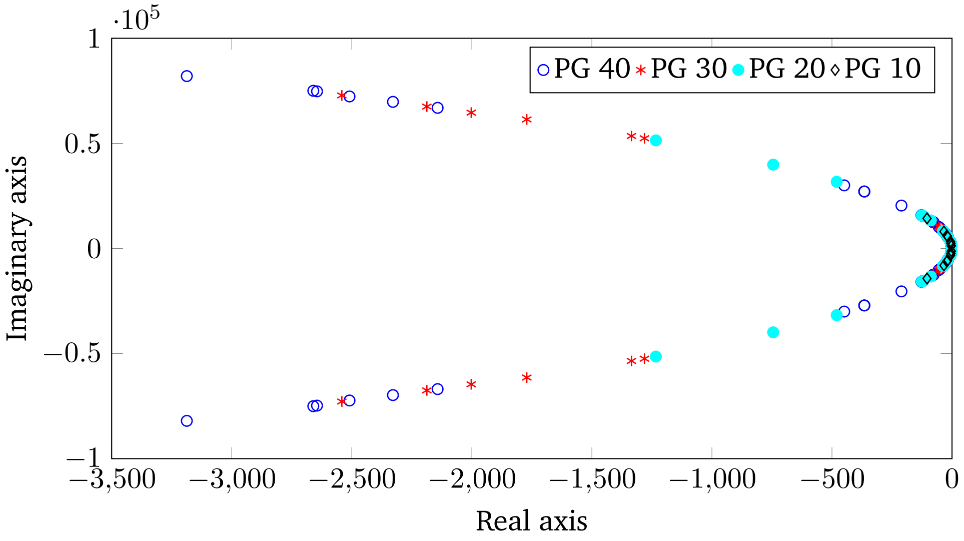


Figure 4.11: Eigenvalues of the ROM via projecting onto the dominant eigenspace of the position Gramian.

Chapter 5

Second Order Index 3 Descriptor Systems

This chapter presents model reduction methods for a class of structured second order index 3 descriptor systems of the form

$$
\begin{aligned}
\underbrace{\begin{bmatrix} M_1 & 0 \\ 0 & 0 \end{bmatrix}}_{\check{M}} \begin{bmatrix} \ddot{\xi}(t) \\ \ddot{\varphi}(t) \end{bmatrix} + \underbrace{\begin{bmatrix} D_1 & 0 \\ 0 & 0 \end{bmatrix}}_{\check{D}} \begin{bmatrix} \dot{\xi}(t) \\ \dot{\varphi}(t) \end{bmatrix} + \underbrace{\begin{bmatrix} K_1 & G_1^T \\ G_1 & 0 \end{bmatrix}}_{\check{K}} \begin{bmatrix} \xi(t) \\ \varphi(t) \end{bmatrix} &= \underbrace{\begin{bmatrix} H_1 \\ 0 \end{bmatrix}}_{\check{H}} u(t), \\
\underbrace{\begin{bmatrix} L_1 & 0 \end{bmatrix}}_{\check{L}} \begin{bmatrix} \xi(t) \\ \varphi(t) \end{bmatrix} &= y(t),
\end{aligned}
\tag{5.1}
$$

where $\xi(t) \in \mathbb{R}^{n_\xi}$, $\varphi(t) \in \mathbb{R}^{n_\varphi}$ are the states, $n_\xi > n_\varphi$, $u(t) \in \mathbb{R}^m$ are the inputs, $y(t) \in \mathbb{R}^p$ are the outputs, and $\check{M}$, $\check{D}$, $\check{K}$, $\check{H}$, $\check{L}$ are all sparse matrices with appropriate dimensions. Such structured systems arise in many applications, e.g., in constraint multibody system dynamics [107, 48] (see next section for details) or mechanical systems with *holonomic constraints* [128, 87]. The system (5.1) is called an index 3 system due to the analogy to first order index 3 (see, e.g., section 5.1) linear time-invariant (LTI) systems. Following Chapter 3, we also eliminate the algebraic elements by projecting the system onto the subspace where the solutions of the descriptor system exist. We show the projected and original systems are equivalent in the sense that they have the same finite spectrum. Then both second-order-to-first-order and second-order-to-second-order reduction techniques are shown for the projected systems. In the case of second-order-to-first-order reduction, we discuss both balanced truncation and an interpolatory technique via IRKA. To implement the methods, the second order projected system is converted into a first order form. The first order projected systems are very similar to the projected system considered in [70, 68]. Following their strategies (see also, e.g., Chapter 3) we show a technique to avoid the computation of the projector for implementing the

BT and interpolatory methods. On the other hand, for the second-order-to-second-order reduction method, besides the balanced truncation we also discuss the PDEG method, introduced in the previous chapter. In this case we also discuss the issues in avoiding the projector. The BT and PDEG methods rely on controllability and observability Gramian factors. To compute the Gramian factors we need to solve two projected continuous-time algebraic Lyapunov equation. Following Chapter 3, here we show an efficient technique to solve the projected Lyapunov equations handling the projector implicitly. Moreover, we discuss the difficulties of the ADI shift parameter computations using both heuristic and adaptive approaches and suggest how to overcome these. The proposed techniques are applied to several test examples. Numerical results are discussed to show the efficiency of the techniques.

5.1 Motivating examples

In classical mechanics or multibody dynamics, see, e.g., [8, 48, 44], the governing mathematical model can often be described by a simple equation of motion

$$M_1(\xi)\ddot{\xi} = f_a(\xi, \dot{\xi}, u(t)), \tag{5.2}$$

where $M_1(\xi) \in \mathbb{R}^{n_\xi \times n_\xi}$ is the positive definite mass matrix, $f_a \in \mathbb{R}^{n_\xi}$ the vector of the force function, $u(t) \in \mathbb{R}^m$ is the input vector, and ξ, $\dot{\xi}$ and $\ddot{\xi}$ denote, respectively, the time dependent vector of the position, velocity and acceleration.

In the simple case the mechanical models can be described by the unconstrained equation of motion (5.5). However, the more general case is the equation of motion under constraints. Constraints are the conditions restricting possible geometrical positions of the mechanical system or limiting its motion. Sometimes constraints are required in a system to guide the motion along a prescribed curve or surface. The equation of motion with constraints has the following form [48, 128]

$$M_1(\xi)\ddot{\xi}(t) = f_a(\xi, \dot{\xi}, u(t)) - f_c(\xi, \varphi), \tag{5.3}$$

$$g(\xi) = 0, \tag{5.4}$$

where $g(\xi)$ is a vector valued function describing n_φ constraints, f_c represents the generalized constraint forces acting on the system. This additional force term f_c is imposed for the constraint to be satisfied. It can be shown that these constraint forces are orthogonal to the constraints which define the manifold of the system. This means $f_c(\xi, \varphi) = G_1(\xi)^T \varphi$, where $G_1(\xi) := \frac{d}{d\xi} g(\xi)$ and φ is the vector of the *Lagrange multipliers*. Thus the system in (5.3) can be written as

$$M_1(\xi)\ddot{\xi}(t) = f_a(\xi, \dot{\xi}, u(t)) - G_1(\xi)^T \varphi, \tag{5.5}$$

$$g(\xi) = 0. \tag{5.6}$$

Now linearizing (5.5) around the equilibrium point (see, e.g., [48, Chap 1] for a linearizion technique) we obtain the linearized constraint equation of motion of the form

$$\begin{aligned} M_1\ddot{\xi}(t) + D_1\dot{\xi}(t) + K_1\xi(t) + G_1^T\varphi(t) &= H_1 u(t), \\ G_1\xi(t) &= 0, \end{aligned} \tag{5.7}$$

where the $n_\xi \times n_\xi$ dimensional coefficient matrices M_1, D_1 and K_1 are called the mechanical mass, damper and stiffness matrices, respectively, and $H_1 \in \mathbb{R}^{n_\xi \times n_\varphi}$ is the input matrix. The corresponding outputs can be measured by

$$y(t) = L_1\xi(t), \tag{5.8}$$

where $L_1 \in \mathbb{R}^{n_\xi \times n_\xi}$ is the output matrix. In matrix vector form the linearized equation of motion (5.7) together with the output equation (5.8) gives (5.1). We call the system (5.1) second order index 3 DAEs since one of the suitable first order conversions

$$\begin{aligned} \begin{bmatrix} I_{n_\xi} & 0 & 0 \\ 0 & M_1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\xi}(t) \\ \ddot{\xi}(t) \\ \dot{\varphi}(t) \end{bmatrix} &= \begin{bmatrix} 0 & I_{n_\xi} & 0 \\ -K_1 & -D_1 & -G_1^T \\ G_1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \xi(t) \\ \dot{\xi}(t) \\ \varphi(t) \end{bmatrix} + \begin{bmatrix} 0 \\ H_1 \\ 0 \end{bmatrix} u(t), \\ y(t) &= \begin{bmatrix} L_1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \xi(t) \\ \dot{\xi}(t) \\ \varphi(t) \end{bmatrix}, \end{aligned} \tag{5.9}$$

is in the index 3 descriptor form [75, 48, 35].

5.2 Index reduction

This section will show how to convert an index 3 descriptor system of the form (5.1) into an equivalent form of a ODE system via projection of the system onto the *hidden manifold* on which the solution evolves. First we focus on the construction of the projector by exploiting the structure of the system. Second, we prove that the finite spectra of the original and projected systems coincide.

5.2.1 Reformulation of the dynamical systems

Let us rewrite the second order index 3 system (5.1) as

$$M_1\ddot{\xi}(t) = -D_1\dot{\xi}(t) - K_1\xi(t) - G_1^T\varphi(t) + H_1u(t), \tag{5.10a}$$

$$G_1\xi(t) = 0, \tag{5.10b}$$

$$y(t) = L_1\xi(t). \tag{5.10c}$$

From (5.10b) we obtain $G_1\ddot{\xi}(t) = 0$. Inserting this identity after multiplying both sides of (5.10a) by $G_1M_1^{-1}$, we find

$$0 = -G_1M_1^{-1}D_1\dot{\xi}(t) - G_1M_1^{-1}K_1\xi(t) - G_1M_1^{-1}G_1^T\varphi(t) + G_1M_1^{-1}H_1u(t), \tag{5.11}$$

which implies

$$\begin{aligned}\varphi(t) = -(G_1M_1^{-1}G_1^T)^{-1}G_1M_1^{-1}D_1\dot{\xi}(t) - (G_1M_1^{-1}G_1^T)^{-1}G_1M_1^{-1}K_1\xi(t) + \\ (G_1M_1^{-1}G_1^T)^{-1}G_1M_1^{-1}H_1u(t).\end{aligned} \tag{5.12}$$

Inserting $\varphi(t)$ into (5.10a) we obtain

$$M_1\ddot{\xi}(t) = -\Pi D_1\dot{\xi}(t) - \Pi K_1\xi(t) + \Pi H_1u(t), \tag{5.13}$$

where

$$\Pi := I_{n_\xi} - G_1^T(G_1M_1^{-1}G_1^T)^{-1}G_1M_1^{-1}, \tag{5.14}$$

in which I_{n_ξ} is an identity matrix of size n_ξ. In fact, Π is a projector since

$$\begin{aligned}\Pi^2 &= (I_{n_\xi} - G_1^T(G_1M_1^{-1}G_1^T)^{-1}G_1M_1^{-1})(I_{n_\xi} - G_1^T(G_1M_1^{-1}G_1^T)^{-1}G_1M_1^{-1}) \\ &= I_{n_\xi} - 2G_1^T(G_1M_1^{-1}G_1^T)^{-1}G_1M_1^{-1} + G_1^T(G_1M_1^{-1}G_1^T)^{-1}G_1M_1^{-1} \\ &= I_{n_\xi} - G_1^T(G_1M_1^{-1}G_1^T)^{-1}G_1M_1^{-1} = \Pi.\end{aligned}$$

The projector Π satisfies the following properties.

Proposition 5.1. *Let Π be the projector defined above. The following conditions hold.*

1. $\Pi M_1 = M_1\Pi^T$.
2. $\text{Null}\,(\Pi) = \text{Range}\left(G_1^T\right)$.
3. $\text{Range}\,(\Pi) = \text{Null}\left(G_1M_1^{-1}\right)$.

Proof. 1. We have

$$\begin{aligned}\Pi M_1 &= M_1 - G_1^T(G_1M_1^{-1}G_1^T)^{-1}G_1 \\ &= M_1(I_{n_\xi} - M_1^{-1}G_1^T(G_1M_1^{-1}G_1^T)^{-1}G_1) = M_1\Pi^T.\end{aligned}$$

2. Suppose that the vector a belongs to the nullspace of Π, i.e., $\Pi a = 0$. By the definition of Π, $(I_{n_\xi} - G_1^T(G_1M_1^{-1}G_1^T)^{-1}G_1M_1^{-1})a = 0$, which implies $a = G_1^T(G_1M_1^{-1}G_1^T)^{-1}G_1M_1^{-1}a$. So $a = G_1^Tb$, where $(G_1M_1^{-1}G_1^T)^{-1}G_1M_1^{-1}a = b$, which implies that a is in the range of G_1^T. Therefore,

$$\text{Null}\,(\Pi) \subseteq \text{Range}\left(G_1^T\right). \tag{5.15}$$

Conversely, suppose that a is in the range of G_1^T. Therefore, there exists a non-zero vector b, such that $G_1^Tb = a$. Multiplying both sides by $G_1M_1^{-1}$, $G_1M_1^{-1}G_1^Tb = G_1M_1^{-1}a$, which implies $b = (G_1M_1^{-1}G_1^T)^{-1}G_1M_1^{-1}a$. Again multiplying both sides by G_1^T, $G_1^Tb = G_1^T(G_1M_1^{-1}G_1^T)^{-1}G_1M_1^{-1}a$. So $a = G_1^T(G_1M_1^{-1}G_1^T)^{-1}G_1M_1^{-1}a$, which implies $\Pi a = 0$. Therefore a is also in the nullspace of Π, and hence

$$\text{Range}\left(G_1^T\right) \subseteq \text{Null}\,(\Pi). \tag{5.16}$$

Equation (5.15) and (5.16) prove $\mathrm{Null}\,(\Pi) = \mathrm{Range}\left(G_1^T\right)$.

3. Again, we assume a is in the range of Π, i.e., $\Pi a = a$, which implies $(I_{n_\xi} - G_1^T(G_1M_1^{-1}G_1^T)^{-1}G_1M_1^{-1})a = a$, or $G_1^T(G_1M_1^{-1}G_1^T)^{-1}G_1^TM_1^{-1}a = 0$. Let $\Phi b = 0$, where $\Phi = G_1^T(G_1M_1^{-1}G_1^T)^{-1}$ and $b = G_1^TM_1^{-1}a$. Multiplying both sides by Φ^T, we obtain $\Phi^T\Phi b = 0$. Since $\Phi^T\Phi$ is invertible we see $b = 0$, or $G_1M_1^{-1}a = 0$. This proves that a is in the nullspace of $G_1M_1^{-1}$. Therefore,

$$\mathrm{Range}\,(\Pi) \subseteq \mathrm{Null}\left(G_1M_1^{-1}\right). \tag{5.17}$$

Conversely, again suppose that $a \in \mathrm{Null}\left(G_1M_1^{-1}\right)$, i.e., $G_1M_1^{-1}a = 0$. Multiplying both sides by $\Phi^T\Phi$, we get $\Phi^T\Phi G_1M_1^{-1}a = 0$. Again multiplying both sides by b^T, $b^T\Phi^T\Phi b = 0$, which implies $(\Phi b)^T(\Phi b) = 0$, hence $\Phi b = 0$. Therefore, $G_1^T(G_1M_1^{-1}G_1^T)^{-1}G_1M_1^{-1}a = 0$ i.e., $a = 0$, and therefore, $\Pi a = a$, such that

$$\mathrm{Null}\left(G_1M_1^{-1}\right) \subseteq \mathrm{Range}\,(\Pi). \tag{5.18}$$

Therefore, equation (5.17) and (5.18) yield $\mathrm{Range}\,(\Pi) = \mathrm{Null}\left(G_1M_1^{-1}\right)$. □

Theorem 5.1. *The vector a is in the nullspace of G_1, i.e., $G_1a = 0$ iff $\Pi^Ta = a$, where Π is defined in (5.14).*

Proof. Suppose the vector a is in the nullspace of G_1, i.e., $G_1a = 0$. Multiplying both sides by $-M_1^{-1}G_1^T(G_1M_1^{-1}G_1^T)^{-1}$, we obtain $-M_1^{-1}G_1^T(G_1M_1^{-1}G_1^T)^{-1}G_1a = 0$, which is equivalent to $(I_{n_\xi} - M_1^{-1}G_1^T(G_1M_1^{-1}G_1^T)^{-1}G_1)a = a$, i.e., $\Pi^Ta = a$. Conversely, suppose that $\Pi^Ta = a$, which implies $(I_{n_\xi} - M_1^{-1}G_1^T(G_1M_1^{-1}G_1^T)^{-1}G_1)a = a$. We see $M_1^{-1}G_1^T(G_1M_1^{-1}G_1^T)^{-1}G_1a = 0$. Multiplying both sides by G_1 we obtain $G_1a = 0$. □

Following Theorem 5.1, equation (5.10b) implies

$$\Pi^T\xi(t) = \xi(t). \tag{5.19}$$

Inserting this identity into (5.13) and multiplying the resulting equation by Π, we obtain

$$\Pi M_1\Pi^T\ddot{\xi}(t) = -\Pi K_1\Pi^T\xi(t) - \Pi D_1\Pi^T\dot{\xi}(t) + \Pi H_1u(t). \tag{5.20}$$

Moreover, applying (5.19) into the output equation (5.10c), we find the system in (5.10) is equivalent to

$$\Pi M_1\Pi^T\ddot{\xi}(t) = -\Pi D_1\Pi^T\dot{\xi}(t) - \Pi K_1\Pi^T\xi(t) + \Pi H_1u(t), \tag{5.21a}$$
$$y(t) = L_1\Pi^T\xi(t). \tag{5.21b}$$

The system dynamics of (5.21) are projected onto the $n_m := n_\xi - n_\varphi$ dimensional subspace $\operatorname{Range}(\Pi^T)$. This subspace is, however, still represented in the coordinates of the n_ξ dimensional space. The n_m dimensional representation can be made explicit by employing the *thin* singular value decomposition (SVD)

$$\Pi = U\Sigma V^T = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} = U_1 \Sigma_1 V_1^T = \Theta_l \Theta_r^T, \tag{5.22}$$

where $\Theta_l = U_1\Sigma_1^{\frac{1}{2}}$, $\Theta_r = V_1\Sigma_1^{\frac{1}{2}}$, and $U_1,\ V_1 \in \mathbb{R}^{n_\xi \times n_m}$ consist of the corresponding leading n_m columns of U, $V \in \mathbb{R}^{n_\xi \times n_\xi}$. Moreover, Θ_l, Θ_r satisfy

$$\Theta_l^T \Theta_r = I_{n_m}. \tag{5.23}$$

Inserting the decomposition of Π from (5.22) into (5.21) and considering $\tilde{\xi}(t) = {\Theta_l}^T \xi(t)$ the resulting dynamical system leads to

$$\Theta_r^T M_1 \Theta_r \ddot{\tilde{\xi}}(t) = -\Theta_r^T D_1 \Theta_r \dot{\tilde{\xi}}(t) - \Theta_r^T K_1 \Theta_r \tilde{\xi}(t) + \Theta_r^T H_1 u(t), \tag{5.24a}$$
$$y(t) = L_1 \Theta_r \tilde{\xi}(t). \tag{5.24b}$$

System (5.24) is now a standard second order system like (2.21). This system practically is system (5.21) with the redundant equation removed by the Θ_r projection. The dynamical systems (5.10), (5.21) and (5.24) are equivalent in a sense that they are different realizations of the same transfer function. Moreover, their finite spectrum is the same, which we prove in the following section.

