

**COMPUTATIONAL TECHNIQUES FOR
RICCATI-BASED FEEDBACK STABILIZATION OF
LARGE-SCALE SPARSE INDEX-2 DESCRIPTOR
SYSTEM**

The thesis submitted to the
Department of Mathematics, BUET, DHAKA-1000
in partial fulfillment of the requirements for the degree of

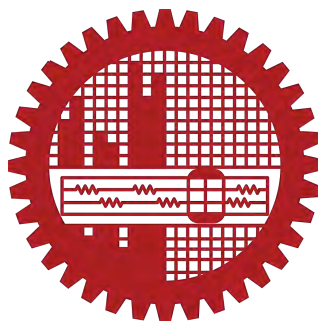
**MASTER OF SCIENCE
IN
MATHEMATICS**

By

MD.TORIQUL ISLAM

Student ID: 0419092505F

Registration No. 0419092505F, Session: April- 2019



Under the supervision

of

Dr. Md. Abdul Hakim Khan

Professor

Department of Mathematics

Bangladesh University of Engineering and Technology (BUET)

Dhaka-1000, Bangladesh

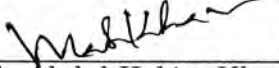
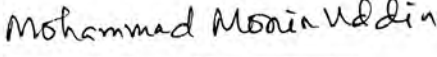


March 6, 2022


The thesis entitled
**COMPUTATIONAL TECHNIQUES FOR
RICCATI-BASED FEEDBACK STABILIZATION OF
LARGE-SCALE SPARSE INDEX-2 DESCRIPTOR
SYSTEM**

Submitted by
MD.TORIQUUL ISLAM

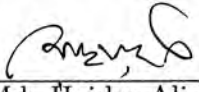
Student ID:0419092505F, Registration No.0419092505F, Session: April- 2019
has been accepted as satisfactory in partial fulfillment for the degree of
Master of Science in Mathematics on 06.03.2022

BOARD OF EXAMINERS

1. 
Dr. Md. Abdul Hakim Khan **Chairman (Supervisor)**
Professor
Department of Mathematics,
BUET, Dhaka-1000
2. 
Dr. Mohammad Monir Uddin **Member (Co-Supervisor)**
Associate Professor
Department of Mathematics and Physics
North South University, Dhaka-1229
3. 
(Dr. Khandker Farid Uddin Ahmed) **Member (Ex-Officio)**
Professor and Head
Department of Mathematics,
BUET, Dhaka-1000
4. 
Dr. Md. Abdul Alim **Member**
Professor
Department of Mathematics,
BUET, Dhaka-1000

5. 
Dr. Mohammed Forhad Uddin
Professor
Department of Mathematics,
BUET, Dhaka-1000

Member

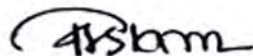
6. 
Dr. Md. Haider Ali Biswas
Professor
Mathematics Discipline,
Khulna University, Khulna-9208.

Member (External)

Declaration of Authorship

I, Md. Toriqul Islam declare that this thesis titled, COMPUTATIONAL TECHNIQUES FOR RICCATI-BASED FEEDBACK STABILIZATION OF LARGE-SCALE SPARSE INDEX-2 DESCRIPTOR SYSTEM and the work presented in this thesis is the outcome of the investigation carried out by the author under the supervision of Dr. Md. Abdul Hakim Khan, Professor, Department of Mathematics, Bangladesh University of Engineering and Technology (BUET), Dhaka-1000 under the requirement of the University's Regulations and Code of Practice for Research Degree Programs and that it has not been submitted anywhere for any other academic award of any degree or diploma.