5.2.2 Equivalent finite spectra

The quadratic matrix polynomial [100, 115, 116, 10] associated with the index 3 DAEs system (5.1) is

$$Q(\lambda) = \lambda^2 \underbrace{\begin{bmatrix} M_1 & 0 \\ 0 & 0 \end{bmatrix}}_{M} + \lambda \underbrace{\begin{bmatrix} D_1 & 0 \\ 0 & 0 \end{bmatrix}}_{D} + \underbrace{\begin{bmatrix} K_1 & G_1^T \\ G_1 & 0 \end{bmatrix}}_{K}, \tag{5.25}$$

where $\lambda \in \mathbb{C}$. Although $Q(\lambda)$ is regular, due to the singularity of M, it contains some infinite eigenvalues as well. If the degree of $\det(Q(\lambda))$ is $r < 2\tilde{n}$, where $\tilde{n} = n_\xi + n_\varphi$, then $Q(\lambda)$ has r finite and $2\tilde{n} - r$ infinite eigenvalues [116]. Again the quadratic matrix polynomials corresponding to the systems (5.21) and (5.24) are respectively,

$$\tilde{Q}(\lambda) = \lambda^2 \Pi M_1 \Pi^T + \lambda \Pi D_1 \Pi^T + \Pi K_1 \Pi^T \tag{5.26}$$

and

$$\bar{Q}(\lambda) = \lambda^2 \Theta_r^T M_1 \Theta_r + \lambda \Theta_r^T D_1 \Theta_r + \Theta_r^T K_1 \Theta_r. \tag{5.27}$$

We know $\Theta_r^T M_1 \Theta_r \in \mathbb{R}^{n_\xi \times n_\xi}$ is non singular and $\bar{Q}$ is regular. Hence, all of the eigenvalues of the polynomial $\bar{Q}(\lambda)$ are finite [116]. The degree of $\det\left(\bar{Q}(\lambda)\right)$ is $2(n_\xi - n_\varphi)$. Hence the number of finite eigenvalues of $\bar{Q}(\lambda)$ is exactly $2(n_\xi - n_\varphi)$, to which we can add $2\tilde{n} - 2(n_\xi - n_\varphi) = 2n_\xi + 2n_\varphi - 2n_\xi + 2n_\varphi = 4n_\varphi$ infinite eigenvalues. Applying the appropriate projectors onto the index 3 DAE system, we can preserve all the finite eigenvalues of the system (5.24). The following theorem demonstrates that all the finite eigenvalues of the original and projected systems are the same.

Theorem 5.2. *Let us consider the matrix polynomials $Q(\lambda)$ and $\bar{Q}(\lambda)$, defined respectively, in (5.25) and (5.26). An eigenvalue λ_1 is a finite eigenvalue of $Q(\lambda)$ with corresponding eigenvector $\begin{bmatrix} v_1^T & v_2^T \end{bmatrix}^T$ if and only if λ_1 is an eigenvalue of $\bar{Q}(\lambda)$ with corresponding eigenvector $\tilde{v}_1$ where $\tilde{v}_1 = \Theta_l^T v_1$ and Θ_l is defined in (5.23).*

Proof. Suppose, λ_1 is a finite eigenvalue of $Q(\lambda)$ corresponding to the eigenvector $\begin{bmatrix} v_1^T & v_2^T \end{bmatrix}^T$. Then the quadratic eigenvalue problem of the matrix polynomial (5.25) is

$$\left(\lambda_1^2 \begin{bmatrix} M_1 & 0 \\ 0 & 0 \end{bmatrix} + \lambda \begin{bmatrix} D_1 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} K_1 & G_1^T \\ G_1 & 0 \end{bmatrix}\right) \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{5.28}$$

The last line of (5.28) gives $G_1 v_1 = 0$, i.e., v_1 is in the nullspace of G_1. Now applying Theorem 5.1, we obtain $\Pi^T v_1 = v_1$. Plug $\Pi^T v_1 = v_1$ into the first equation of (5.28) and then project the resulting equation from the left by Π. Since $\Pi G_1^T = 0$, by Proposition 5.1, this leads to

$$(\lambda_1^2 \Pi M_1 \Pi^T + \lambda_1 \Pi D_1 \Pi^T + \Pi K_1 \Pi^T) v_1 = 0, \tag{5.29}$$

which is the eigenvalue problem for the matrix polynomial $\tilde{Q}(\lambda)$. Applying the decompositions of Π as defined above to (5.22) and using $\tilde{v}_1 = \Theta_l^T v_1$ we obtain

$$\Theta_l(\lambda_1^2 \Theta_r^T M_1 \Theta_r + \lambda_1 \Theta_r^T D_1 \Theta_r + \Theta_r^T K_1 \Theta_r)\tilde{v}_1 = 0.$$

Multiplying by Θ_r from the left and using (5.23) yields

$$(\lambda_1^2 \Theta_r^T M_1 \Theta_r + \lambda_1 \Theta_r^T D_1 \Theta_r + \Theta_r^T K_1 \Theta_r)\tilde{v}_1 = 0, \tag{5.30}$$

which is the eigenvalue problem of the matrix polynomial (5.26), where λ_1 is an eigenvalue of the polynomial. Conversely, we want to demonstrate that if $\tilde{v}_1$ is an eigenvector of $\bar{Q}(\lambda)$ to the corresponding eigenvalue λ_1, i.e., equation (5.30) holds, then $\begin{bmatrix} v_1^T & v_2^T \end{bmatrix}^T$ is an eigenvector of $Q(\lambda)$ with the same eigenvalue. Again plugging $\tilde{v}_1 = \Theta_l^T v_1$ in (5.30) and multiplying the resulting equation by Θ_l from the left we obtain

$$(\lambda_1^2 \Pi M_1 \Pi^T + \lambda_1 \Pi D_1 \Pi^T + \Pi K_1 \Pi^T) v_1 = 0, \tag{5.31}$$

Since the projector Π satisfies $\Pi^T v_1 = v_1$, (5.31) gives

$$\Pi(\lambda_1^2 M_1 v_1 + \lambda_1 D_1 v_1 + K_1 v_1) = 0,$$

which means that $\lambda_1^2 M_1 v_1 + \lambda_1 D_1 v_1 + K_1 v_1$ is in the nullspace of Π. We know that $\mathrm{Null}\,(\Pi) = \mathrm{Range}\left(G_1^T\right)$ (see Proposition 5.1 (2.)). Therefore, there exists a vector v_2 such that

$$\lambda_1 M_1 v_2 + K_1 v_1 + D_1 v_2 = -G_1^T v_2,$$

which implies

$$\lambda_1 M_1 v_2 + K_1 v_1 + D_1 v_2 + G_1^T v_2 = 0. \tag{5.32}$$

Again if $\Pi^T v_1 = v_1$, using Theorem 5.1 we have

$$G_1 v_1 = 0 \tag{5.33}$$

Equations (5.32) and (5.33) yield (5.28). □

Example: In order to show the equivalent finite spectra of the second order index 3 system (5.1) numerically, we consider the damped spring-mass system (DSMS) form [87]. See, e.g., Section 5.5 for details. Here we consider $n_\xi = g = 10$. As a result, the dimension of the second order index 3 model is 11. Using the MATLAB `polyeig` command we compute the eigenvalues of $Q(\lambda)$, $\tilde{Q}(\lambda)$ and $\bar{Q}(\lambda)$, respectively. As we can see in Table 5.1, by applying appropriate projectors to the index 3 system we can preserve all the finite eigenvalues.

$Q(\lambda)$	$\tilde{Q}(\lambda)$	$\bar{Q}(\lambda)$
∞	0	
∞	0	
-0.1220 ± 0.2876i	-0.1220 ± 0.2876i	-0.1220 ± 0.2876i
-0.1171 ± 0.2827i	-0.1171 ± 0.2827i	-0.1171 ± 0.2827i
-0.1000 ± 0.2646i	-0.1000 ± 0.2646i	-0.1000 ± 0.2646i
-0.0958 ± 0.2597i	-0.0958 ± 0.2597i	-0.0958 ± 0.2597i
-0.0270 ± 0.1445i	-0.0270 ± 0.1445i	-0.0270 ± 0.1445i
-0.0367 ± 0.1674i	-0.0367 ± 0.1674i	-0.0367 ± 0.1674i
-0.0423 ± 0.1789i	-0.0423 ± 0.1789i	-0.0423 ± 0.1789i
-0.0679 ± 0.2229i	-0.0679 ± 0.2229i	-0.0679 ± 0.2229i
-0.0663 ± 0.2206i	-0.0663 ± 0.2206i	-0.0663 ± 0.2206i
∞		
∞		

Table 5.1: Eigenvalues for the matrix polynomials $Q(\lambda)$, $\tilde{Q}(\lambda)$ and $\bar{Q}(\lambda)$, defined in (5.25-5.27).

5.3 Model reduction

5.3.1 Second-order-to-first-order reduction

Let us consider the second order index 3 system (5.1). In the preceding section, it is shown that this system can be converted into the equivalent form of the projected second order system (5.21). The first order transformed form of this second order projected system can be written as

$$\begin{aligned} \tilde{\Pi}E_1\tilde{\Pi}^T\dot{x}_1(t) &= \tilde{\Pi}A_1\tilde{\Pi}^T x_1(t) + \tilde{\Pi}B_s u(t), \\ y(t) &= C_s\tilde{\Pi}^T x_1(t), \end{aligned} \tag{5.34}$$

where

$$\begin{aligned} &\tilde{\Pi} = \begin{bmatrix} I_{n_\xi} & \\ & \Pi \end{bmatrix}, \quad E_1 = \begin{bmatrix} I_{n_\xi} & 0 \\ 0 & M_1 \end{bmatrix}, \quad A_1 = \begin{bmatrix} 0 & I_{n_\xi} \\ -K_1 & -D_1 \end{bmatrix}, \\ &B_s = \begin{bmatrix} 0 \\ H_1 \end{bmatrix}, \qquad C_s = \begin{bmatrix} L_1 & 0 \end{bmatrix} \quad \text{and} \quad x_1(t) = \begin{bmatrix} \xi(t) \\ \dot{\xi}(t) \end{bmatrix}. \end{aligned} \tag{5.35}$$

In system (5.34) all the coefficient matrices are singular since $\tilde{\Pi}$ has rank deficiency (due to the singularity of Π). This means the system contains redundant elements. To remove the redundant elements let us decompose $\tilde{\Pi}$ as

$$\tilde{\Pi} = \bar{\Theta}_l\bar{\Theta}_r^T, \quad \text{with} \quad \bar{\Theta}_l^T\bar{\Theta}_r = I_k, \tag{5.36}$$

where $\bar{\Theta}_l$, $\bar{\Theta}_r \in \mathbb{R}^{2n_\xi \times k}$ and $k = \operatorname{rank}\left(\tilde{\Pi}\right)$. Now applying the decomposition of $\tilde{\Pi}$ from (5.36) to (5.34) and defining $\tilde{x}_1(t) := \bar{\Theta}_l^T x_1(t)$, we obtain

$$\begin{aligned} \bar{\Theta}_r^T E_1\bar{\Theta}_r\dot{\tilde{x}}_1(t) &= \bar{\Theta}_r^T A_1\bar{\Theta}_r\tilde{x}_1(t) + \bar{\Theta}_r^T B_s u(t), \\ y(t) &= C_s\bar{\Theta}_r\tilde{x}_1(t). \end{aligned} \tag{5.37}$$

This system can be compared with the generalized state space system (2.1), and hence one can directly apply a naive approach of balanced truncation or IRKA based model reduction methods. Unfortunately, considering computational costs, forming (5.37) is prohibitive for a large scale system, since the actual computation of $\bar{\Theta}_l$ and $\bar{\Theta}_r$ by decomposing $\tilde{\Pi}$ is expensive. Moreover, the coefficient matrices in the system (5.37) are typically dense. Therefore, following the approaches as discussed in [70, 68] for first order index 2 systems, we apply balanced truncation and IRKA to the system (5.34) and compute the substantially reduced dimensional model

$$\begin{aligned} \hat{E}\dot{\hat{x}}(t) &= \hat{A}\hat{x}(t) + \hat{B}u(t), \\ \hat{y}(t) &= \hat{C}x(t). \end{aligned} \tag{5.38}$$

In the following we will show how to achieve this goal efficiently.

Balancing based technique:

Let us assume we want to apply the balanced truncation method to the system (5.36). Thus, we need to solve the Lyapunov equations

$$\bar{\Theta}_r^T A_1 \bar{\Theta}_r \bar{P} \bar{\Theta}_r^T E_1^T \bar{\Theta}_r + \bar{\Theta}_r^T E_1 \bar{\Theta}_r \bar{P} \bar{\Theta}_r^T A_1^T \bar{\Theta}_r = -\bar{\Theta}_r^T B_s B_s^T \bar{\Theta}_r, \tag{5.39a}$$

$$\bar{\Theta}_r^T A_1^T \bar{\Theta}_r \bar{Q} \bar{\Theta}_r^T E_1 \bar{\Theta}_r + \bar{\Theta}_r^T E_1^T \bar{\Theta}_r \bar{Q} \bar{\Theta}_r^T A_1 \bar{\Theta}_r = -\bar{\Theta}_r^T C_s^T C_s \bar{\Theta}_r, \tag{5.39b}$$

where $\bar{P}, \bar{Q} \in \mathbb{R}^{k \times k}$ are, respectively, the controllability and observability Gramians of the system (5.37). The solutions $\bar{P}$, $\bar{Q}$ of the Lyapunov equations are unique, since (according to Theorem 5.2) the corresponding system is asymptotically stable and symmetric positive (semi-)definite since the right hand side is semidefinite. Now multiplying both equations in (5.39) by $\bar{\Theta}_l$ from the left and $\bar{\Theta}_l^T$ from the right and exploiting the property in (5.36), we obtain

$$\tilde{\Pi} A_1 \tilde{\Pi}^T \tilde{P} \tilde{\Pi} E_1^T \tilde{\Pi}^T + \tilde{\Pi} E_1 \tilde{\Pi}^T \tilde{P} \tilde{\Pi} A_1^T \tilde{\Pi}^T = -\tilde{\Pi} B_s B_s^T \tilde{\Pi}^T, \tag{5.40a}$$

$$\tilde{\Pi} A_1^T \tilde{\Pi}^T \tilde{Q} \tilde{\Pi} E_1 \tilde{\Pi}^T + \tilde{\Pi} E_1^T \tilde{\Pi}^T \tilde{Q} \tilde{\Pi} A_1 \tilde{\Pi}^T = -\tilde{\Pi} C_s^T C_s \tilde{\Pi}^T, \tag{5.40b}$$

where

$$\tilde{P} = \bar{\Theta}_r \bar{P} \bar{\Theta}_r^T, \quad \tilde{Q} = \bar{\Theta}_r \bar{Q} \bar{\Theta}_r^T. \tag{5.41}$$

The Lyapunov equations in (5.40) are nothing but the Lyapunov equations of the projected system (5.34), where $\tilde{P}$, $\tilde{Q} \in \mathbb{R}^{2n_\xi \times 2n_\xi}$ are the systems controllability and observability Gramians. Under the condition (5.41) it can be shown that $\tilde{P}$ and $\tilde{Q}$ satisfy

$$\tilde{P} = \tilde{\Pi} \tilde{P} \tilde{\Pi}^T \quad \text{and} \quad \tilde{Q} = \tilde{\Pi} \tilde{Q} \tilde{\Pi}^T, \tag{5.42}$$

which ensures that the solutions are unique, although the equations in (5.40) are singular due to the singular projectors. The solution techniques of the projected Lyapunov equations (5.40) for computing the low-rank Gramian factors will be discussed in Section 5.4. Let $\tilde{R}$ and $\tilde{L}$ be the low-rank factors of the controllability and observability Gramians of the system (5.34) such that

$$\tilde{P} \approx \tilde{R} \tilde{R}^T, \quad \tilde{Q} \approx \tilde{L} \tilde{L}^T, \tag{5.43}$$

and $\bar{R}$ and $\bar{L}$ be the low-rank factors of the controllability and observability Gramians of the system (5.37) such that

$$\bar{P} \approx \bar{R} \bar{R}^T, \quad \bar{Q} \approx \bar{L} \bar{L}^T.$$

Then the controllability Gramian factors and the observability Gramian factors of the systems, (5.34) and (5.37), are related by

$$\tilde{R} = \bar{\Theta}_r \bar{R} \quad \text{and} \quad \tilde{L} = \bar{\Theta}_r \bar{L}. \tag{5.44}$$

This relation can easily be obtained, since

$$\begin{aligned}\tilde{R}\tilde{R}^T \approx \tilde{P} = \bar{\Theta}_r \bar{P} \bar{\Theta}_r^T \approx \bar{\Theta}_r \bar{R}\bar{R}^T \bar{\Theta}_r^T \quad \text{and} \\ \tilde{L}\tilde{L}^T \approx \tilde{Q} = \bar{\Theta}_r \bar{Q} \bar{\Theta}_r^T \approx \bar{\Theta}_r \bar{L}\bar{L}^T \bar{\Theta}_r^T.\end{aligned}$$

Let us consider the singular value decomposition of $\bar{L}^T \bar{\Theta}_r^T E_1 \bar{\Theta}_r \bar{R}$, i.e.,

$$\bar{L}^T \bar{\Theta}_r^T E_1 \bar{\Theta}_r \bar{R} = U \Sigma V^T.$$

Now construct the left and right *balancing and truncating* transformations $\bar{W}$ and $\bar{V}$ as

$$\bar{W} = \bar{L} U_1 \Sigma_1^{-\frac{1}{2}}, \quad \bar{V} = \bar{R} V_1 \Sigma_1^{-\frac{1}{2}},$$

where U_1, V_1 consist of the corresponding leading l ($l \ll k$) columns of U, V, and Σ_1 is the first leading $l \times l$ block of Σ. Again considering the singular value decomposition (using the Gramian factors of the system (5.34))

$$\tilde{L}^T E_1 \tilde{R} = \bar{R}^T \bar{Q}_r^T E_1 \bar{Q}_r \bar{L} = U \Sigma V^T, \tag{5.45}$$

we can construct the left and right balancing and truncating transformations as

$$\tilde{W} = \tilde{L} U_1 \Sigma_1^{-\frac{1}{2}}, \quad \tilde{V} = \tilde{R} V_1 \Sigma_1^{-\frac{1}{2}}. \tag{5.46}$$

We observe that

$$\begin{aligned}\tilde{W} = \tilde{L} U_1 \Sigma_1^{-\frac{1}{2}} = \bar{\Theta}_r \bar{L} U_1 \Sigma_1^{-\frac{1}{2}} = \bar{\Theta}_r \bar{W} = \bar{\Theta}_r \bar{\Theta}_l^T \bar{\Theta}_r \tilde{W} = \tilde{\Pi}^T \tilde{W}, \\ \tilde{V} = \tilde{R} U_1 \Sigma_1^{-\frac{1}{2}} = \bar{\Theta}_r \bar{R} U_1 \Sigma_1^{-\frac{1}{2}} = \bar{\Theta}_r \bar{V} = \bar{\Theta}_r \bar{\Theta}_l^T \bar{\Theta}_r \tilde{V} = \tilde{\Pi}^T \tilde{V}.\end{aligned} \tag{5.47}$$

Applying the balancing and truncating transformations $\bar{W}$ and $\bar{V}$ to the system (5.37), we can construct the reduced order model (5.38) where the coefficient matrices are formed by

$$\hat{E} = \bar{W}^T \bar{E} \bar{V}, \quad \hat{A} = \bar{W}^T \bar{A} \bar{V}, \quad \hat{B} = \bar{W}^T \bar{B}, \quad \text{and } \hat{C} = \bar{C} \bar{V}.$$

Close observation reveals that applying the property (5.47) the above reduced matrices can be computed efficiently by

$$\begin{aligned}\hat{E} = \bar{W}^T \bar{E} \bar{V} = \bar{W}^T \bar{\Theta}_r^T E_1 \bar{\Theta}_r \bar{V} = \tilde{W}^T \Pi E_1 \Pi^T \tilde{V} = \tilde{W}^T E_1 \tilde{V}, \\ \hat{A} = \bar{W}^T \bar{A} \bar{V} = \bar{W}^T \bar{\Theta}_r^T A_1 \bar{\Theta}_r \bar{V} = \tilde{W}^T \Pi A_1 \Pi^T \tilde{V} = \tilde{W}^T A_1 \tilde{V}, \\ \hat{B} = \bar{W}^T \bar{B} = \bar{W}^T \bar{\Theta}_r^T B_s = \tilde{W}^T \Pi B_s = \tilde{W}^T B_s, \\ \hat{C} = \bar{C} \bar{V} = C_s \bar{\Theta}_r \bar{V} = C_s \Pi^T \tilde{V} = C_s \tilde{V}.\end{aligned} \tag{5.48}$$

Therefore, from the above discussion it is clear that to obtain the reduced model (5.38) we need not form the system (5.34) or the system (5.37). We must just form the balancing and truncating transformations $\tilde{W}$, $\tilde{V}$ as given in (5.46) and then construct the reduced matrices as

$$\hat{E} = \tilde{W}^T E_1 \tilde{V}, \; \hat{A} = \tilde{W}^T A_1 \tilde{V}, \; \hat{B} = \tilde{W}^T B_s \quad \text{and} \quad \hat{C} = C_s \tilde{V}. \tag{5.49}$$

The procedure to compute the reduced first order system (5.38) from the second order index 3 DAEs (5.10) is summarized in Algorithm 13.

Algorithm 13: LR-SRM for second order index 3 systems.

```
Input  : $M_1,\ D_1,\ K_1,\ G_1,\ H_1,\ L_1$.
Output: $\hat{E},\ \hat{A},\ \hat{B},\ \hat{C}$.
Set up the matrices $E_1,\ A_1,\ B_s,\ C_s$ as in (5.34).
Compute the low-rank Gramian factors $\tilde{R}$, $\tilde{L}$ by solving the projected
  Lyapunov equations (5.40).
Construct the balancing and truncating transformations $\tilde{W}$ and $\tilde{V}$ by
  performing (5.45)-(5.46).
Form the reduced matrices using (5.49).
```

Interpolatory method via IRKA:

We concentrate on the interpolatory method via IRKA for model reduction of the system (5.1). The method that we follow here was introduced in [68] for first order index 2 DAEs. However, based on [68], the authors in [1] also discuss IRKA for such second order index 3 systems. In contrast to [1] our implementation procedure is different and more efficient, since inside the algorithm we show that the arising linear systems can be solved more efficiently by splitting, as we show in the following. To follow the procedure in [68], first convert the system (5.1) into the first order form (5.34). Note that the system (5.37) is an equivalent form of the system (5.34). Following the discussion in [68, Section 6.1] to construct a reduced system of the projected system (5.34) based on the interpolatory method, we need to construct the right and and left transformations defined in (2.47) as

$$\tilde{V} = \left[(\alpha_1\tilde{E} - \tilde{A})^I\tilde{B}b_1, \cdots, (\alpha_r\tilde{E} - \tilde{A})^I\tilde{B}b_r\right], \quad \text{and} \tag{5.50a}$$

$$\tilde{W} = \left[(\alpha_1\tilde{E}^T - \tilde{A}^T)^I\tilde{C}^Tc_1, \cdots, (\alpha_r\tilde{E}^T - \tilde{A}^T)^I\tilde{C}^Tc_r\right], \tag{5.50b}$$

where $\tilde{E} := \tilde{\Pi}E_1\tilde{\Pi}^T$, $\tilde{A} := \tilde{\Pi}A_1\tilde{\Pi}^T$, $\tilde{B} := \tilde{\Pi}B_s$, $\tilde{C} := C_s\tilde{\Pi}^T$, and $(\alpha_i\tilde{E} - \tilde{A})^I = (\alpha_i\tilde{E}^T - \tilde{A}^T)^I := \bar{\Theta}_r(\bar{\Theta}_r^TE_1\bar{\Theta}_r - \alpha_i\bar{\Theta}_r^TA_1\bar{\Theta}_r)^{-1}\bar{\Theta}_r^T$, for $i = 1, \cdots, r$. Recalling [68, Theorem 6.2], if we can construct $\tilde{V}$ and $\tilde{W}$ as in (5.50), then the reduced model in (5.38) can be formed by computing the reduced matrices as in (5.49). Therefore, to construct the reduced model (5.38) we can avoid the projectors. However, in each term of $\tilde{V}$ and $\tilde{W}$ the projectors are implicitly hidden. The solution of this problem has been shown in [68] based on [70, Theorem 5.2]. In our case each column of $\tilde{V}$ contains a vector such as

$$v = (\alpha\tilde{E} - \tilde{A})^I\tilde{B}b.$$

Following [68, Lemma 6.3], it can be shown (see also, e.g., Theorem 5.3) that the vector $v = \begin{bmatrix} v_1^T & v_2^T \end{bmatrix}^T$ solves

$$\begin{bmatrix} \alpha I_{n_\xi} & -I_{n_\xi} & 0 \\ K_1 & \alpha M_1 + D_1 & G_1^T \\ G_1 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \Gamma \end{bmatrix} = \begin{bmatrix} 0 \\ H_1 b \\ 0 \end{bmatrix}. \tag{5.51}$$

Again, simple algebraic manipulations to the linear system (5.51) leads us to solve

$$\begin{bmatrix} \alpha^2 M_1 + \alpha D_1 + K_1 & G_1^T \\ G_1 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ \Gamma \end{bmatrix} = \begin{bmatrix} H_1 b \\ 0 \end{bmatrix}, \tag{5.52}$$

for v_1 and then compute $v_2 = \alpha v_1$. Analogously, a vector

$$w = \begin{bmatrix} w_1^T & w_2^T \end{bmatrix}^T = (\alpha \tilde{E}^T - \tilde{A}^T)^I \tilde{C}^T c,$$

in each term in $\tilde{W}$ can be computed by solving the linear system

$$\begin{bmatrix} \alpha I_{n_\xi} & K_1^T & -G_1^T \\ I_{n_\xi} & \alpha M_1^T + D_1^T & 0 \\ 0 & G_1 & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \Gamma \end{bmatrix} = \begin{bmatrix} L_1^T c \\ 0 \\ 0 \end{bmatrix},$$

which is again equivalent to solving the linear system

$$\begin{bmatrix} -\alpha^2 M_1^T - \alpha D_1^T + K_1^T & -G_1^T \\ G_1 & 0 \end{bmatrix} \begin{bmatrix} \mathrm{w}_2 \\ \Gamma \end{bmatrix} = \begin{bmatrix} L_1^T c \\ 0 \end{bmatrix} \tag{5.53}$$

for w_2 and $w_1 = -(\alpha M_1^T + D_1^T) W_2$. A complete procedure to compute the reduced model (5.38) from the second order index 3 system (5.1) is presented in Algorithm 14.

5.3.2 Second-order-to-second-order reduction

Like the second order index 1 system in Chapter 4, in this section we contribute the BT and PDEG methods to obtain second order reduced systems from the second order index 3 descriptor systems. In both methods we must first convert the second order index 3 descriptor system (5.10) into the second order projected system (5.21). Then applying the model reduction techniques to the system (5.21) we obtain a reduced model

$$\begin{aligned} \hat{M}_1 \ddot{\hat{\xi}}(t) + \hat{D}_1 \dot{\hat{\xi}}(t) + \hat{K}_1 \hat{\xi}(t) &= \hat{H}_1 u(t), \\ \hat{y}(t) &= \hat{L}_1 \hat{\xi}(t). \end{aligned} \tag{5.54}$$

In the following we will show how to apply the aforementioned methods to the projected system (5.49) avoiding the projector (Π).

Algorithm 14: IRKA for second order index 3 systems.

```
Input : $M_1,\ D_1,\ K_1,\ G_1,\ H_1, L_1$.
Output: $\hat{E},\ \hat{A},\ \hat{B},\ \hat{C}$.
Setup the matrices $E_1,\ A_1,\ B_s,\ C_s$ as in (5.34).
Select initial interpolation points $\{\sigma_i\}_{i=1}^r$ and tangent directions $\{b_i\}_{i=1}^r$ and $\{c_i\}_{i=1}^r$.
while (not converged) do
    for $i = 1, 2, \cdots, r$ do
        $\begin{bmatrix} \alpha_i^2 M_1 + \alpha_i D_1 + K_1 & G_1^T \\ G_1 & 0 \end{bmatrix} \begin{bmatrix} v_i^{(1)} \\ \Gamma \end{bmatrix} = \begin{bmatrix} H_1 b_i \\ 0 \end{bmatrix}$,
        $\begin{bmatrix} -\alpha_i^2 M_1^T - \alpha_i D_1^T + K_1^T & G_1^T \\ G_1 & 0 \end{bmatrix} \begin{bmatrix} w_i^{(2)} \\ \Gamma \end{bmatrix} = \begin{bmatrix} L_1^T c_i \\ 0 \end{bmatrix}$,
        Form $\mathrm{v}_i = \begin{bmatrix} v_i^{(1)} \\ \alpha_i v_i^{(1)} \end{bmatrix}$, $\mathrm{w}_i = \begin{bmatrix} -(\alpha_i M_1^T + D_1^T) w_i^{(2)} \\ w_i^{(2)} \end{bmatrix}$.
        Construct $\tilde{V} = \begin{bmatrix} \mathrm{v}_1, \mathrm{v}_1, \cdots, \mathrm{v}_r \end{bmatrix}$, $\tilde{W} = \begin{bmatrix} \mathrm{w}_1, \mathrm{w}_1, \cdots, \mathrm{w}_r \end{bmatrix}$.
    end for
    $\hat{E} = \tilde{W}^T E_1 \tilde{V}$, $\hat{A} = \tilde{W}^T A_1 \tilde{V}$, $\hat{B} = \tilde{W}^T B_s$ and $\hat{C} = C_s \tilde{V}$
    Compute $\hat{A} z_i = \lambda_i \hat{E} z_i$ and $y^* \hat{A} = \lambda_i y^* \hat{E}$.
    $\alpha_i \leftarrow -\lambda_i$, $b_i^* \leftarrow y^* \hat{B}$ and $c_i \leftarrow \hat{C} z_i$.
end while
Form $\hat{E},\ \hat{A},\ \hat{B}$, and $\hat{C}$ with (5.49).
```

The BT method:

Chapter 2 discussed the second-order-to-second-order balancing criterion for second order systems, referring to the literature (e.g., [104, 114, 11, 97]). An overview of such techniques, all for standard second order systems, is found in [97]. For a second order index 1 system (4.1), of a slightly different form than (5.1), a second-order-to-second-order balancing technique is shown in Chapter 4. The fundamental procedure is the same as in Chapter 4. We can convert the index 3 system (5.10) to the ODE system (5.24). Thus, the balancing idea from [97] can be employed. As already mentioned, forming (5.24) is infeasible for a large-scale system. Therefore, instead of (5.24) we want to apply the BT technique to the equivalent system (5.21). For this purpose, recalling the discussion in Chapter 2, we need to solve the projected Lyapunov equations (5.40). We already know that the controllability and observability Gramians $\tilde{P}$ and $\tilde{Q}$ are the unique positive semi-definite solutions of the projected Lyapunov equations (5.40). Employ the block subdivision of the controllability and observability Gramians as in, e.g. [29]

$$\tilde{P} = \begin{bmatrix} \tilde{P}_{pp} & \tilde{P}_{pv} \\ \tilde{P}_{pv}^T & \tilde{P}_{vv} \end{bmatrix}, \qquad \tilde{Q} = \begin{bmatrix} \tilde{Q}_{pp} & \tilde{Q}_{pv} \\ \tilde{Q}_{pv}^T & \tilde{Q}_{vv} \end{bmatrix},$$

where $\tilde{P}_{pp},\ \tilde{Q}_{pp} \in \mathbb{R}^{n_\xi \times n_\xi}$ and $\tilde{P}_{vv},\ \tilde{Q}_{vv} \in \mathbb{R}^{n_\xi \times n_\xi}$ are called *position*, *velocity* controllability and observability Gramians, respectively. Using the LRCF-ADI iterations we can compute the low-rank controllability Gramian factor $\tilde{R}$ and observability Gramian factor $\tilde{L}$ defined in (5.43) by solving the Lyapunov equations (5.40). Due to the block structure of the $\tilde{P}$ and $\tilde{Q}$, the low-rank Gramian factors can be partitioned as (see [29]

$$\tilde{R} = \begin{bmatrix} \tilde{R}_p \\ \tilde{R}_v \end{bmatrix}, \qquad \tilde{L} = \begin{bmatrix} \tilde{L}_p \\ \tilde{L}_v \end{bmatrix},$$

where $\tilde{R}_p$, $\tilde{L}_p$ and $\tilde{R}_v$, $\tilde{L}_v$ denote the low-rank position, velocity controllability and observability Gramian factors, respectively. Let us consider the Gramian factors $\tilde{R}_p$ and $\tilde{L}_p$ to compute the thin SVD

$$\tilde{L}_p^T M_1 \tilde{R}_p = \begin{bmatrix} U_{pp,1} & U_{pp,1} \end{bmatrix} \begin{bmatrix} \Sigma_{pp,1} & 0 \\ 0 & \Sigma_{pp,2} \end{bmatrix} \begin{bmatrix} V_{pp,1}^T & V_{pp,2}^T \end{bmatrix}, \tag{5.55}$$

and construct the balancing and truncating transformations

$$W_s = \tilde{L}_p U_{pp,1} \Sigma_{pp,1}^{-\frac{1}{2}}, \quad V_s = \tilde{R}_p U_{pp,1} \Sigma_{pp,1}^{-\frac{1}{2}}. \tag{5.56}$$

Now applying W_s and V_s to the system (5.21) we can construct the reduced model (5.54). Like the second-order-to-first-order balancing based reduction method (see the discussion above), we can also prove the constructed balancing and truncating transformations are Π^T invariant, i.e., $\Pi^T V_s = V_s$ and $\Pi^T W_s = W_s$. Therefore, the coefficient matrices in (5.54) can be constructed as

$$\begin{aligned} &\hat{M}_1 = W_s^T M_1 V_s, \qquad \hat{D}_1 = W_s^T D_1 V_s, \\ &\hat{K}_1 = W_s^T M_1 V_s, \qquad \hat{H}_1 = W_s^T H_1, \qquad \hat{L}_1 = L_1 V_s, \end{aligned} \tag{5.57}$$

which prevent from constructing the projected system (5.21). The process of obtaining a reduced model by using a pair of low-rank controllability and observability position Gramian factors, i.e., $(\tilde{R}_p, \tilde{L}_p)$ is summarized in Algorithm 15. The constructed reduced model via $(\tilde{R}_p, \tilde{L}_p)$ is called position-position (PP) balancing. Likewise, the reduced models are called velocity-velocity (VV), velocity-position (VP), and position-velocity (PV) balancing if we use the pairs (R_v, L_v), (R_v, L_p) and (R_p, L_v), respectively,.