Signed:



Date:

DEDICATION

This thesis is dedicated

To

My beloved parents

For their endless love, support, and encouragement

Abstract

This thesis mainly focuses on computational techniques applied to stabilize unstable Navier-Stokes models. The models arising from the Navier-Stokes equation are an essential aspect in engineering applications and applied mathematics in fluid mechanics, which significantly depends on Reynolds number (Re), and if $Re \geq 300$, the corresponding model will be unstable. The computation steps are designed to approximate the full models with the ROMs, find the reduced-order feedback matrices, and attain the optimal feedback matrices for stabilizing the desired Navier-Stokes models. The prime concern is exploring the Riccati-based boundary feedback stabilization of incompressible Navier-Stokes flow via Krylov subspace techniques. Since the volume of data derived from the original models is large, the feedback stabilization process through the Riccati equation is always infeasible. Therefore, a \mathcal{H}_2 optimal model-order reduction scheme for reduced-order modeling, preserving the sparsity of the system, is required. Some conventional methods exist, but they have some adversities, such as the requirement of high computation time and memory allocation, complex matrix algebra, and uncertainty of the stability of the reduced-order models. To overcome these drawbacks, an extended form of Krylov subspace-based Two-Sided Iterative Algorithm (TSIA) is implemented to stabilize non-symmetric index-2 descriptor systems explored from unstable Navier-Stokes models. The proposed techniques are sparsity-preserving and utilize the Wilson condition to efficiently satisfy the reduced-order modeling approach through the sparse-dense Sylvester equations. To solve the desired Sylvester equations, sparsity-preserving Krylov subspaces are structured via the system of linear equations with a compact form of matrix-vector operations. Inverse projections approaches are applied to get the optimal feedback matrix from reduced-order models. To validate the efficiency of the proposed techniques, transient behaviors of the target systems are observed, incorporating the tabular and figurative comparisons with MATLAB simulations. Finally, to reveal the advancement of the proposed techniques, we compare our work with some existing results. From the tabular and graphical comparisons of the results of numerical computations, it is observed that RKSM is not applicable for the target models due to the non-symmetric structure. In contrast, TSIA can be suitably applied to solve Sparse-dense Sylvester equations for reduced-order modeling. Furthermore, by the TSIA, full models can be efficiently approximated by the corresponding

ROMs with minimized \mathcal{H}_2 error norm, and the inverse projection scheme is effective in computing the optimal feedback matrices from the reduced-order feedback matrices to stabilize the target models more efficiently than existing methods. Thus, it can be concluded that by utilizing TSIA, unstable Navier-Stokes models can be stabilized with better accuracy and less computing time.

Acknowledgements

All praises for Almighty ALLAH, He is the One who has the power to fulfill the wishes of all, and without his help, nothing would be a success.

I want to express my cordial gratitude and profound honor to my thesis supervisor Professor Dr. Md. Abdul Hakim Khan, Department of Mathematics, Bangladesh University of Engineering and Technology (BUET). For his continuous support, motivation, and guidance throughout my research journey. I am thankful to him and highly fortunate to have a Thesis under his supervision. I am very grateful to him for introducing me to this fascinating and applicable research area and finishing this Thesis successfully. I will remember the memories of working with him for the rest of my life. I am gratefully indebted to him.

In particular, My Co-supervisor, Dr. Mohammad Monir Uddin, Associate professor, Department of Mathematics and Physics, North South University. Deserves my gratitude and indebtedness for his outstanding support, collaboration, and invaluable direction during my research collaboration. He shared his knowledge of subject matter analysis with me while also appreciating my approach to synthesizing such themes. His insightful advice encouraged me to think in new ways, his critiques strengthened my problem-solving skills, and his encouragement gave me strength during a difficult period. I consider the information I have gained from him to be a valuable asset in my life.

I express my gratitude to all my teachers from the Department of Mathematics, Bangladesh University of Engineering and Technology. Furthermore, I praise my Departmental head, Professor Dr. Khandker Farid Uddin Ahmed, for allowing me to use The Departmental computational lab and other facilities during my research period. Moreover, I am grateful to all of the staff of the Department of Mathematics for their supportive attitudes and friendly behavior toward me during any necessity of mine.

I want to show my hearty gratitude to the Bangladesh Bureau of Educational Information and Statistics (BANBEIS), Dhaka, Bangladesh, for their financial support under the project ID: MS0419092505 throughout this research work.

I am grateful to Professor Dr. Md Abdul Alim and Professor Dr. Mohammed Forhad Uddin, Department of Mathematics, BUET, for being on my defense

committee, reading my Thesis, and suggesting improvements. Their treasured supports were influential in sharpening my experiment methods and making me thoughtful to understand mathematics deeply.

Also, I am deeply indebted to Mahtab Uddin, Assistant Professor in Mathematics at United International University, for their enthusiastic inspiration throughout the investigation.

Finally, I would like to express my gratitude to my parents and siblings for their wise counsel and sympathetic ear and my friends' unwavering support and motivation. Their genuine feelings for me were crucial in the research.

Contents

Declaration of Authorship	iii
Abstract	v
Acknowledgements	vii
Contents	ix
List of Figures	xii
List of Tables	xiii
List of Algorithms	xiv
Notations and Symbols	xv
List of Acronyms	xvi
1 Introduction	1
1.1 Motivation	1
1.2 Literature Review	2
1.3 Objective	3
1.4 Outlines of the Thesis	3
2 Preliminaries	5
2.1 Basic Concepts	5
2.1.1 State-Space Representations of Control Systems	5
2.1.2 Standard and Generalized System	8
2.1.3 Descriptor System	8
2.1.4 Input-Output Relations	10
2.1.5 Transfer Function	11
2.1.6 System Rank	11
2.1.7 Reduced-Order Model	12
2.2 Matrix Equations	13
2.2.1 Riccati Equation	13
2.2.2 Lyapunov Equation	14

2.2.3	Sylvester Equation	15
2.2.4	Solution of Sylvester Equation	15
2.3	Stability and Related Topics	17
2.3.1	Stable and Unstable System	17
2.3.2	Feedback Stabilization	19
2.3.3	Riccati Stabilization	20
2.4	Background of Linear Algebra	20
2.4.1	Formation of the Matrices	21
2.4.2	Sparse and Dense Matrix	21
2.4.3	Applications of Sparse and Dense Matrices	21
2.4.4	Matrix pencil	22
2.4.5	Eigenvalue Problem	22
2.4.6	Matrix Definiteness	23
2.4.7	Hessenberg matrix	24
2.4.8	Projection Matrix	25
2.4.9	Matrix Decomposition Techniques	25
2.4.10	Singular-Value Decomposition	25
2.4.11	Eigenvalue Decomposition	26
2.4.12	Schur Decomposition	27
2.4.13	QR Decomposition	27
2.4.14	Cholesky Decomposition	27
2.4.15	Arnoldi Decomposition	28
2.5	Existing Methods	29
2.5.1	Schur Decomposition Method	30
2.5.2	Iterative Rational Krylov Algorithm	31
2.5.3	Alternative Direction Implicit Method	33
2.5.4	Kleinman-Newton Method	36
2.5.5	Rational Krylov Subspace Method	37
2.6	System norms	38
2.6.1	Matrix Norms	38
2.6.2	Vector Norms	39
2.6.3	H_2 -Norm	40
2.6.4	H_∞ -Norm	40
2.7	Error System	41
2.8	Shift Parameters	43
2.8.1	Adaptive ADI Shifts	43
2.9	Existing data models	44
2.9.1	Finite element method	44
2.9.2	Navier–Stokes Model	44
2.9.3	Stokes Model	46
2.9.4	Oseen Model	47
3	TSIA for index-2 descriptor system	50
3.1	Krylov subspace for index-2 descriptor system	50

3.1.1	Structure of the incompressible Navier-Stokes model	51
3.1.2	Conversion of index-2 descriptor system to generalized system	53
3.1.3	Sparsity-preserving Krylov subspace bases for IRKA	54
3.2	Two Sided Iterative Algorithm for index-2 descriptor systems	55
3.2.1	Formulation of the generalized sparse-dense Sylvester equation	56
3.2.2	Solving generalized sparse-dense Sylvester equation	57
3.2.3	Two Sided Iterative Algorithm to estimate the optimal feed-back matrix	57
3.2.4	Stabilization of index-2 descriptor system	59
3.2.5	\mathcal{H}_2 - norm of the error system	59
4	Numerical result	61
4.1	Model description	61
4.2	Approximation of the full models with the reduced-order models . .	62
4.2.1	Comparison of the transfer functions	62
4.2.2	\mathcal{H}_2 -norm of the error system for the ROMs	64
4.3	Graphical Comparisons of Stabilization of the Unstable Systems . .	64
4.3.1	Stabilization of the eigenvalues	64
4.3.2	Stabilization of the step-responses	65
4.4	Comparison of the TSIA with IRKA	65
5	Conclusion and future research	73
5.1	Summary	73
5.2	Limitations	74
5.3	Future Research	75
	References	76