The PDEG method:

Here we want to construct the ROMs via projecting the system onto the dominant eigenspaces of the Gramians. The PDEG technique is introduced in Chapter 4 to

Algorithm 15: SOLR-SRM for second order index 3 system.

```
Input  : $M_1$, $D_1$, $K_1$, $H_1$, $L_1$.
Output: $\hat{M}_1$, $\hat{D}_1$, $\hat{K}_1$, $\hat{H}_1$, $\hat{L}_1$.
Solve the Lyapunov equations (5.40a) to compute $\tilde{R}_p$ and $\tilde{L}_p$.
Compute the balancing and truncating transformations as in (5.56).
Construct $\hat{M}_1$, $\hat{D}_1$, $\hat{K}_1$, $\hat{H}_1$, $\hat{L}_1$ following (5.57).
```

obtain second-order-to-second-order reduced models for second order index 1 systems, we follow the same procedure for the second order index 3 systems. We first convert the second order index 3 system (5.1) into the equivalent form of the second order projected system (5.21). In the above we already defined the Gramians for the second order projected system. Let us first consider the controllability position Gramian $\tilde{P}_{pp}$. Since $\tilde{P}_{pp}$ is symmetric positive (semi-)definite, it has the singular value decomposition,

$$\tilde{P}_{pp} = \tilde{U}_{pp}\tilde{\Sigma}_{pp}\tilde{V}_{pp}^T. \tag{5.58}$$

If $\operatorname{rank}\left(\tilde{P}_{pp}\right) = k$, where $k \ll n_\xi$, then the first k columns of $\tilde{U}_{pp}$ are the eigenvectors of $\tilde{P}_{pp}$. Now suppose $\tilde{R}_p$ is the low-rank factor (as defined above) of the controllability position Gramian $\tilde{P}_{pp}$ such that $\tilde{P}_{pp} \approx \tilde{R}_p\tilde{R}_p^T$. Compute the *thin* SVD

$$\tilde{R}_p = U_k\Sigma_k V_k^T, \tag{5.59}$$

where it can be proved that U_k consists of the first k columns of $\tilde{U}_{pp}$. Now forming

$$V_s = \begin{bmatrix} u_1, u_2, \cdots, u_r \end{bmatrix}, \tag{5.60}$$

where u_i, $i = 1, \cdots, r$, are the first r columns of U_k and applying V_s to the system (5.21), we can construct the r dimensional reduced model (5.54), where the reduced coefficient matrices can be formed as

$$\begin{aligned} \hat{M}_1 = V_s^T\Pi M_1\Pi^T V_s, \hat{D}_1 = V_s^T\Pi D_1\Pi^T V_s, \hat{K}_1 = V_s^T\Pi K_1\Pi^T V_s,\\ \hat{H}_1 = V_s^T\Pi H_1, \hat{L}_1 = L_1\Pi^T V_s. \end{aligned} \tag{5.61}$$

Applying Theorem 5.1 we have $\Pi^T V_s = V_s$, which implies $V_s^T\Pi = V_s^T$. Therefore, the reduced matrices in (5.54) can be constructed as

$$\hat{M}_1 = V_s^T M_1 V_s, \hat{D}_1 = V_s^T D_1 V_s, \hat{K}_1 = V_s^T K_1 V_s, \hat{H}_1 = V_s^T H_1, \hat{L}_1 = L_1 V_s. \tag{5.62}$$

The above procedure to compute the ROM via projecting the system onto the dominant eigenspace of the controllability position Gramian is summarized in Algorithm 16. A reduced model is obtained via projecting the system onto the dominant eigenspace of the controllability position Gramian, called PDEG-CP. Similarly, the reduced models are called PDEG-CV, PDEG-OP and PDEG-OV if the reduced models are obtained via projecting the system onto the eigenspaces of the controllability position, observability position and observability velocity Gramians, respectively.

Algorithm 16: PDEG for second order index 3 system.

```
Input  : $M_1$, $D_1$, $K_1$, $H_1$, $L_1$.
Output: $\hat{M}_1$, $\hat{D}_1$, $\hat{K}_1$, $\hat{H}_1$, $\hat{L}_1$.
Solve the Lyapunov equations (5.40a) to compute $\tilde{R}_p$.
Compute $V_s$ by performing (5.59-5.60).
Construct $\hat{M}_1$, $\hat{D}_1$, $\hat{K}_1$, $\hat{H}_1$, $\hat{L}_1$ following (5.62).
```

5.4 Solution of the projected Lyapunov equations

In the previous sections we noted that to implement the balancing and PDEG based model reduction methods for the second order index 3 descriptor system (5.1) we need to solve the projected Lyapunov equations (5.40) for computing the low-rank Gramian factors $\tilde{R}$ and $\tilde{L}$. This section discusses how to apply the LRCF-ADI method introduced in Chapter 2 to solve such projected Lyapunov equations efficiently. For convenience rewrite the projected Lyapunov equations (5.40) as

$$\tilde{A}\tilde{P}\tilde{E}^T + \tilde{E}\tilde{P}\tilde{A}^T = -\tilde{B}\tilde{B}^T, \tag{5.63a}$$

$$\tilde{A}^T\tilde{Q}\tilde{E} + \tilde{E}^T\tilde{Q}\tilde{A} = -\tilde{C}^T\tilde{C}, \tag{5.63b}$$

where $\tilde{E} = \tilde{\Pi}E_1\tilde{\Pi}^T$, $\tilde{A} = \tilde{\Pi}A_1\tilde{\Pi}^T$, $\tilde{B} = \tilde{\Pi}B_s$ and $\tilde{C} = C_s\tilde{\Pi}^T$. These Lyapunov equations look like the projected Lyapunov equations in (3.14) for the first order index 2 DAEs. An efficient solution of such projected Lyapunov equations is discussed in [70, Section 5] using the LRCF-ADI method for computing low-rank Gramian factors. This idea is updated in Chapter 3 including the concepts of computing the real low-rank Gramian factors and low-rank residual factor based stopping criterion. Here we are generalizing the ideas of Chapter 3 to solve the projected Lyapunov equation (5.63). In this section we mainly focus on two important issues. First we modify the GS-LRCF-ADI iteration in Chapter 3 for solving the projected Lyapunov equations (5.63). Second, we address the ADI shift parameter computation. We resolve some technical problems arising in the computation of both heuristic and adaptive shift parameters for the underlying system.

5.4.1 GS-LRCF-ADI iteration

Let us first concentrate on the solution of the controllability Lyapunov equation (5.63a) and recall Algorithm 7.

Initial residual factor: A close observation reveals that in this case the initial residual factor $\tilde{W}_0$ can be partitioned as

$$\tilde{W}_0 = \tilde{B} = \tilde{\Pi}B_s = \begin{bmatrix} I_{n_\xi} & \\ & \Pi \end{bmatrix} \begin{bmatrix} 0 \\ H_1 \end{bmatrix} = \begin{bmatrix} 0 \\ \Pi H_1 \end{bmatrix} = \begin{bmatrix} \tilde{W}_0^{(1)} \\ \tilde{W}_0^{(2)} \end{bmatrix}, \tag{5.64}$$

which gives $\tilde{W}_0^{(1)} = 0$ and $\tilde{W}_0^{(2)} = \Pi H_1$. We can compute $\tilde{W}_0^{(2)} = \Pi H_1$ efficiently using the following observation.

Lemma 5.1. *The matrix Ξ satisfies $\Xi = \Pi^T \Xi$ and $M_1 \Xi = \Pi H_1$, where Π is defined in (5.14) if and only if*

$$\begin{bmatrix} M_1 & G_1^T \\ G_1 & 0 \end{bmatrix} \begin{bmatrix} \Xi \\ \Lambda \end{bmatrix} = \begin{bmatrix} H_1 \\ 0 \end{bmatrix}. \tag{5.65}$$

Proof. For a proof, follow Lemma 3.1. □

Efficient solution of the linear systems: To solve the Lyapunov equation (5.63) using Algorithm 5, at the i-th iteration step we need to solve the linear system

$$(\tilde{A} + \mu_i \tilde{E}) V_i = \tilde{W}_{i-1}, \tag{5.66}$$

where $\tilde{W}_{i-1}$ is the ADI residual factor computed from the $(i-1)$-st iteration. Now partitioning $\tilde{W}_{i-1}$ as $\tilde{W}_{i-1} = \begin{bmatrix} \tilde{W}_{i-1}^{(1)} \\ \tilde{W}_{i-1}^{(2)} \end{bmatrix}$, equation (5.66) can be written as

$$\begin{bmatrix} \mu_i I_{n_\xi} & I_{n_\xi} \\ -\Pi K_1 \Pi^T & -\Pi D_1 \Pi^T + \mu_i \Pi M_1 \Pi^T \end{bmatrix} \begin{bmatrix} V_i^{(1)} \\ V_i^{(2)} \end{bmatrix} = \begin{bmatrix} \tilde{W}_{i-1}^{(1)} \\ \tilde{W}_{i-1}^{(2)} \end{bmatrix}. \tag{5.67}$$

Theorem 5.3. *Assume that χ_1 and χ_2 are in the nullspace of G_1. The matrix $\begin{bmatrix} \chi_1^T & \chi_2^T \end{bmatrix}^T$ satisfies the linear system*

$$\begin{bmatrix} \mu I_{n_\xi} & I_{n_\xi} \\ \Pi K_1 \Pi^T & \Pi D_1 \Pi^T + \mu \Pi M_1 \Pi^T \end{bmatrix} \begin{bmatrix} \chi_1 \\ \chi_2 \end{bmatrix} = \begin{bmatrix} F_1 \\ \Pi F_2 \end{bmatrix}, \tag{5.68}$$

iff the matrix $\begin{bmatrix} \chi_1^T & \chi_2^T & \Gamma^T \end{bmatrix}^T$ satisfies

$$\begin{bmatrix} \mu I_{n_\xi} & I_{n_\xi} & 0 \\ K_1 & D_1 + \mu M_1 & G_1^T \\ G_1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \chi_1 \\ \chi_2 \\ \Gamma \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \\ 0 \end{bmatrix}. \tag{5.69}$$

Proof. Suppose that $\begin{bmatrix} \chi_1^T \\ \chi_2^T \end{bmatrix}$ satisfies the linear system (5.68). From the second line of (5.68), we obtain

$$\Pi K_1 \Pi^T \chi_1 + (\Pi D_1 \Pi^T + \mu \Pi M_1 \Pi^T) \chi_2 = \Pi F_2. \tag{5.70}$$

Since $\Pi^T \chi_1 = \chi_1$, $\Pi^T \chi_2 = \chi_2$, this equation yields

$$\Pi (K_1 \chi_1 + D_1 \chi_2 + \mu M_1 \chi_2 - F_2) = 0, \tag{5.71}$$

i.e., $K_1\chi_1 + D_1\chi_2 + \mu M_1\chi_2 - F_2$ is in the nullspace of Π. Since $\mathrm{Null}\,(\Pi) = \mathrm{Range}\,(G_1^T)$, there exists a Γ, such that

$$K_1\chi_1 + D_1\chi_2 + \mu M_1\chi_2 - F_2 = -G_1^T\Gamma,$$

which implies

$$K_1\chi_1 + D_1\chi_2 + \mu M_1\chi_2 + G_1^T\Gamma = F_2. \tag{5.72}$$

Again if χ_1 is in the nullspace of G_1, i.e., $\Pi^T\chi_1 = \chi_1$, Theorem 5.1 gives

$$G_1\chi_1 = 0. \tag{5.73}$$

The first equation of (5.68) together with (5.72) and (5.73) produces the linear system (5.69). Conversely, suppose that $\begin{bmatrix}\chi_1^T & \chi_2^T & \Gamma^T\end{bmatrix}^T$ is the solution of (5.69). Then the second line of (5.69) gives (5.72). Using Theorem 5.1, the third line of (5.69) implies $\Pi^T\chi_1 = \chi_1$. It can be shown that $\Pi^T\chi_2 = \chi_2$, since $G_1\xi_1 = 0$ implies $G_1\xi_2 = 0$. Inserting these identities into (5.72), we obtain

$$K_1\Pi^T\chi_1 + D_1\Pi^T\chi_2 + \mu M_1\Pi^T\chi_2 + G_1^T\Gamma = F_2. \tag{5.74}$$

Multiply (5.74) from the left by Π. Since $\Pi G_1 = 0$, the resulting equation leads to (5.70). Together with the first line of (5.69), the result is the linear system (5.68). □

According to Theorem 5.3, instead of solving the linear system (5.67) we can solve the linear system

$$\begin{bmatrix}\mu I_{n_\xi} & I_{n_\xi} & 0\\ -K_1 & -D_1+\mu M_1 & -G_1^T\\ G_1 & 0 & 0\end{bmatrix}\begin{bmatrix}V_i^{(1)}\\ V_i^{(2)}\\ \Gamma\end{bmatrix} = \begin{bmatrix}\tilde{W}_{i-1}^{(1)}\\ \tilde{W}_{i-1}^{(2)}\\ 0\end{bmatrix}, \tag{5.75}$$

for $\begin{bmatrix}{V_i^{(1)}}^T {V_i^{(2)}}^T\end{bmatrix}^T$. The matrix (vector) $\begin{bmatrix}\tilde{W}_{i-1}^{(1)}\\ \tilde{W}_{i-1}^{(2)}\end{bmatrix}$ is updated in each iteration which is computed from the ADI residual factor of the previous step. For our problem, in each iteration, the ADI residual factor can be computed by (see, Step 6 in Algorithm 5)

$$\tilde{W}_i = \tilde{W}_{i-1} - 2\,\mathrm{Re}\,(\mu_i)\tilde{E}V_i,$$

which can be partitioned as

$$\begin{aligned}\begin{bmatrix}\tilde{W}_i^{(1)}\\ \tilde{W}_i^{(2)}\end{bmatrix} &= \begin{bmatrix}\tilde{W}_{i-1}^{(1)}\\ \tilde{W}_{i-1}^{(2)}\end{bmatrix} - 2\,\mathrm{Re}\,(\mu_i)\begin{bmatrix}I_{n_\xi} & 0\\ 0 & \Pi M_1\Pi^T\end{bmatrix}\begin{bmatrix}V_i^{(1)}\\ V_i^{(2)}\end{bmatrix}.\\ &= \begin{bmatrix}\tilde{W}_{i-1}^{(1)} - 2\,\mathrm{Re}\,(\mu_i)V_i^{(1)}\\ \tilde{W}_{i-1}^{(2)} - 2\,\mathrm{Re}\,(\mu_i)\Pi M_1\Pi^T V_i^{(2)}\end{bmatrix}.\end{aligned}$$

Exploiting the properties of Π, i.e., $\Pi M_1 = M_1\Pi^T$ and $\Pi^T V_i^{(2)} = V_i^{(2)}$, the above equation results in

$$\begin{aligned}\tilde{W}_i^{(1)} &= \tilde{W}_{i-1}^{(1)} - 2\,\mathrm{Re}\,(\mu_i)V_i^{(1)},\\ \tilde{W}_i^{(2)} &= \tilde{W}_{i-1}^{(2)} - 2\,\mathrm{Re}\,(\mu_i)M_1V_i^{(2)}.\end{aligned} \tag{5.76}$$

We already mentioned earlier that the initial residual factors should be $\tilde{W}_0^{(1)} = 0$ and $\tilde{W}_0^{(2)} = \Pi H_1$, where $\tilde{W}_0^{(2)}$ can be computed cheaply by using Lemma 5.1. If the two consecutive shift parameters are complex conjugates of each other, i.e., $\{\mu_i, \mu_{i+1} := \overline{\mu_i}\}$, then recalling (2.62) we see

$$\tilde{W}_{i+1} = \tilde{W}_{i-1} - 4\,\mathrm{Re}\,(\mu_i)\tilde{E}\left(\mathrm{Re}\,(V_i) + \delta\,\mathrm{Im}\,(V_i)\right),$$

where $\delta = \frac{\mathrm{Re}\,(\mu_i)}{\mathrm{Im}\,(\mu_i)}$, which gives

$$\begin{aligned}\begin{bmatrix}\tilde{W}_{i+1}^{(1)}\\ \tilde{W}_{i+1}^{(2)}\end{bmatrix} &= \begin{bmatrix}\tilde{W}_{i-1}^{(1)}\\ \tilde{W}_{i-1}^{(2)}\end{bmatrix} - 4\,\mathrm{Re}\,(\mu_i)\begin{bmatrix}I_{n_\xi} & 0\\ 0 & \Pi M_1\Pi^T\end{bmatrix}\begin{bmatrix}\mathrm{Re}\,(V_i^{(1)}) + \delta\,\mathrm{Im}\,(V_i^{(1)})\\ \mathrm{Re}\,(V_i^{(2)}) + \delta\,\mathrm{Im}\,(V_i^{(2)})\end{bmatrix}\\ &= \begin{bmatrix}\tilde{W}_{i-1}^{(1)} - 4\,\mathrm{Re}\,(\mu_i)(\mathrm{Re}\,(V_i^{(1)}) + \delta\,\mathrm{Im}\,(V_i^{(1)}))\\ \tilde{W}_{i-1}^{(2)} - 4\,\mathrm{Re}\,(\mu_i)\Pi M_1\Pi^T(\mathrm{Re}\,(V_i^{(2)}) + \delta\,\mathrm{Im}\,(V_i^{(2)}))\end{bmatrix}.\end{aligned}$$

Again following the properties of Π (i.e., Proposition 5.1 and Theorem 5.1), the above relation results in

$$\begin{aligned}\tilde{W}_{i+1}^{(1)} &= \tilde{W}_{i-1}^{(1)} - 4\,\mathrm{Re}\,(\mu_i)(\mathrm{Re}\,(V_i^{(1)}) + \delta\,\mathrm{Im}\,(V_i^{(1)})),\\ \tilde{W}_{i+1}^{(2)} &= \tilde{W}_{i-1}^{(2)} - 4\,\mathrm{Re}\,(\mu_i)M_1(\mathrm{Re}\,(V_i^{(2)}) + \delta\,\mathrm{Im}\,(V_i^{(2)})).\end{aligned} \tag{5.77}$$

Now let us see how to split the linear system (5.75) to accelerate the solution. From the first line of (5.75), we obtain

$$V_i^{(2)} = \tilde{W}_{i-1}^{(1)} - \mu_i V_i^{(1)}. \tag{5.78}$$

Inputting this into the second line of (5.75), we obtain

$$(K_1 + \mu_i D_1 + \mu_i^2 M_1)V_i^{(1)} + G_1^T\Gamma = (\mu_i M_1 - D_1)\tilde{W}_{i-1}^{(1)} - \tilde{W}_{i-1}^{(2)}. \tag{5.79}$$

Together with (5.79) and the third line of (5.75), we obtain

$$\begin{bmatrix}K_1 + \mu_i D_1 + \mu_i^2 M_1 & G_1^T\\ G_1 & 0\end{bmatrix}\begin{bmatrix}V_i^{(1)}\\ \Gamma\end{bmatrix} = \begin{bmatrix}(\mu_i M_1 - D_1)\tilde{W}_{i-1}^{(1)} - \tilde{W}_{i-1}^{(2)}\\ 0\end{bmatrix}. \tag{5.80}$$

Applying the above splitting approach to the linear system (5.75), the solution becomes much faster. The procedure to compute the low-rank controllability Gramian factor by solving the controllability Lyapunov equation (5.63a) is stated in Algorithm 17.