List of Figures

2.1	State-space system	8
2.2	Technique of reduced-order modeling	13
2.3	Feedback approach in a system	19
2.4	Initial discretization of the von Kármán vortex street with coordinates, boundary parts and observation points	45
2.5	Mesh structure of a pipe flow	48
2.6	Velocity shape of a pipe flow	49
2.7	Pressure shape of a pipe flow	49
4.1	Comparison of full model and ROM of 3-dimensional model for $Re = 500$	63
4.2	Unstable eigenvalues of 3-dimensional models	66
4.3	Stabilized eigenvalues of 3-dimensional models	67
4.4	Unstable step response for 1st input and 1st output of 3-dimensional models	68
4.5	Stabilized step response for 1st input and 1st output of 3-dimensional models	69
4.6	Unstable step response for 2nd input and 7th output of 3-dimensional models	70
4.7	Stabilized step response for 2nd input and 7th output of 3-dimensional models	71

List of Tables

2.1	Dimension of Oseen model	48
4.1	Structure of the target Navier-Stokes models	62
4.2	\mathcal{H}_2 error norm of the ROMs of the target models	64
4.3	Comparison of computation time and \mathcal{H}_2 error norm of the ROMs of 3-dimensional models archived by TSIA and IRKA	65

List of Algorithms

1	Solution of the Sparse-Dense Generalized Sylvester Equation.	17
2	Arnoldi decomposition (Modified Gram-Schmidt).	29
3	Schur decomposition method.	31
4	IRKA for generalized systems.	33
5	LRCF-ADI for generalized systems.	36
6	Sparsity-preserving IRKA for index-2 descriptor systems.	56
7	Solution of generalized sparse-dense Sylvester equation.	58
8	TSIA for the optimal feedback matrix of index-2 descriptor systems.	58

Notations and Symbols

\mathbb{R}	field of real numbers
\mathbb{C}	field of complex numbers
\mathbb{C}^-	left complex half-plane
\mathbb{C}^+	right complex 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 complex matrices of order $m \times n$
\subset	subset of any matrix
\in	belongs to
a_{ij}	the i, j -th entry of the matrix A
$\operatorname{Re}(z)$	real part of $z \in \mathbb{C}$
$\operatorname{Im}(z)$	imaginary part of $z \in \mathbb{C}$
I_n	$n \times n$ identity matrix of order n
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)
$G(s)$	transfer function or transfer function matrix
$G(jw)$	frequency response
$\ \cdot\ _{\mathcal{H}_2}$	\mathcal{H}_2 -norm
$\ \cdot\ _{\mathcal{H}_\infty}$	\mathcal{H}_∞ -norm
$\sigma_i(A)$	i -th singular value of A
$\sigma_{\max}(A)$	largest singular value of A
Σ	diagonal matrix containing singular values
$\operatorname{diag}(d_1, \dots, d_k)$	diagonal matrix with d_1, \dots, d_k on the diagonal
$\Lambda(A, E)$	spectrum of the matrix pair (A, E)
$\lambda_j(A, E)$	j -th eigenvalue of the matrix pair (A, E)
$\operatorname{tr}(A)$	$\sum_{i=1}^n a_{ii}$, where a_{ii} be the diagonal entry of A
$\ker(A)$	kernel of the matrix A
\mathcal{K}_m	basis for the m -dimensional Krylov subspace