Algorithm 17: SOGS-LRCF-ADI for the second order index 3 systems.

```
   Input  : $M_1,\ D_1,\ K_1,\ G_1,\ H_1,\ \{\mu_i\}_{i=1}^{J}$.
   Output: $\tilde{R} = Z_i$, such that $\tilde{P} \approx \tilde{R}\tilde{R}^T$.
 1 Set $Z_0 = [\,]$, $i = 1$ and $\tilde{W}_0^{(1)} = 0$.
 2 Solve (5.65) for $\Xi$ and compute $\tilde{W}_0^{(2)} = M_1\Xi$.
 3 while $\|\tilde{W}_0^{(1)T}\tilde{W}_0^{(1)} + \tilde{W}_0^{(2)T}\tilde{W}_0^{(2)}\| \geq tol$ and $i \leq i_{max}$ do
 4     Solve $\begin{bmatrix} K_1 + \mu_i D_1 + \mu_i^2 M_1 & G_2 \\ G_1 & 0 \end{bmatrix} \begin{bmatrix} V_i^{(1)} \\ \Gamma \end{bmatrix} = \begin{bmatrix} (\mu_i M_1 - D_1)\tilde{W}_{i-1}^{(1)} - \tilde{W}_{i-1}^{(2)} \\ 0 \end{bmatrix}$.
 5     Compute $V_i^{(2)} = \tilde{W}_{i-1}^{(1)} - \mu_i V_i^{(1)}$ and $V_i = \begin{bmatrix} {V_i^{(1)}}^T & {V_i^{(2)}}^T \end{bmatrix}^T$.
 6     if $\mathrm{Im}\,(\mu_i) = 0$ then
 7         $Z_i = \begin{bmatrix} Z_{i-1} & \sqrt{-2\mu_i}\,\mathrm{Re}\,(V_i) \end{bmatrix}$,
 8         $\tilde{W}_i^{(1)} = \tilde{W}_{i-1}^{(1)} - 2\mu_i V_i^{(1)}, \quad \tilde{W}_i^{(2)} = \tilde{W}_{i-1}^{(2)} - 2\mu_i M_1 V_i^{(2)}$.
 9     else
10         $\gamma = -2\,\mathrm{Re}\,(\mu_i)$, $\delta = \frac{\mathrm{Re}\,(\mu_i)}{\mathrm{Im}\,(\mu_i)}$,
11         $Z_{i+1} = \begin{bmatrix} Z_{i-1} & \sqrt{2\gamma}(\mathrm{Re}\,(V_i) + \delta\,\mathrm{Im}\,(V_i)) & \sqrt{2\gamma}\sqrt{(\delta^2+1)}\,\mathrm{Im}\,(V_i) \end{bmatrix}$,
12         $V_{i+1}^{(1)} = \mathrm{Re}\,(V_i^{(1)}) + \delta\,\mathrm{Im}\,(V_i^{(1)}), \quad V_{i+1}^{(2)} = \mathrm{Re}\,(V_i^{(2)}) + \delta\,\mathrm{Im}\,(V_i^{(2)})$,
13         $\tilde{W}_{i+1}^{(1)} = \tilde{W}_{i-1}^{(1)} + 2\gamma V_{i+1}^{(1)}, \quad \tilde{W}_{i+1}^{(2)} = \tilde{W}_{i-1}^{(2)} + 2\gamma M_1 V_{i+1}^{(2)}$.
14         $i = i + 1$
15     end if
16     $i = i + 1$
17 end while
```

Solution of observability Lyapunov equation: Algorithm 17, can solve the projected observability Lyapunov equation (5.63) considering a few changes. First, in the input matrices replace H_1 by L_1. The initial residual factors are $\tilde{W}_0^{(1)} = \Pi L^T$, $\tilde{W}_0^{(2)} = 0$. Replacing H_1 by L_1^T, we solve the linear system (5.65) for Ξ to compute $W_0^{(1)} = M_1\Xi$. In Step 4, we solve the linear system

$$\begin{bmatrix} \mu_i D_1^T - \mu_i^2 M_1^T - K_1^T & G_1^T \\ G_1 & 0 \end{bmatrix} \begin{bmatrix} V_i^{(2)} \\ \Gamma \end{bmatrix} = \begin{bmatrix} \tilde{W}_{i-1}^{(1)} - \mu_i \tilde{W}_{i-1}^{(2)} \\ 0 \end{bmatrix}$$

for $V_i^{(2)}$. In Step 5, compute $V_i^{(1)} = \tilde{W}_{i-1}^{(2)} - (\mu_i M_1^T - D_1^T)V_i^{(2)}$ for $V_i = \begin{bmatrix} {V_i^{(1)}}^T & {V_i^{(2)}}^T \end{bmatrix}^T$. Applying these changes in the algorithm, compute $\tilde{L} = Z_i$ such that $\tilde{Q} \approx \tilde{L}\tilde{L}^T$.

5.4.2 ADI shift parameter selection

Algorithm 17 depends on certain shift parameters that are crucial for fast convergence of the method. We investigate two types of ADI shift parameters. Penzl's heuristic approach introduced in [93] is one of the most commonly used approaches to compute the ADI shift parameters for a large-scale system. Recently another approach is introduced in [21] to compute the ADI shifts adaptively. Both approaches were introduced in Chapter 2. We proposed an update on the adaptive shift selection approach in Chapters 3 which is considered here as well. This section focuses on some technical problems arising in both approaches for the system considered in this chapter.

Heuristic shifts: In this approach, we often need a set of approximate finite eigenvalues which consist of some large magnitude and small magnitude Ritz values of the matrix pencil corresponding to the underlying system (see, e.g., Penzl's heuristic in [93]). For the second order index 3 descriptor system (5.10) the corresponding matrix pencil is

$$\lambda \underbrace{\begin{bmatrix} I_{n_\xi} & 0 & 0 \\ 0 & M_1 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{\check{E}} - \underbrace{\begin{bmatrix} 0 & I_{n_\xi} & 0 \\ -K_1 & -D_1 & -G_1^T \\ G_1 & 0 & 0 \end{bmatrix}}_{\check{A}_3}. \tag{5.81}$$

Due to the singularity of $\check{E}$, the matrix pencil features some infinite eigenvalues that prevent the direct usage of Arnoldi's method for the approximation of large magnitude eigenvalues. To overcome this problem, we can employ the strategy introduced in [39], looking at the modified eigenvalue problem of the first order structured index 2 DAEs system such as (3.1). Following the strategy, we modify the matrix pencil (5.81) as

$$\lambda \underbrace{\begin{bmatrix} I_{n_\xi} & 0 & 0 \\ 0 & M_1 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{\check{E}} - \underbrace{\begin{bmatrix} 0 & I_{n_\xi} & 0 \\ -K_1 & -D_1 & -G_1^T \\ 0 & G_1 & 0 \end{bmatrix}}_{\check{A}_2}. \tag{5.82}$$

The matrix pencil (5.82) has the same structure as the matrix pencil corresponding to the system (3.1). Moreover, according to [48, Theorem 2.7.3][1], the matrix pencils in (5.82) and (5.81) share the same non-zero finite spectrum. Now the modified matrix pencil

$$\lambda \underbrace{\begin{bmatrix} I_{n_\xi} & 0 & 0 \\ 0 & M_1 & -\alpha G_1^T \\ 0 & \alpha G_1 & 0 \end{bmatrix}}_{\check{E}_\alpha} - \underbrace{\begin{bmatrix} 0 & I_{n_\xi} & 0 \\ -K_1 & -D_1 & -G_1^T \\ 0 & G_1 & 0 \end{bmatrix}}_{\check{A}_2} \tag{5.83}$$

moves all infinite eigenvalues to $\frac{1}{\alpha}$ ($\alpha \in \mathbb{R}$) (see e.g., Chapter 3), without altering the finite eigenvalues. The parameter α can be chosen such that $\frac{1}{\alpha}$ is close to the

[1] The theorem originates in [107] but we prefer the textbook reference.

smallest magnitude eigenvalues after those have been determined with respect to the original matrices. Note, the matrix $\check{A}_2$ in (5.83) will always be singular, such that the small eigenvalue approximations cannot be computed since the inversion of $\check{A}_2$ would be required. Therefore, small magnitude Ritz values should be always computed from the matrix pencil (5.81).

Adaptive shifts: A second shift computation strategy, which is rather simple and more efficient, is already discussed in Chapters 3 and 4. Here the computed shift parameters are associated to the projected system (5.34) in which the corresponding matrix pencil is

$$\lambda\tilde{\Pi}E_1\tilde{\Pi}^T - \tilde{\Pi}A_1\tilde{\Pi}^T. \tag{5.84}$$

From the deliberation of Section 5.2 we already know the matrix pencil (5.84) incorporates all of the finite eigenvalues of the index 3 system (5.10). For the initialization of the shifts, we can proceed with the same procedure given in [21] as long as the system has sufficiently many inputs and outputs. If the input or output matrix consists of only a few columns, particularly, for SISO systems, sometimes we may not achieve any stable eigenvalue from the projected pencil of (5.84). To overcome this problem, we propose a different initialization technique. Instead of using $\tilde{W}_0$ to project the pencil (5.84), we want to use a random thin rectangular matrix $\check{B} \in \mathbb{R}^{2n_\xi \times k}$, where $k \ll n_\xi$. For the updated shifts, we follow the same procedure as discussed in Chapter 4.

5.5 Numerical results

5.5.1 Test examples and hardware

To assess the accuracy and efficiency of the proposed model reduction methods, we illustrate numerical results for two model examples. The first example is a holonomically constrained damped spring-mass system (DSMS) [87] as shown in Figure 5.1. The i-th mass of weight m_i is connected to the $(i-1)$-st mass by a spring and a damper with constants k_i and δ_i, respectively. Moreover, the first mass is connected to the last one by a rigid bar and influenced by the control $u(t)$. M_1 is a diagonal mass matrix, $K_1 \in \mathbb{R}^{n_\xi \times n_\xi}$ and D_2 both are $n_\xi \times n_\xi$ dimensional tridiagonal stiffness and damping matrices, respectively. $G_1 = [1, 0, \cdots, 0, -1] \in \mathbb{R}^{1\times n_\xi}$ is the constraint matrix, $H_1 = e_1$ and $L_1 = [e_1, e_2, e_{n_\xi - 1}]^T$, where e_i denotes the $i-$th

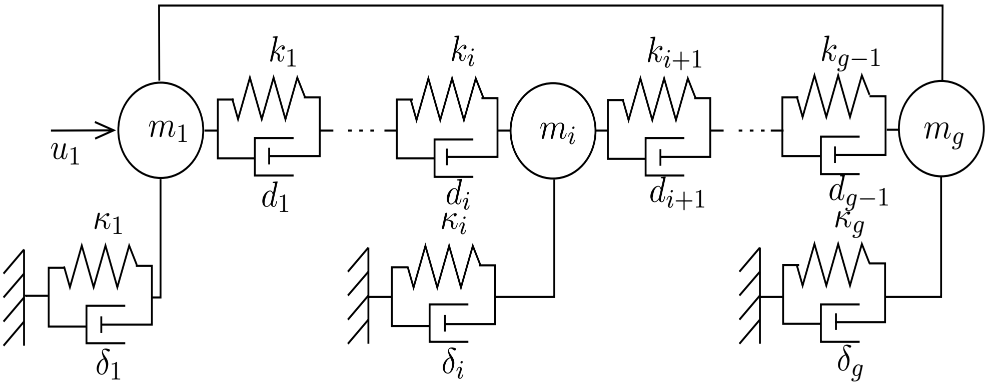


Figure 5.1: A spring-mass-damper system with holonomic constraint (source [87]).

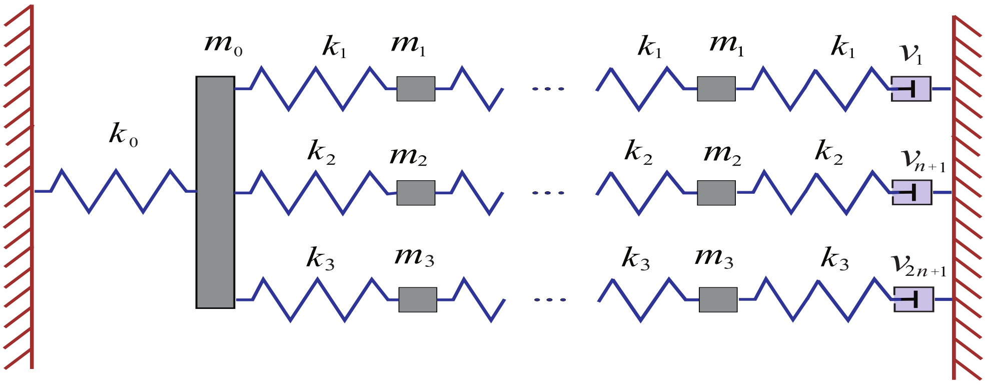


Figure 5.2: Triple chain oscillation (source [118]).

column of the identity matrix I_{n_ξ}. In our experiments we take $M_1 = 100\, I_{n_\xi}$ and

$$K_1 = \begin{bmatrix} -6 & 2 & & & \\ 2 & -6 & 2 & & \\ & 2 & -6 & \ddots & \\ & & \ddots & \ddots & 2 \\ & & & 2 & -6 \end{bmatrix}, \qquad D_1 = \begin{bmatrix} -15 & 5 & & & \\ 5 & -15 & 5 & & \\ & 5 & -15 & \ddots & \\ & & \ddots & \ddots & 5 \\ & & & 5 & -15 \end{bmatrix}.$$

For our numerical test we consider $n_\xi = 10\,000$ masses. Therefore, we obtain a $10\,001$ dimensional second order index 3 system. The model has 1 input and 3 outputs. Note that in Figure 5.1, $g = n_\xi$.

The second example is a triple chain oscillator model (TCOM) as shown in Figure 5.2. This example originates in [119] with the setup described in [102] which results in ODEs. To transform it into an index 3 DAEs, the following holonomic constraints are considered. The constraint matrix $G_1 \in \mathbb{R}^{n_\varphi \times n_\xi}$ is chosen as a random sparse matrix. In this particular test example, there are $2\,000$ masses and $5\,000$

models	sizes	tolerance	no. of iterations			
			heuristic shifts		adaptive shifts	
			$\tilde{R}$	$\tilde{L}$	$\tilde{R}$	$\tilde{L}$
DSMS	10001	10^{-8}	80	92	26	31
TCOM	11001	10^{-8}	270	282	153	240

Table 5.2: The performances of the heuristic and adaptive shifts in the GS-LRCF-ADI method with respect to iteration number.

model size	CPU time (sec)				
	heuristic shift			adaptive shift	
	$\tilde{R}$	$\tilde{L}$	μ	$\tilde{R}$	$\tilde{L}$
DSMS	2.27	3.84	119	1.16	1.56
TCOM	2.41	3.55	35	2.19	3.45

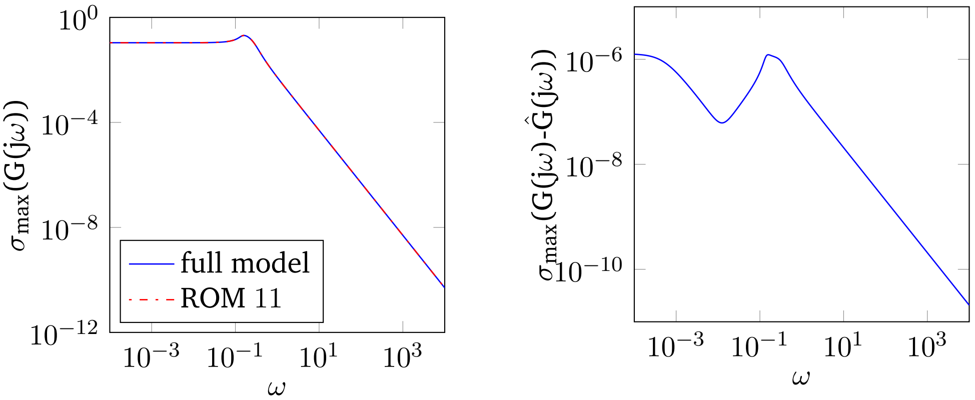
Table 5.3: The performances of the heuristic and adaptive shifts in the GS-LRCF-ADI method with respect to computational time.

constraints. Therefore, G_1 becomes a $5\,000 \times 6\,001$ matrix. Here we consider the $2\,000$-th off-diagonal and $4\,000$-th diagonal elements of G_1 are all 1 and -1, respectively. The dimension of the second order index 3 system is $11\,001$. The input and output matrices $H_1 \in \mathbb{R}^{n_\xi \times 1}$, $L_1 \in \mathbb{R}^{1 \times n_\xi}$ are chosen randomly.

All the results were obtained using MATLAB 7.11.0 (R2012a) on a board with 2 Intel® Xeon® X5650 CPUs with a 2.67 GHz clock speed, 6 Cores each and 48 GB of total RAM.

5.5.2 GS-LRCF-ADI iteration

In order to perform the BT and PDEG based techniques, we must compute the low-rank control and observability Gramian factors $\tilde{R}$ and $\tilde{L}$. These Gramian factors are computed by applying Algorithm 17. To execute this algorithm we use both heuristic and adaptive ADI shift parameters. The comparison of the heuristic and adaptive shifts for both model examples is shown in Tables 5.2. In Table 5.3 the comparison is shown in terms of computational time. From both tables, we can conclude that in both cases (number of iterations taken to converge within the given tolerance and execution time), the adaptive shifts perform better than the heuristic shifts. Note that for the model DSMS, we selected 15 optimal heuristic shifts out of 30 large and 25 small magnitude Ritz-values. On the other hand, for the model TCOM, from 50 large and 180 small magnitude Ritz-values, 100 heuristic shifts were selected. For the adaptive shifts, in each cycle, we were restricted to



(a) Sigma plot.

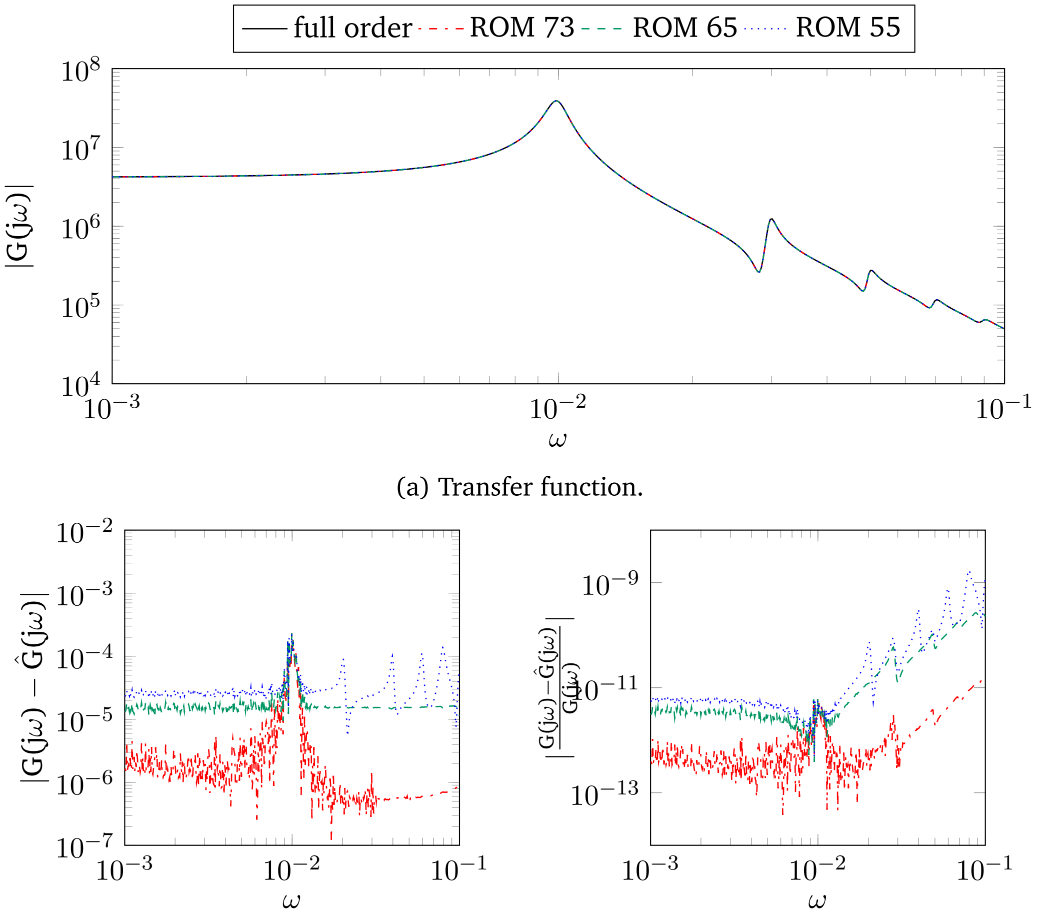
(b) Absolute error.

Figure 5.3: Comparison of full and first order reduced model of the DSMS example.

10 proper shift parameters for the DSMS model. In case of the TCOM model, the number of adaptive shifts was 70.

5.5.3 Second-order-to-first-order reduction

First we apply Algorithm 13 to the DSMS example, which generates a 11 dimensional standard state space model, using the truncation tolerance 10^{-5}. Figure 5.3a shows the sigma plot, i.e., the maximum singular values of the transfer function matrix of the full and reduced models on a wide frequency domain, i.e., 10^{-4}Hz to 10^{4}Hz. The corresponding absolute error between the full and reduced dimensional models is shown in Figure 5.3b. We observe that the error is below the MOR tolerance already for a very low dimensional model. When the same algorithm is applied to the system TCOM model, we obtain a 73 dimensional reduced system for the truncation tolerance 10^{-5}. However, the dimension of the ROM can be reduced further by using higher truncation tolerances. For instance, 10^{-4} and 10^{-3} truncation tolerances generate respectively, 65 and 55 dimensional reduced systems. The comparison of the full and different dimensional reduced systems are shown in Figure 5.4. This figure depicts that the frequency responses of the full and different dimensional reduced systems are matching nicely and both errors indicate good accuracy. We also show the time domain simulation of the full and 55 dimensional reduced models and their respective errors in Figure 5.5. From the absolute (Figure 5.5b) and the relative deviation (Figure 5.5c) we can conclude that the proposed methods can produce a good reduced system. To compare the balancing based method with IRKA, we compute 60, 50, 40, 30, and 20 dimensional reduced models using both Algorithms 13 and 14. The absolute and relative deviations between the full and 60 dimensional reduced models are shown in Figure 5.6. On the other hand, Table 5.4 lists the absolute and relative H_∞ norm of the error systems for 50, 40, 30, and 20 dimensional ROMs. From the figure and table, one



(a) Transfer function.

(b) Absolute error.

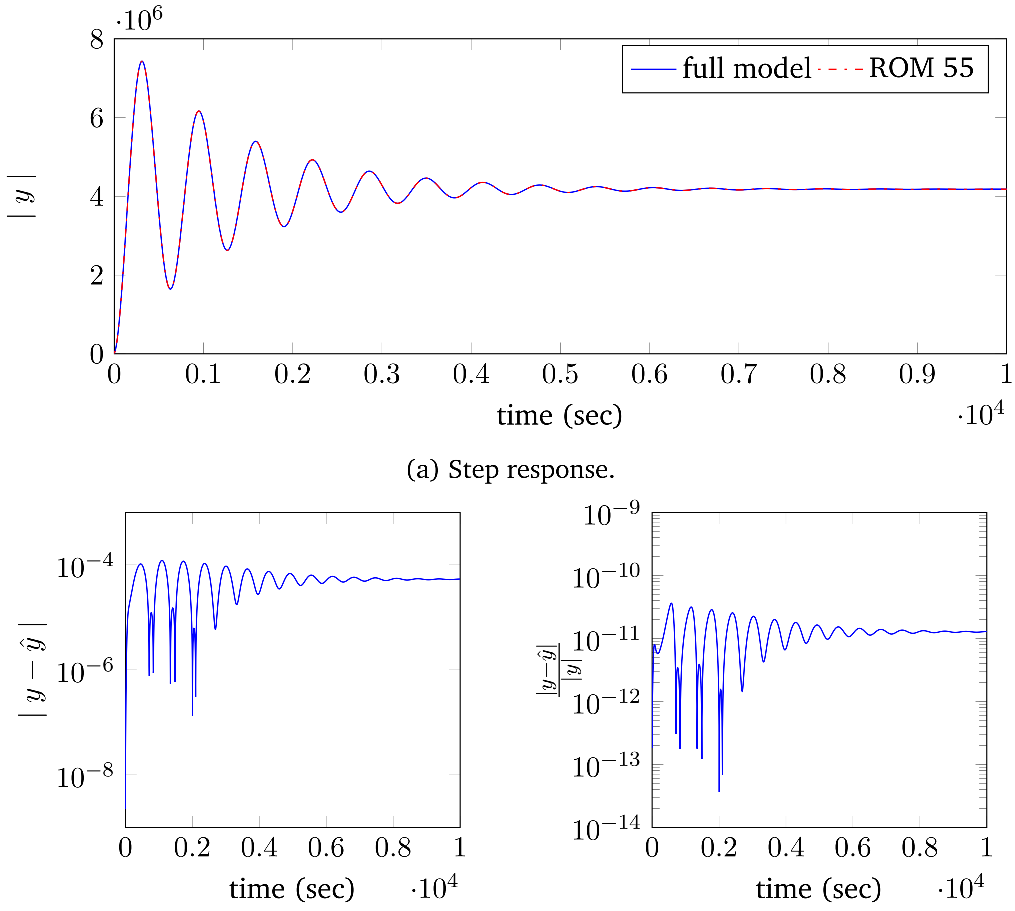
(c) Relative error.

Figure 5.4: Comparison of different dimensional first order reduced and original models in the frequency domain for the TCOM example.

can observe that the balancing based methods generate more accurate ROMs.

5.5.4 Second-order-to-second-order-reduction

We consider an 11001 dimensional second order index 3 model for the TCOM example. To compute the Gramian factors, we follow the same strategy discussed above. Applying Algorithm 15, we compute a 44 dimensional reduced order model via balancing the system on the position-position level. The same algorithm generates 41, 44, and 38 dimensional reduced systems via balancing the system onto velocity-velocity, position-velocity, and velocity-position levels. In all cases, the truncation tolerance is set to 10^{-3}. The frequency responses of the full and the reduced models and their absolute and relative errors are shown in Figure 5.7. Although the accuracy is not satisfactory for the approximated models on the position-position and



(a) Step response.

(b) Absolute error.

(c) Relative error.

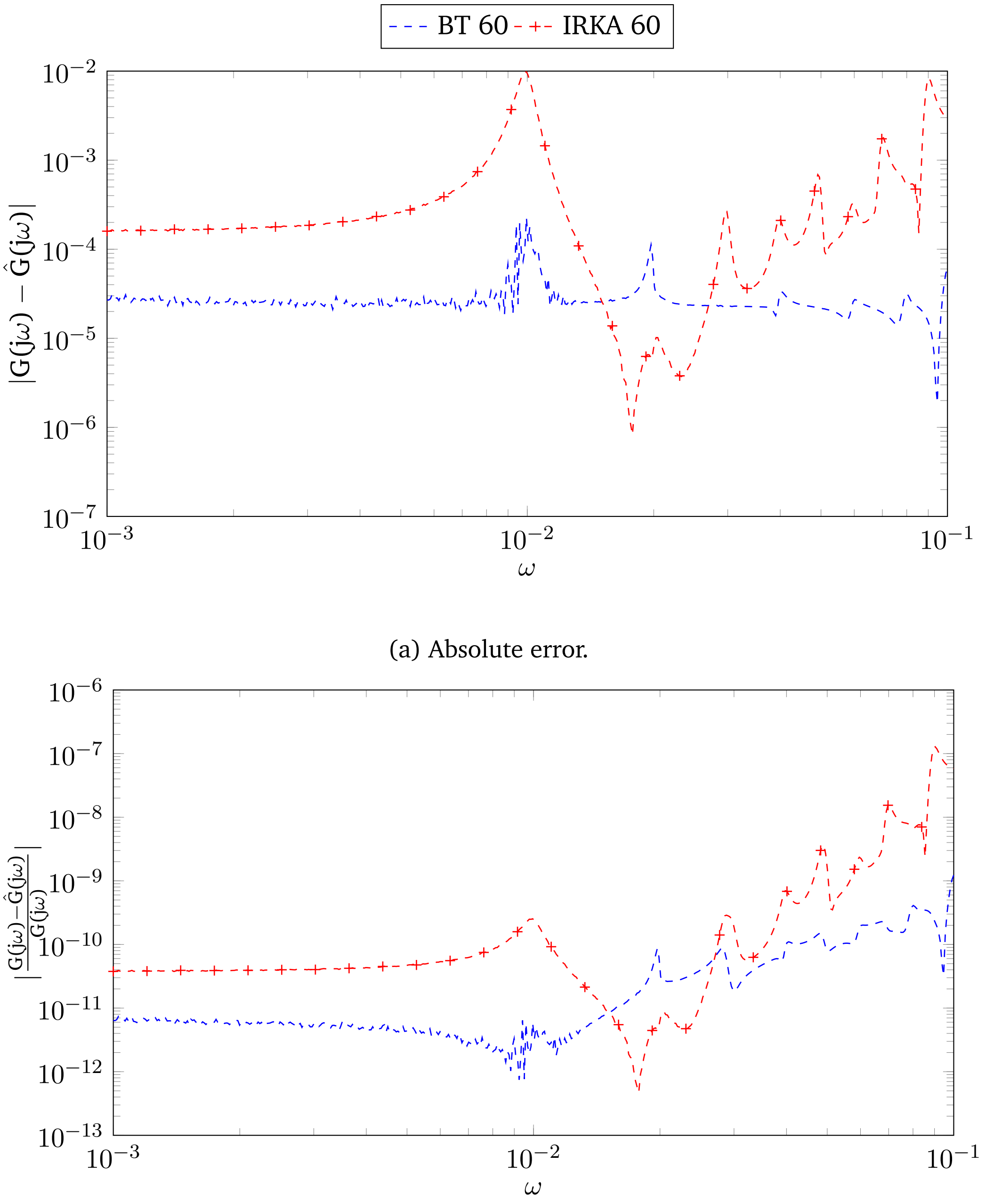
Figure 5.5: Comparison of full 55 dimensional ROMs for the TCOM example in the time domain.

the velocity-position levels, for the other balancing levels the accuracy is fine. Note that, although this feature appears for this particular model example, we can see in Figure 5.8, for the other test examples all of the balancing levels give reduced systems with good accuracy.

We also apply the PDEG method to the TCOM model. In this case, we construct 45 dimensional reduced models by projection of the systems onto the dominant eigenspaces of the different Gramians. Figure 5.9 shows very good accuracy of all the ROMs computed by the PDEG method. Moreover, the constructed ROMs by this method, ensures the stability of the original model, which is reflected in Figure 5.10.

dimension of ROM	H_∞ norm			
	absolute		relative	
	BT	IRKA	BT	IRKA
50	2.88×10^{-4}	8.00×10^{-2}	3.77×10^{-9}	1.25×10^{-6}
40	9.00×10^{-3}	1.33×10^{0}	1.70×10^{-7}	2.42×10^{-5}
30	6.19×10^{-1}	6.29×10^{2}	1.24×10^{-5}	1.00×10^{-2}
20	1.17×10^{1}	9.59×10^{2}	2.35×10^{-4}	1.50×10^{-2}

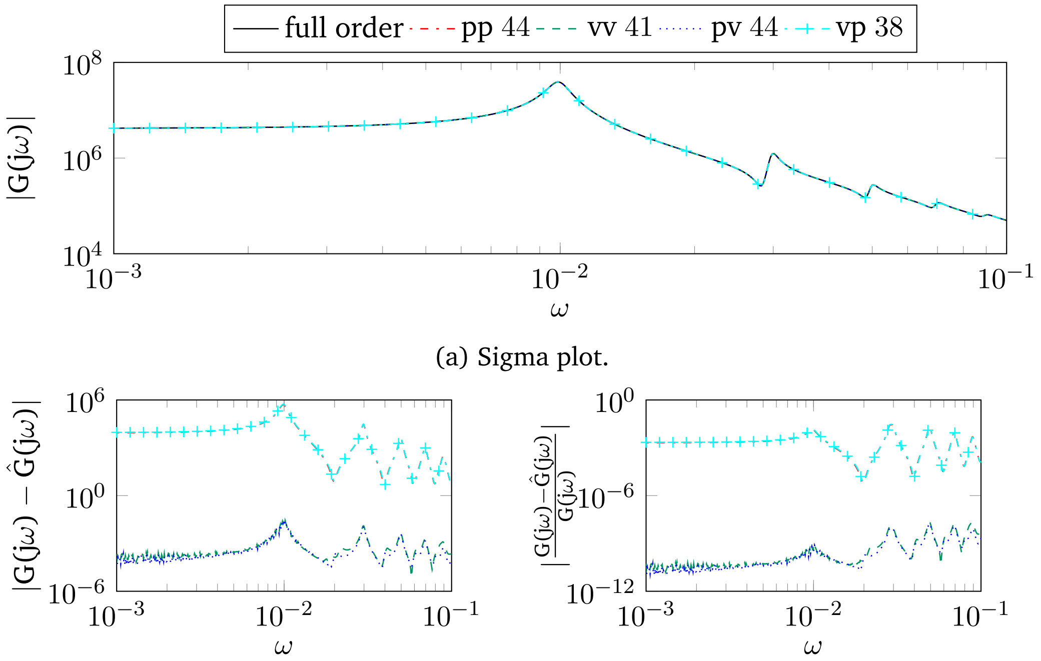
Table 5.4: Comparisons of balancing and IRKA based methods for different dimensional ROMs with TCOM example.



(a) Absolute error.

(b) Relative error.

Figure 5.6: Comparison of balanced truncation and IRKA with a 60 dimensional reduced model of the TCOM example.

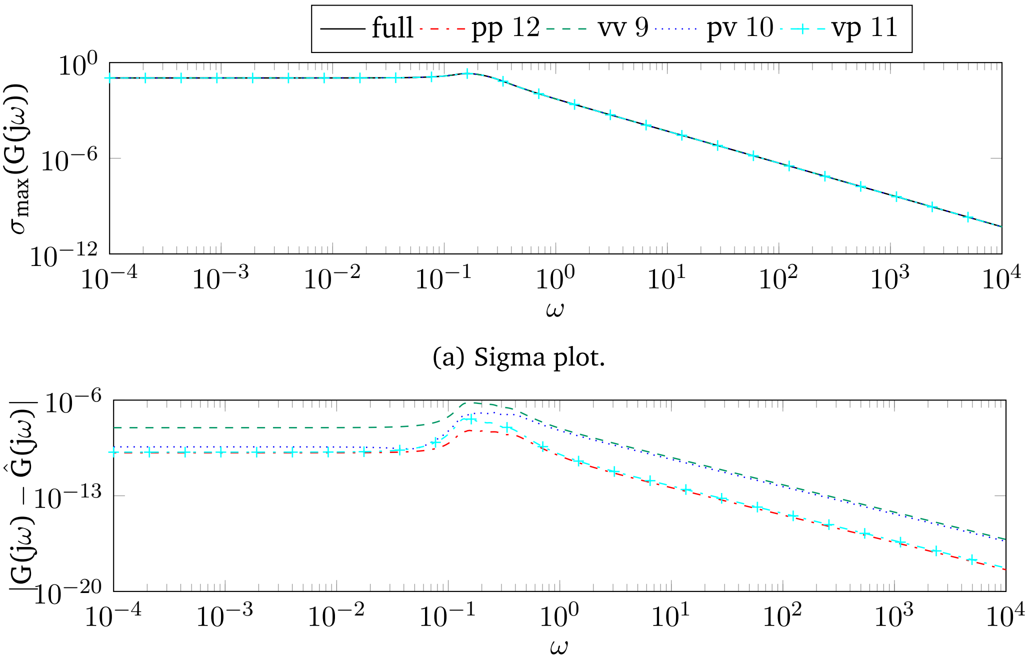


(a) Sigma plot.

(b) Absolute error.

(c) Relative error.

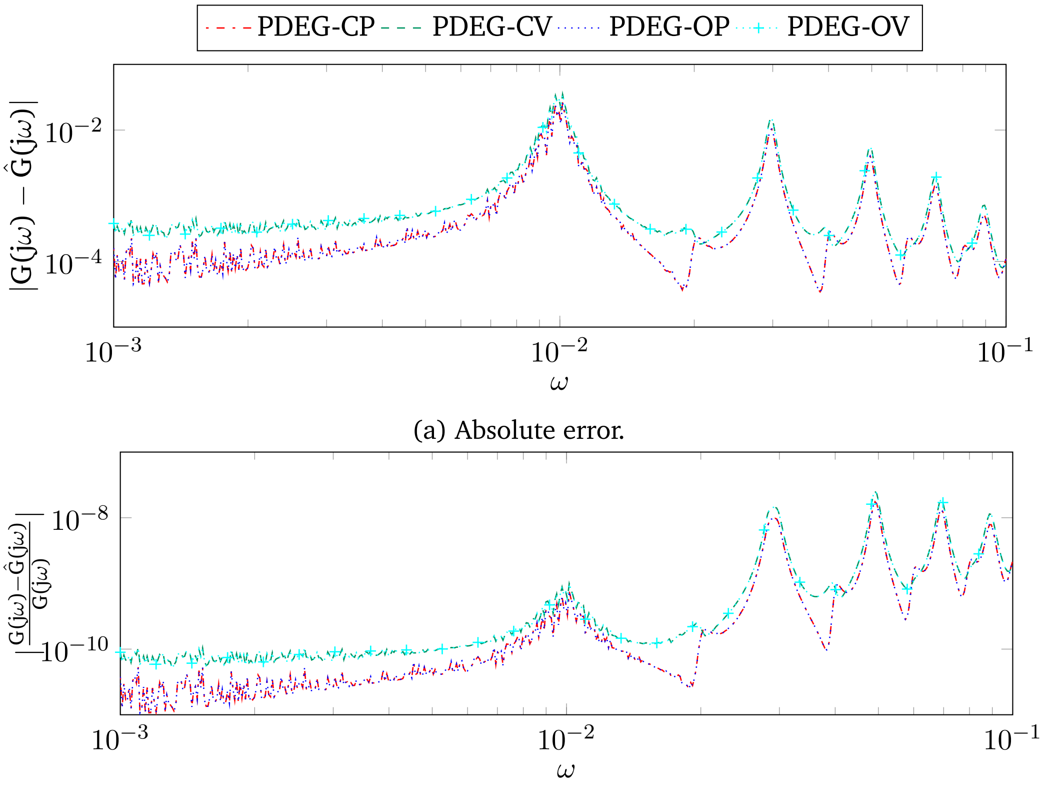
Figure 5.7: Comparison of full and reduced models via balancing on different levels for the TCOM model.



(a) Sigma plot.

(b) Absolute error.