List of Acronyms

LTI	Liner Time Invariant
BIPS	Brazilian Interconnected Power System
CALE	Continuous Algebraic Lyapunov Equation
CARE	Continuous Algebraic Riccati Equation
IRKA	Iterative Rational Krylov Algorithm
RKSM	Rational Krylov Subspace Method
TSIA	Two Sided Iterative Algorithm
LQR	Linear Quadratic Regulator
MOR	Model Order Reduction
ROM	Reduced Order Model
EVP	Eigen Value Problem
SPD	Symmetric Positive Definite
SVD	Singular Value Decomposition
ODE	Ordinary Differential Equation
PDE	Partial Differential Equation
DAE	Differential Algebraic Equation
DoF	Degrees of Freedom
FDM	Finite Difference Method
FEM	Finite Element method
BT	Balanced Truncation
TF	Transfer Function
SVD	singular-value decomposition
SFM	Stabilizing Feedback Matrix
HSV	Hankel Singular Value
KN	Kleinman-Newton
ADI	Alternating Direction Implicit
LRCF-ADI	Low-Rank Cholesky Factor-ADI
<code>care</code>	MATLAB library command for solving CARE
<code>lyap</code>	MATLAB library command for solving CALE

Chapter 1

Introduction

1.1 Motivation

Control theory, system analysis, optimization, signal processing, large-scale space flexible structures, game theory, and physical system design are all areas where mathematical modeling is used today. In addition, multi-tasking systems with various components appear in many engineering applications, such as microelectronics, micro-electro-mechanical systems, aerospace, computer control of industrial processes, chemical processes, communication systems, etc. [1]. They are created up of branches of subsystems controlled by extensive mathematical models that use an interconnected inner mathematical system with enormous dimensions. The Linear Time-Invariant (LTI) system is closely related to the Continuous-time Algebraic Riccati Equation (CARE). CAREs may be seen in various fields of science and engineering, particularly in control issues. Modern mathematical models rely heavily on the quadratic cost functional. The solution matrix CARE is utilized to optimize the Linear Quadratic Regulator (LQR) issue, which consists of an optimum control function associated with the LTI continuous-time system, where the quadratic cost functional reaches its infimum [2]. Engineering applications need optimization of LQR in conjunction with descriptor systems (unstable in particular). However, there are no effective computational solutions or analytical tools [3]. When physical models are transformed into mathematical models, their dimensions often grow to enormous proportions, and system analysis is compelled to choose inadequate methodologies. The most significant challenge in storing CARE matrices in computational tools is their size. Simulation approaches need

costly time dealing and are plagued by a slow pace of convergence due to large-scale matrix dimensions. Also, the accuracy of the solution reduces over time for huge and sparse system-oriented structures descriptor systems [4]. Some conventional methods are compatible with only symmetric systems and exploit the system structures. A large-scale system needs more excellent memory and significant processing effort. They also eliminate the need for regular simulations, common in many applications. Because of computer memory limitations, the produced systems are sometimes too vast to store. Therefore, it is unavoidable to reduce the computational size of the systems. Model order reduction (MOR) is a method for reducing a higher-dimensional object to a lower-dimensional [5]. The reduction procedure does not need prior knowledge of the underlying systems, and system features like stability and passivity are included. However, the method must be resilient, and the global error bound must be reduced to a specific margin, as determined by an appropriate norm [6].

1.2 Literature Review

Some Model-Order Reduction (MOR) techniques currently exist. Namely, Singular-Value Decomposition (SVD) based Balanced Truncation (BT) Krylov subspace-based Iterative Rational Krylov Algorithm (IRKA). Those techniques are elaborately discussed in [7, 8] and references therein. From the discussion of the BT method, some obstacles are identified, such as the demands to solve two large dimensional Lyapunov equations for Controllability and Observability gramians, which is very costly for computational time and memory requirement. On the other hand, IRKA is better in simulation time and needs less memory allocation, but ROM stability is not guaranteed [9]. For solving the Riccati equation without explicit estimation of the ROM, some techniques are available in practice, for example, Low-Rank Alternative Direction Implicit (LR-ADI) integrated Newton-Klenman (NK) method and Krylov subspace associated Rational Krylov Subspace Method (RKSM). Those methods are derived and analyzed in detail in [10, 11] and references therein. However, a Newton-Klenman process is a very complex approach, and at each Newton step, LR-ADI iterations need to be executed once, which is a very time-laborious task. Also, the Newton-Klenman process cannot ensure the definiteness of the solution of the Riccati equation derived from an unstable system. On the contrary, RKSM is easy for simulation. Still, the lack of