Figure 5.8: Comparison of full and reduced models via balancing on different levels for the DSMS model.



(a) Absolute error.

(b) Relative error.

Figure 5.9: PDEG based 45 dimensional ROMs for the TCOM example.

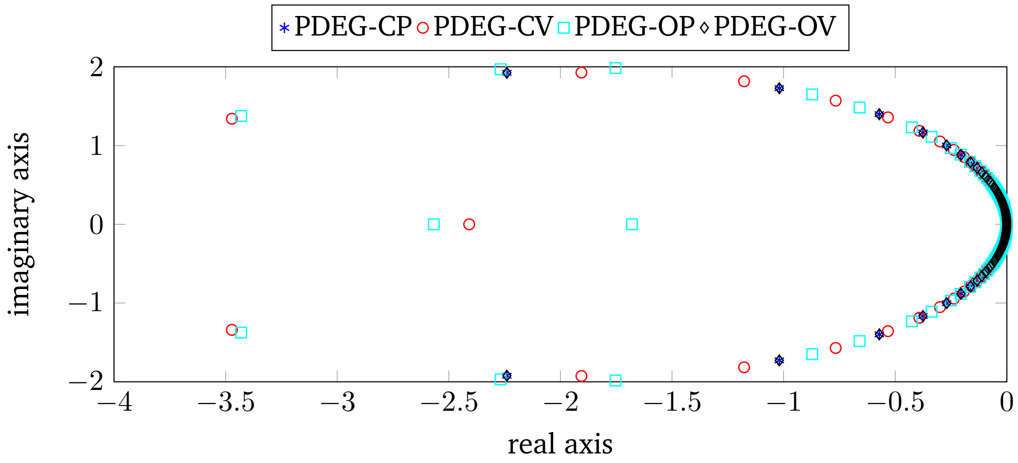


Figure 5.10: Eigenvalues of the ROMs obtained by the PDEG method.

Chapter 6

Conclusion

6.1 Summary

It is clear that descriptor systems, i.e., systems whose dynamics obey algebraic constraints, play a vital role in a wide range of real-life applications. In most cases, the systems are well natured, i.e., structured and sparse. If the model is very large, performing simulation is computationally prohibitively expensive, or is simply impossible due to limited memory. Therefore, reducing the size of the system is unavoidable for fast simulation. The pioneer of model reduction for large-scale descriptor systems is Stykel [111, 112]. However, Stykel discusses a general framework of the BT method for descriptor systems. In principle, one applies the spectral projectors to split the descriptor systems into finite and infinite sub-systems. Then the model reduction is applied to the finite sub-system. Recently, another approach of model reduction has been proposed, see, e.g., [2, 3] for the DAEs. Note we have not considered this method in our work. In this method one must split the system into one differential and several algebraic parts by using the projectors. The number of algebraic parts essentially depends on the number of indices. The important notion is that the ROM preserves the indices of the original model. That means the ROM is in descriptor form with the same index of the original system. In either of the above procedures, one can not avoid explicit computation of the projectors.

This thesis is mainly devoted to model reduction of large-scale sparse descriptor systems avoiding (explicit) computation of the projectors. Such an idea has already been investigated in [53] and [70] for, respectively, (first order) index 1 and index 2 stable DAEs to implement the BT based MOR. The same idea (i.e., the avoidance of projectors) is generalized in [68] for interpolatory model reduction via IRKA of first order structure index 1 and 2 DAEs. In this thesis, we have generalized the idea in [70] for unstable index 2 DAE systems. The major part of this thesis has been dedicated to the MOR of structured second order DAEs arising in different applications. In particular, we considered index 1 and index 3 descriptor systems. In this case, both second-order-to-first order and second-order-to-second-order reduction methods have been discussed. We mainly emphasized on balancing based techniques. The balancing based method has been compared with that of IRKA when second-order-to-first-order reduction was carried out. In case of the second-order-to-second order reduction methods besides BT, we discussed the PDEG method as well. In the following discussion, we call attention to some noteworthy contributions of this thesis.

In Chapter 3 we discussed a balancing based model reduction technique for unstable index 2 descriptor systems arising from flow control problems. Particularly, we considered FEM semi-discretized linearized Navier-Stokes models [12] with a moderate Reynolds number which lead to index 2 DAEs. We showed, by using the idea in [70], that explicit computation of projectors can be avoided for implementing balanced truncation. In the implementation of the BT, the severe complexity arises in solving the two continuous time algebraic Lyapunov equations using the LRCF-ADI iteration for the Bernoulli stabilized system. Bernoulli stabilization essentially makes the system dense and hence causes expensive computation. To avoid this problem we used the Sherman-Morrison-Woodbury formula in solving the linear system inside the LRCF-ADI method. This formula allows to solve the linear systems by exploiting the sparsity of the original model. We also discussed the ADI shift parameter generation (both the heuristic and adaptive) techniques for the underlying system to ensure fast convergence of the LRCF-ADI iteration. Moreover, we showed how to compute an approximate Riccati based boundary feedback stabilization matrix for the full order model from the ROM. The efficiency of our proposed method is discussed using numerical results obtained by applying our algorithms to the linearization of the von Kármán vortex shedding at a moderate Reynolds number. We also demonstrated how the resulting reduced model can be used to accurately simulate the unstable linearized model and to design a stabilizing controller. The balancing based results were also compared with those of IRKA.

Chapter 4 was dedicated to the model reduction of second order index 1 systems arising from multi-physics, mechatronics, constraint mechanics, and so forth. In particular, we investigated MOR of a finite element model of a spindle head configuration in a machine tool. The special feature of this spindle head is that it is partially driven by a set of piezo actuators. Due to this piezo actuation, the resulting model is a second order differential-algebraic system of index 1. We focused on a special first order transformation of the second order form, and found a symmetric system where input and output matrices are transposes of each other. This formulation is important since it helps to reduce computations. We then presented second-order-to-first-order reduction techniques using the BT and IRKA methods. Next, we showed structure preserving MOR techniques using the BT and PDEG methods for the model considered in the chapter. It is understood that to perform the BT and PDEG methods, we require to solve the continuous-time algebraic equations for computing the low-rank Gramian factors. Due to the fact that the special first order transformation leads to a symmetric realization, only one Lyapunov equation was to be solved. The Lyapunov equation that arose was based on the ODE formulation of the DAE system. In this setting, the system matrix was in a Schur complement form, which is typically dense. To avoid this problem, we solved the linear system in each iteration in the LRCF-ADI method by undoing Schur complement [53]. Moreover, for faster computation we converted the large linear system into a smaller one by exploiting the knowledge of the structure of the system. To ensure fast convergence of the LRCF-ADI iteration, we proposed a different technique on an adaptive shift selection approach. Finally we have applied our methods in the real-world, in one large FEM model of a micro-mechanical piezo-actuators based adaptive spindle support (ASS) with almost 300 000 degrees of freedom. Numerical results have been discussed to show the efficiency, accuracy and capability of our proposed methods.

We have discussed model reduction of second order index 3 systems arising in the constrained mechanics or multibody dynamics in Chapter 5. We showed how to convert the second order index 3 DAE system into an equivalent second order projected ODE system. Second-order-to-first-order conversion gave a structured first order index 3 systems as in [87]. Neither balancing nor interpolatory methods of such structured index 3 systems were discussed in [70] or [68]. This gap was closed in this chapter. Then structure preserving MOR techniques were shown using the BT and PDEG methods. In the implementation (for both second-order-to-first-order and second-order-to-second-order reductions), we showed the explicit computation of the projected ODE form of the DAE is not required. To compute the low-rank Gramian factors, we discussed the solution of the projected Lyapunov equation without explicit use of spectral or hidden manifold projectors. In this case, we also discussed how to solve the linear systems efficiently inside the LRCF-ADI by splitting them, exploiting the block structure of the second order DAEs. Further, we discussed an efficient ADI shift parameter computation using heuristic and adaptive approaches for these particularly structured descriptor systems. The efficiency and accuracy of our proposed methods were tested by applying them to several examples with a large number of degrees of freedom.

6.2 Future work

Although there are several questions and challenges that remain open and should be discussed in future research, this work has revealed some new aspects within the area of model order reduction of large-scale linear dynamical systems.

This thesis has concentrated on the model reduction methods for particularly structured DAEs by avoiding (or implicitly handling) the spectral or hidden manifold projectors. This has been possible only by exploiting the knowledge of the structure of the system. The idea that avoidance of spectral or hidden manifold projectors in the model reduction of other classes of descriptor systems may be a fruitful and exciting direction for future research.

Secondly, the idea of balanced truncation for structured index 2 descriptor systems presented in Chapter 3 can be extended to index 1 or index 3 DAEs. We applied the (balancing and truncating transformations) projectors directly to the unstable system. In this case, one cannot guarantee that the error system is stable. Hence, $\mathcal{H}_\infty$ error analysis was infeasible, reflected from the numerical results. Only by partitioning the stable and unstable elements of the system, then applying the MOR method to the stable part, one can obtain a ROM where the unstable dynamics are included. Then the error system guarantees the stability. The open question is whether one can implicitly handle the projectors to partition the system.

In the case of second-order-to-first-order reduction, besides the balanced truncation method, we discussed the interpolatory projection via IRKA. The IRKA based reduction techniques can be extended to the structure preserving model reduction to compare with the balanced truncation or PDEG methods. The PDEG method is easier and computationally a bit less expensive than balanced truncation. In second-order-to-second-order reduction, balanced truncation usually cannot preserve the stability of the original system. Numerically, we showed the PDEG method can preserve the stability of the original systems. In the future this can be investigated from a theoretical perspective. It is also an open question whether this method is applicable for other structured dynamical systems. The method can be useful particularly for the dynamical system having no output equations, because then the projector can be generated from only the low-rank factor of the controllability Gramian.

We presented LRCF-ADI algorithms capable of solving the Lyapunov equations of large-scale sparse systems. In the algorithm solving linear systems at each iteration is expensive. We used direct sparse solvers to solve the linear system. Future research would be conducted to determine if an existing iterative solver can better solve the linear systems. In this case we can exploit the shift-invariance property by modifying the linear system. Further, we encourage exploration to find more efficient means to compute the ADI shift parameters using the adaptive approach.

Bibliography

[1] M. I. Ahmad and P. Benner, *Interpolatory model reduction techniques for linear second-order descriptor systems*, in Control Conference (ECC), 2014 European, IEEE, 2014, pp. 1075–1079. 88

[2] G. Alì, N. Banagaaya, W. Schilders, and C. Tischendorf, *Index-aware model order reduction for linear index-2 DAEs with constant coefficients*, SIAM Journal on Scientific Computing, 35 (2013), pp. A1487–A1510. 2, 109

[3] ——, *Index-aware model order reduction for differential-algebraic equations*, Mathematical and Computer Modelling of Dynamical Systems, 20 (2014), pp. 345–373. 2, 109

[4] L. Amodei and J.-M. Buchot, *A stabilization algorithm of the Navier-Stokes equations based on algebraic Bernoulli equation*, Numer. Lin. Alg. Appl., 19 (2012), pp. 700–727. 35, 46

[5] A. C. Antoulas, *Approximation of Large-Scale Dynamical Systems*, SIAM Publications, Philadelphia, PA, 2005. 2, 12, 17, 18, 19, 35

[6] A. C. Antoulas, C. A. Beattie, and S. Gugercin, *Interpolatory model reduction of large-scale dynamical systems*, in Efficient modeling and control of large-scale systems, J. Mohammadpour and K. M. Grigoriadis, eds., Springer-Verlag, New York, 2010, pp. 3–58. 2

[7] A. C. Antoulas, D. C. Sorensen, and S. Gugercin, *A survey of model reduction methods for large-scale systems*, Contemp. Math., 280 (2001), pp. 193–219. 2

[8] V. I. Arnold, *Mathematical Methods of Classical Mechanics*, Springer-Verlag, New York, 1989. 13, 78

[9] U. M. Ascher and L. R. Petzold, *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*, SIAM, Philadelphia, PA, 1998. 30

[10] Z. Bai and Y. Su, *SOAR: A second-order Arnoldi method for the solution of the quadratic eigenvalue problem*, SIAM J. Matrix Anal. Appl., 26 (2005), pp. 640–659. 82

[11] Z. Bai and Y.-F. Su, *Dimension reduction of large-scale second-order dynamical systems via a second-order Arnoldi method*, SIAM J. Sci. Comput., 26 (2005), pp. 1692–1709. 4, 23, 90

[12] E. Bänsch, P. Benner, J. Saak, and H. Weichelt, *Riccati-based boundary feedback stabilization of incompressible Navier-Stokes flow*, Preprint SPP1253-154, DFG-SPP1253, 2013. 3, 34, 35, 44, 45, 46, 110

[13] S. Barrachina, P. Benner, and E. Quintana-Ortí, *Efficient algorithms for generalized algebraic Bernoulli equations based on the matrix sign function*, Numerical Algorithms, 46 (2007), pp. 351–368. 36

[14] R. Bartels and G. Stewart, *Solution of the matrix equation $AX+XB=C$: Algorithm 432*, Comm. ACM, 15 (1972), pp. 820–826. 19

[15] U. Baur, *Control-Oriented Model Reduction for Parabolic Systems*, PhD thesis, Inst. f. Mathematik, Technische Universität Berlin, Berlin, Jan. 2008. ISBN 978-3639074178 Vdm Verlag Dr. Müller, Available from `http://www.nbn-resolving.de/urn:nbn:de:kobv:83-opus-17608`. 19

[16] C. A. Beattie and S. Gugercin, *Interpolatory projection methods for structure-preserving model reduction*, Sys. Control Lett., 58 (2009), pp. 225–232. 4, 23

[17] T. Bechtold, G. Schrag, and L. Feng, eds., *System-Level Modeling of MEMS*, Advanced Micro & Nanosystems, Wiley-VCH, 2013. 12

[18] P. Benner, M. Hinze, and E. J. W. ter Maten, *Model Reduction for Circuit Simulation*, vol. 74 of Lecture Notes in Electrical Engineering, Springer-Verlag, Heidelberg, Germany, 2011. 2

[19] P. Benner, P. Kürschner, and J. Saak, *Efficient handling of complex shift parameters in the low-rank Cholesky factor ADI method*, Numerical Algorithms, 62 (2013), pp. 225–251. 10.1007/s11075-012-9569-7. 4, 24, 25

[20] P. Benner, P. Kürschner, and J. Saak, *An improved numerical method for balanced truncation for symmetric second order systems*, Math. Comput. Model. Dyn. Syst., 19 (2013), pp. 593–615. 4, 22, 23, 24, 25, 26, 61, 63

[21] P. Benner, P. Kürschner, and J. Saak, *Self-generating and efficient shift parameters in adi methods for large Lyapunov and Sylvester equations*, Electr. Trans. Num. Anal., 43 (2014), pp. 142–162. 4, 5, 28, 29, 42, 43, 68, 98, 99

[22] P. Benner, J.-R. Li, and T. Penzl, *Numerical solution of large Lyapunov equations, Riccati equations, and linear-quadratic control problems*, Numer. Lin. Alg. Appl., 15 (2008), pp. 755–777. 4, 19, 24

[23] P. Benner, V. Mehrmann, and D. Sorensen, *Dimension Reduction of Large-Scale Systems*, vol. 45 of Lecture Notes in Computational Science and Engineering, Springer-Verlag, Berlin/Heidelberg, Germany, 2005. 2, 116, 118, 120, 121

[24] P. Benner, H. Mena, and J. Saak, *On the parameter selection problem in the Newton-ADI iteration for large-scale Riccati equations*, Electr. Trans. Num. Anal., 29 (2008). 28

[25] P. Benner and E. Quintana-Ortí, *Solving stable generalized Lyapunov equations with the matrix sign function*, Numer. Algorithms, 20 (1999), pp. 75–100. 19

[26] P. Benner, E. Quintana-Ortí, and G. Quintana-Ortí, *A portable subroutine library for solving linear control problems on distributed memory computers*, in Workshop on Wide Area Networks and High Performance Computing, Essen (Germany), September 1998, G. Cooperman, E. Jessen, and G. Michler, eds., Lecture Notes in Control and Information, Springer-Verlag, Berlin/Heidelberg, Germany, 1999, pp. 61–88. 19

[27] ——, *Singular perturbation approximation of large, dense linear systems*, in Proc. 2000 IEEE Intl. Symp. CACSD, Anchorage, Alaska, USA, September 25–27, 2000, IEEE Press, Piscataway, NJ, 2000, pp. 255–260. 2

[28] P. Benner, E. S. Quintana-Orti, and G. Quintana-Orti, *Balanced truncation model reduction of large-scale dense systems on parallel computers*, Math. Comput. Model. Dyn. Syst., 6 (2000), pp. 383–405. 19

[29] P. Benner and J. Saak, *Efficient balancing based MOR for large scale second order systems*, Math. Comput. Model. Dyn. Syst., 17 (2011), pp. 123–143. 4, 22, 23, 91

[30] ——, *Numerical solution of large and sparse continuous time algebraic matrix Riccati and Lyapunov equations: a state of the art survey*, GAMM Mitteilungen, 36 (2013), pp. 32–52. 19

[31] P. Benner, J. Saak, and M. M. Uddin, *Second order to second order balancing for index-1 vibrational systems*, in 2012 7th International Conference on Electrical & Computer Engineering (ICECE), IEEE, 2012, pp. 933–936. 62, 65, 68, 69

[32] P. Benner, J. Saak, and M. M. Uddin, *Balancing based model reduction for structured index-2 unstable descriptor systems with application to flow control*, Preprint MPIMD/14-20, Max Planck Institute Magdeburg, Nov. 2014. Available from `http://www.mpi-magdeburg.mpg.de/preprints/`. 34, 49

[33] ——, *Structure preserving MOR for large sparse second order index-1 systems and application to a mechatronic model*, Preprint MPIMD/14-23, Max

Planck Institute Magdeburg, Dec. 2014. Available from http://www.mpi-magdeburg.mpg.de/preprints/. 62, 66

[34] P. Benner and T. Stykel, *Numerical solution of projected algebraic Riccati equations*, SIAM J. Numer. Anal., 52 (2014), pp. 581–600. 40, 45

[35] K. Brenan, S. Campbell, and L. Petzold, *Numerical Solution of Initial–Value Problems in Differential–Algebraic Equations*, Elsevier Science Publishing, North-Holland, 1989. 79

[36] A. Bunse-Gerstner, D. Kubalinska, G. Vossen, and D. Wilczek, *h_2-norm optimal model reduction for large scale discrete dynamical MIMO systems*, J. Comput. Appl. Math., 233 (2010), pp. 1202–1216. 22

[37] Y. Chahlaoui, K. A. Gallivan, A. Vandendorpe, and P. V. Dooren, *Model reduction of second-order systems*. Chapter 6 (pages 149–172) of [23]. 22

[38] Y. Chahlaoui, D. Lemonnier, A. Vandendorpe, and P. V. Dooren, *Second-order balanced truncation*, Linear Algebra Appl., 415 (2006), pp. 373–384. 15

[39] K. A. Cliffe, T. J. Garratt, and A. Spence, *Eigenvalues of block matrices arising from problems in fluid mechanics*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 1310–1318. 43, 98

[40] R. Craig and J. Kurdila, *Fundamentals of structural dynamics*, John Wiley and Sons, New Jersey, 2006. 13

[41] B. Datta, *Linear and numerical linear algebra in control theory: Some research problems*, Linear Algebra Appl., 197/198 (1994), pp. 755–790. 9

[42] ———, *Numerical Methods for Linear Control Systems*, Elsevier Academic Press, 2004. 11, 44

[43] T. A. Davis, *Direct Methods for Sparse Linear Systems*, SIAM Publications, Philadelphia, PA, 2006. 60, 66

[44] J. G. de Jalón and B. Eduardo, *Kinematic and Dynamic Simulation of Multibody Systems*, Springer-Verlag, 1984. 78

[45] W. G. Drossel and V. Wittstock, *Adaptive spindle support for improving machining operations*, CIRP Annals - Manufacturing Technology, 57 (2008), pp. 395–398. 55

[46] V. Druskin and L. Knizhnerman, *Extended Krylov subspaces: Approximation of the matrix square root and related functions*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 755–771. DOI: 10.1137/S0895479895292400. 4, 19

[47] I. Duff, A. Erisman, and J. Reid, *Direct methods for sparse matrices*, Clarendon Press, Oxford, UK, 1989. 60

[48] E. EICH-SOELLNER AND C. FÜHRER, *Numerical Methods in Multibody Dynamics*, European Consortium for Mathematics in Industry, B. G. Teubner GmbH, Stuttgart, 1998. 13, 37, 53, 77, 78, 79, 98

[49] I. M. ELFADEL AND D. D. LING, *A block rational Arnoldi algorithm for multipoint passive model-order reduction of multiport RLC networks*, in Proceedings of the 1997 IEEE/ACM international conference on Computer-aided design, IEEE Computer Society, 1997, pp. 66–71. 2

[50] D. F. ENNS, *Model reduction with balanced realizations: An error bound and a frequency weighted generalization*, The 23rd IEEE Conference on Decision and Control, 23 (1984), pp. 127–132. 2

[51] P. FELDMANN AND R. W. FREUND, *Efficient linear circuit analysis by Padé approximation via the Lanczos process*, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., 14 (1995), pp. 639–649. 2, 20

[52] K. V. FERNANDO AND H. NICHOLSON, *Singular perturbational model reduction of balanced systems*, IEEE Trans. Automat. Control, (1982), pp. 466–468. 2

[53] F. FREITAS, J. ROMMES, AND N. MARTINS, *Gramian-based reduction method applied to large sparse power system descriptor models*, IEEE Trans. Power Systems, 23 (2008), pp. 1258–1270. 3, 29, 31, 58, 109, 111

[54] R. W. FREUND, *Krylov-subspace methods for reduced-order modeling in circuit simulation*, J. Comput. Appl. Math., 123 (2000), pp. 395–421. 2

[55] Z. GAJIĆ AND M. QURESHI, *Lyapunov Matrix Equation in System Stability and Control*, Math. in Science and Engineering, Academic Press, San Diego, CA, 1995. 10

[56] K. GALLIVAN, E. GRIMME, AND P. VAN DOOREN, *A rational Lanczos algorithm for model reduction*, Numerical Algorithms, 12 (1996), pp. 33–63. 2

[57] K. GALLIVAN, A. VANDENDORPE, AND P. VAN DOOREN, *Model reduction of MIMO systems via tangential interpolation*, SIAM J. Matrix Anal. Appl., 26 (2004), pp. 328–349. 21

[58] M. GERDIN, *Identification and estimation for models described by differential-algebraic equations*, PhD thesis, Linköpings Universitet, Linköpings, Sweden, 2006. 29

[59] K. GLOVER, *All optimal Hankel-norm approximations of linear multivariable systems and their L^∞ norms*, Internat. J. Control, 39 (1984), pp. 1115–1193. 2, 11, 12, 19

[60] G. GOLUB AND C. VAN LOAN, *Matrix Computations*, Johns Hopkins University Press, Baltimore, third ed., 1996. 8, 28, 41

[61] M. Green and D. Limebeer, *Linear Robust Control*, Prentice-Hall, Englewood Cliffs, NJ, 1995. 9

[62] E. J. Grimme, *Krylov Projection Methods for Model Reduction*, PhD thesis, Univ. of Illinois at Urbana-Champaign, USA, 1997. 2, 20

[63] S. Gugercin and A. Antoulas, *A survey of model reduction by balanced truncation and some new results*, Internat. J. Control, 77 (2004), pp. 748–766. 2

[64] S. Gugercin, A. C. Antoulas, and C. Beattie, *$\mathcal{H}_2$ model reduction for large-scale dynamical systems*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 609–638. 2, 20, 21, 22

[65] S. Gugercin and J.-R. Li, *Smith-type methods for balanced truncation of large systems*. Chapter 2 (pages 49–82) of [23]. 65

[66] S. Gugercin, R. V. Polyuga, C. Beattie, and A. Van Der Schaft, *Structure-preserving tangential interpolation for model reduction of port-Hamiltonian systems*, Automatica, 48 (2012), pp. 1963–1974. 4, 23

[67] S. Gugercin, D. Sorensen, and A. Antoulas, *A modified low-rank Smith method for large-scale Lyapunov equations*, Numer. Algorithms, 32 (2003), pp. 27–55. 19

[68] S. Gugercin, T. Stykel, and S. Wyatt, *Model reduction of descriptor systems by interpolatory projection methods*, SIAM J. Sci. Comput., 35 (2013), pp. 1010–1033. 3, 4, 29, 31, 33, 51, 77, 85, 88, 89, 109, 111

[69] S. Hammarling, *Numerical solution of the stable, non-negative definite Lyapunov equation*, IMA J. Numer. Anal., 2 (1982), pp. 303–323. 19

[70] M. Heinkenschloss, D. C. Sorensen, and K. Sun, *Balanced truncation model reduction for a class of descriptor systems with applications to the Oseen equations*, SIAM J. Sci. Comput., 30 (2008), pp. 1038–1063. 3, 4, 29, 31, 33, 37, 39, 40, 41, 42, 77, 85, 88, 93, 109, 110, 111

[71] P. Hood and C. Taylor, *Navier-Stokes equations using mixed interpolation*, in Finite Element Methods in Flow Problems, J. T. Oden, R. H. Gallagher, C. Taylor, and O. C. Zienkiewicz, eds., University of Alabama in Huntsville Press, 1974, pp. 121–132. 34

[72] I. Jaimoukha and E. Kasenally, *Krylov subspace methods for solving large Lyapunov equations*, SIAM J. Numer. Anal., 31 (1994), pp. 227–251. 19

[73] K. Jbilou and A. J. Riquet, *Projection methods for large Lyapunov matrix equations*, Linear Algebra Appl., 415 (2006), pp. 344–358. 19

[74] B. Kranz, *Zustandsraumbeschreibung von piezo-mechanischen Systemen auf Grundlage einer Finite-Elemente-Diskretisierung*, in ANSYS Conference & 27th CADFEM users' meeting 2009, Congress Center Leipzig, Germany, November (18-20) 2009. 69

[75] P. Kunkel and V. Mehrmann, *Differential-Algebraic Equations: Analysis and Numerical Solution*, Textbooks in Mathematics, EMS Publishing House, 2006. 29, 79

[76] S. Lall, P. Krysl, and J. E. Marsden, *Structure-preserving model reduction for mechanical systems*, Phys. D, 184 (2003), pp. 304–318. Complexity and nonlinearity in physical systems (Tucson, AZ, 2001). 4, 23

[77] A. J. Laub, M. T. Heath, C. C. Paige, and R. C. Ward, *Computation of system balancing transformations and other applications of simultaneous diagonalization algorithms*, IEEE Trans. Automat. Control, 32 (1987), pp. 115–122. 12, 65

[78] J. R. Leigh, *Functional Analysis and Linear Control Theory*, Academic Press, New York, 1980. 9, 10

[79] J.-R. Li, *Model Reduction of Large Linear Systems via Low Rank System Gramians*, PhD Thesis, Massachusettes Institute of Technology, 2000. 24, 28, 54, 64

[80] J.-R. Li and J. White, *Efficient model reduction of interconnect via approximate system gramians*, in Proceedings of the IEEE/ACM international conference on Computer-aided design, IEEE Press, 1999, pp. 380–384. 2, 4, 54, 64

[81] ——, *Low rank solution of Lyapunov equations*, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 260–280. 4, 19, 24

[82] F. Lin, *Robust Control Design: An Optimal Control Approach*, AFI Press, 1997. 9

[83] Y. Liu and B. D. Anderson, *Singular perturbation approximation of balanced systems*, International Journal of Control, 50 (1989), pp. 1379–1405. 2

[84] A. Lu and E. Wachspress, *Solution of Lyapunov equations by alternating direction implicit iteration.*, Comput. Math. Appl., 21 (1991), pp. 43–58. 24

[85] M. Marcus and H. Minc, *A Survey of Matrix Theory and Matrix Inequalities*, Allyn and Bacon, Boston, 1964. 69

[86] R. März, *Canonical projectors for linear differential algebraic equations*, Computers Math. Applic., 31 (1996), pp. 121–135. 30

[87] V. MEHRMANN AND T. STYKEL, *Balanced truncation model reduction for large-scale systems in descriptor form*. Chapter 3 (pages 83–115) of [23]. 3, 77, 84, 99, 100, 111

[88] D. G. MEYER AND S. SRINIVASAN, *Balancing and model reduction for second-order form linear systems.*, IEEE Trans. Automat. Control, 41 (1996), pp. 1632–1644. 15, 23, 61

[89] B. C. MOORE, *Principal component analysis in linear systems: controllability, observability, and model reduction*, IEEE Trans. Automat. Control, AC-26 (1981), pp. 17–32. 2

[90] R. NEUGEBAUER, W. G. DROSSEL, A. BUCHT, B. KRANZ, AND K. PAGEL, *Control design and experimental validation of an adaptive spindle support for enhanced cutting processes*, CIRP Annals - Manufacturing Technology, 59 (2010), pp. 373–376. 55

[91] R. NEUGEBAUER, W.-G. DROSSEL, K. PAGEL, AND B. KRANZ, *Making of state space models of piezo-mechanical systems with exact impedance mapping and strain output signals*, in Mechatronics and Material Technologies, vol. 2, Zurich, Switzerlan, June 28-30 2010, Swiss Federal Institute of Technology ETH, pp. 73–80. 57

[92] T. PENZL, *Numerische Lösung großer Lyapunov-Gleichungen*, Logos–Verlag, Berlin, Germany, 1998. Dissertation, Fakultät für Mathematik, TU Chemnitz, 1998. 69

[93] ——, *A cyclic low rank Smith method for large sparse Lyapunov equations*, SIAM J. Sci. Comput., 21 (2000), pp. 1401–1418. 4, 19, 24, 28, 42, 68, 98

[94] ——, *Algorithms for model reduction of large dynamical systems*, Linear Algebra Appl., 415 (2006), pp. 322–343. (Reprint of Technical Report SFB393/99-40, TU Chemnitz, 1999.). 2, 4, 54, 64

[95] Z.-Q. QU, *Model Order Reduction Techniques: with Applications in Finite Element Analysis*, Springer-Verlag, Berlin, 2004. 2

[96] T. REIS AND T. STYKEL., *Balanced truncation model reduction of second-order systems*, Math. Comput. Model. Dyn. Syst., 14 (2008), pp. 391–406. 4, 23, 61

[97] T. REIS AND T. STYKEL, *Positive real and bounded real balancing for model reduction of descriptor systems*, Preprint 25-2008, Inst. f. Mathematik, TU Berlin, 2008. 90

[98] A. RUHE, *Rational Krylov algorithms for nonsymmetric eigenvalue problems II: Matrix pairs*, Linear Algebra Appl., 197 (1994), pp. 283–295. 2, 20

[99] Y. SAAD, *Numerical solution of large Lyapunov equation*, in Signal Processing, Scattering, Operator Theory and Numerical Methods, M. A. Kaashoek, J. H. van Schuppen, and A. C. M. Ran, eds., Birkhäuser, 1990, pp. 503–511. 19

[100] Y. SAAD, *Numerical Methods for Large Eigenvalue Problems*, Manchester University Press, Manchester, UK, 1992. 43, 82

[101] Y. SAAD, *Iterative Methods for Sparse Linear Systems*, SIAM, Philadelphia, PA, USA, 2003. 60, 66

[102] J. SAAK, *Efficient Numerical Solution of Large Scale Algebraic Matrix Equations in PDE Control and Model Order Reduction*, PhD thesis, TU Chemnitz, July 2009. available from `http://nbn-resolving.de/urn:nbn:de:bsz:ch1-200901642`. 25, 28, 100

[103] M. G. SAFONOV AND R. Y. CHIANG, *A Schur method for balanced-truncation model reduction*, IEEE Trans. Automat. Control, 34 (1989), pp. 729–733. 2

[104] B. SALIMBAHRAMI, *Structure Preserving Order Reduction of Large Scale Second Order Models*, PhD Thesis, Technische Universität München, 2005. 4, 23, 90

[105] B. SALIMBAHRAMI AND B. LOHMANN, *Order reduction of large scale second-order systems using Krylov subspace methods*, Linear Algebra Appl., 415 (2006), pp. 385–405. 4, 23

[106] W. H. A. SCHILDERS, H. A. VAN DER VORST, AND J. ROMMES, *Model Order Reduction: Theory, Research Aspects and Applications*, Springer-Verlag, Berlin, Heidelberg, 2008. 2

[107] B. SIMEON, C. FÜHRER, AND P. RENTROP, *The Drazin inverse in multibody system dynamics*, Numer. Math., 64 (1993), pp. 521 – 539. 77, 98

[108] V. SIMONCINI, *A new iterative method for solving large-scale Lyapunov matrix equations*, SIAM J. Sci. Comput., 29 (2007), pp. 1268–1288. 4, 19, 24

[109] E. SONTAG, *Mathematical Control Theory*, Springer-Verlag, New York, NY, 2nd ed., 1998. 9, 10, 34, 44

[110] D. C. SORENSEN AND A. C. ANTOULAS, *On model reduction of structured systems*. Chapter 4 (pages 117–130) of [23]. 22

[111] T. STYKEL, *Model reduction of descriptor systems*, Tech. Rep. 720-2001, Institut für Mathematik, TU Berlin, D-10263 Berlin, Germany, 2001. 3, 30, 109

[112] ——, *Analysis and Numerical Solution of Generalized Lyapunov Equations*, Dissertation, TU Berlin, 2002. 3, 24, 29, 30, 31, 109

[113] T. STYKEL, *Gramian-based model reduction for descriptor systems*, Math. Control Signals Systems, 16 (2004), pp. 297–319. 3

[114] T.-J. Su and R. R. Craig jr., *Model reduction and control of flexible structures using Krylov vectors*, Journal of Guidance, Control, and Dynamics, 14 (1991), pp. 260–267. 90

[115] F. Tisseur, *Backward error and condition of polynomial eigenvalue problems*, Linear Algebra Appl., 309 (2000), pp. 339–361. 82

[116] F. Tisseur and K. Meerbergen, *The quadratic eigenvalue problem*, SIAM Review, 43 (2001), pp. 235–286. 13, 82, 83

[117] M. S. Tombs and I. Postlethwaite, *Truncated balanced realization of a stable nonminimal state-space system*, Internat. J. Control, 46 (1987), pp. 1319–1330. 2, 12, 18

[118] N. Truhar and K. Veselić, *Bounds on the trace of a solution to the Lyapunov equation with a general stable matrix*, Sys. Control Lett., 56 (2007), pp. 493–503. 100

[119] ——, *An efficient method for estimating the optimal dampers' viscosity for linear vibrating systems using Lyapunov equation*, SIAM J. Matrix Anal. Appl., 31 (2009), pp. 18–39. 100

[120] M. M. Uddin, *Model Reduction for Piezo-Mechanical System using Balanced Trancation*, Master's thesis, Stockholms University, Stockholm, Sweden, 2011. Available from `http://www.qucosa.de/fileadmin/data/qucosa/documents/7822/Master_Thesis_Uddin.pdf`. 54, 68

[121] M. M. Uddin, J. Saak, B. Kranz, and P. Benner, *Computation of a compact state space model for an adaptive spindle head configuration with piezo actuators using balanced truncation*, Production Engineering, 6 (2012), pp. 577–586. 53, 65

[122] ——, *Efficient reduced order state space model computation for a class of second order index one systems*, Proc. Appl. Math. Mech., 12 (2012), pp. 699–700. 65

[123] H. A. van der Vorst, *Iterative Krylov Methods for Large Linear Systems*, Cambridge University Press, Cambridge, 2003. 60

[124] D. C. Villemagne and R. E. Skelton, *Model reduction using a projection formulation*, Internat. J. Control, 46 (1987), pp. 2141–2169. 2, 20, 21

[125] E. L. Wachspress, *The ADI minimax problem for complex spectra*, Appl. Math. Letters, 1 (1988), pp. 311–314. 28

[126] ——, *Iterative solution of the Lyapunov matrix equation*, Appl. Math. Letters, 107 (1988), pp. 87–90. 19, 27

[127] ——, *The ADI Model Problem*, Springer New York, 2013. 28

[128] D. Wang and M. N. Harris, *Stability analysis of the equilibrium of a constrained mechanical system*, Internat. J. Control, 60 (1994), pp. 733–746. 63, 65, 77, 78

[129] D. S. Weile, E. Michielssen, E. Grimme, and K. Gallivan, *A method for generating rational interpolant reduced order models of two-parameter linear systems*, Appl. Math. Lett., 12 (1999), pp. 93–102. 2

[130] T. Wolf, H. K. F. Panzer, and B. Lohmann, *ADI iteration for Lyapunov equations: a tangential approach and adaptive shift selection*, e-print 1312.1142, arXiv, Dec. 2013. 42

[131] B. Yan, S. X.-D. Tan, and B. McGaughy, *Second-order balanced truncation for passive-order reduction of RLCK circuits*, IEEE Trans. Circuits Syst. II, 55 (2008), pp. 942–946. 13

[132] F. Zhang, *The Schur Complement and Its Applications*, Springer, New York, 2006. 60

[133] K. Zhou, *Essentials of Robust Control*, Prentice Hall, Upper Saddle River, NJ, 1997. 9

[134] K. Zhou, J. Doyle, and K. Glover, *Robust and Optimal Control*, Prentice-Hall, Upper Saddle River, NJ, 1996. 2

[135] K. Zhou, G. Salomon, and E. Wu, *Balanced realization and model reduction for unstable systems*, Internat. J. Robust and Nonlinear Cont., 9 (1999), pp. 183–198. 3, 33, 35, 36