proper shift parameter selection sometimes makes this method ineffective, and adjusting the stopping criteria may encounter the convergence of the plan. An initial feedback matrix can be a remedy for unstable systems in the RKSM approach, but this additional step makes the whole process inconvenient in time management.

1.3 Objective

To overcome the troubles mentioned above and complexities, we propose an extended form of the Krylov subspace-based Two-Sided Iterative Algorithm (TSIA) [12]. In this strategy, initially, we need to find a ROM of the target model implementing the IRKA approach. Then, two sparse-dense Sylvester equations will be solved to find the system gramians and hence the required projection matrices constituted by their orthonormalized columns constructed via the generalized QR-decomposition [13]. The ROM attained by the TSIA approach is stability preserving, and we will derive the sparsity-preserving form. The rest of the activities will follow the classical inverse projection scheme discussed previously [14]. The \mathcal{H}_2 norm optimality will justify the accuracy of the offered strategies. Finally, the validity of the supplied methodologies will be statistically evaluated using the target models' transient behavior.

1.4 Outlines of the Thesis

This thesis consists of 5 chapters, including this introductory Chapter-1. In Chapter 2, the derivation and fundamental concepts of the systems and control theory. The basic ideas of linear algebra, matrix equations, and instabilities are thoroughly narrated. Some existing methods for solving matrix equations and real-world models are provided briefly. The terms and concepts of this chapter are used throughout the rest of the branches.

Chapter-3 consists of the principal work of the thesis. We discuss the conversion of index-2 systems to generalized systems, the sparsity-preserving structure of Krylov subspace for index-2 systems, the Two-sided iterative algorithm for index-2 systems, and the particular structure of the H_2 -norm estimation in detail.

The numerical computation of the optimal feedback matrices for the incompressible Navier-Stokes models is attained and applied to stabilize the eigenvalues and step-responses of the target models in Chapter-4. Comparative analysis for outputs of the target models is shown. Also, a close discussion of previous work is provided.

Finally, Chapter-5 contains the conclusions of the thesis. The possibilities for improvements and future research are highlighted in brief.

Chapter 2

Preliminaries

2.1 Basic Concepts

In control theory, a state-space representation regulates a physical system as a collection of input, output, and state variables connected by first-order differential equations or difference equations. State variables are regularized by time and the values of input variables, whereas state variables can produce output variables. If the dynamical systems are linear, time-invariant, and finite-dimensional, the differential and algebraic equations can be written as matrices. The state-space system is distinguished by significant algebras of general system theory, allowing Kronecker vector-matrix structures to be utilized. The capacity of these structures may be successfully employed to investigate systems with or without modification. More information on state-space systems and control problems may be found in [15].

2.1.1 State-Space Representations of Control Systems

Consider the following set of ordinary first-order differential equations to represent a dynamical system:

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t),$$

$$\mathbf{y}(t) = C\mathbf{x}(t) + D\mathbf{u}(t),$$

$\mathbf{x}(t) \in \mathbb{R}^n$ is the state vector and $x(t_0) = x_0$ is the initial state. $\mathbf{u}(t) \in \mathbb{R}^p$ is the input (control) vector. $\mathbf{y}(t) \in \mathbb{R}^m$ is the output vector. $E \in \mathbb{R}^{n \times n}$ is the

differential coefficient matrix. $A \in \mathbb{R}^{n \times n}$ is the state space matrix. $B \in \mathbb{R}^{n \times p}$ is the control multiplier matrix. $C \in \mathbb{R}^{m \times n}$ is the state multiplier matrix. $D \in \mathbb{R}^{m \times p}$ is the direct transmission map. If $p = m = 1$ the LTI system is called Single-input Single-output (SISO) system. If $p, m > 1$ the LTI system is called Multi-input Multi-output (MIMO) system.

Any physical system's space-state representation is crucial for analyzing controllability, observability, and stability. Furthermore, the structure of space-state terms reveals the pattern of the target systems.

Assume you have a state-space system with integrators, p inputs $u_1(t), u_2(t), \dots, u_p(t)$ and m outputs $y_1(t), y_2(t), \dots, y_m(t)$. Define the integrators' n outputs as state variables $x_1(t), x_2(t), \dots, x_n(t)$ [16]. The system may thus be described as follows

$$\begin{aligned}
 e_1(t)\dot{x}_1(t) &= f_1(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_p; t), \\
 e_2(t)\dot{x}_2(t) &= f_2(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_p; t), \\
 &\quad \vdots \qquad \qquad \qquad \vdots \qquad \qquad \qquad \vdots \\
 e_n(t)\dot{x}_n(t) &= f_n(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_p; t).
 \end{aligned}
 \tag{2.1}$$

The system's outputs may be described as follows:

$$\begin{aligned}
 y_1(t) &= g_1(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_p; t), \\
 y_2(t) &= g_2(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_p; t), \\
 &\quad \vdots \qquad \qquad \qquad \vdots \qquad \qquad \qquad \vdots \\
 y_m(t) &= g_m(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_p; t).
 \end{aligned}
 \tag{2.2}$$

If we define the matrices below, we can see that

$$\begin{aligned}
 E(t) &= \begin{bmatrix} e_1(t) \\ e_2(t) \\ \vdots \\ e_n(t) \end{bmatrix}, x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix}, y(t) = \begin{bmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_m(t) \end{bmatrix}, u(t) = \begin{bmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_p(t) \end{bmatrix}, \\
 f(x, u, t) &= \begin{bmatrix} f_1(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_p; t) \\ f_2(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_p; t) \\ \vdots \\ f_n(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_p; t) \end{bmatrix}, \\
 g(x, u, t) &= \begin{bmatrix} g_1(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_p; t) \\ g_2(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_p; t) \\ \vdots \\ g_m(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_p; t) \end{bmatrix}.
 \end{aligned} \tag{2.3}$$

Then equations (2.1) and (2.2) as if it were a pair of equations

$$\begin{aligned}
 E(t)\dot{\mathbf{x}}(t) &= f(x, u, t), \\
 \mathbf{y}(t) &= g(x, u, t),
 \end{aligned} \tag{2.4}$$

where (2.4) is the state equation and (2.4) is the output equation. A time-invariant system in which the functions f and g do not explicitly include the time t . If the equations in (2.4) are linearized around the operational state. Then the linear time-invariant state-space system with input-output equations follows.

$$\begin{aligned}
 E\dot{\mathbf{x}}(t) &= A\mathbf{x}(t) + B\mathbf{u}(t), \\
 \mathbf{y}(t) &= C\mathbf{x}(t) + D\mathbf{u}(t),
 \end{aligned} \tag{2.5}$$

Where $E, A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times p}$, $C \in \mathbb{R}^{m \times n}$ and $D \in \mathbb{R}^{m \times p}$, with extremely big n and $p, m \ll n$; respectively, represent mass matrix, system matrix, control multiplier matrix, state multiplier matrix, and direct transmission map (gain). $\mathbf{x}(t) : \mathbb{R} \rightarrow \mathbb{R}^n$ and $\mathbf{u}(t) : \mathbb{R} \rightarrow \mathbb{R}^p$ are the state and control (input) vectors, respectively, in the system (2.5), and $\mathbf{y}(t) : \mathbb{R} \rightarrow \mathbb{R}^m$ is the output vector, with $x(t_0) = x_0$ as the starting state. Direct transmission is missing in the majority of state-space systems, and as a result, $D = O$. The state vector $\mathbf{x}(t)$ has a

dimension of n . However, it will be represented as a compact pair of input and output equations for ease of manipulation.

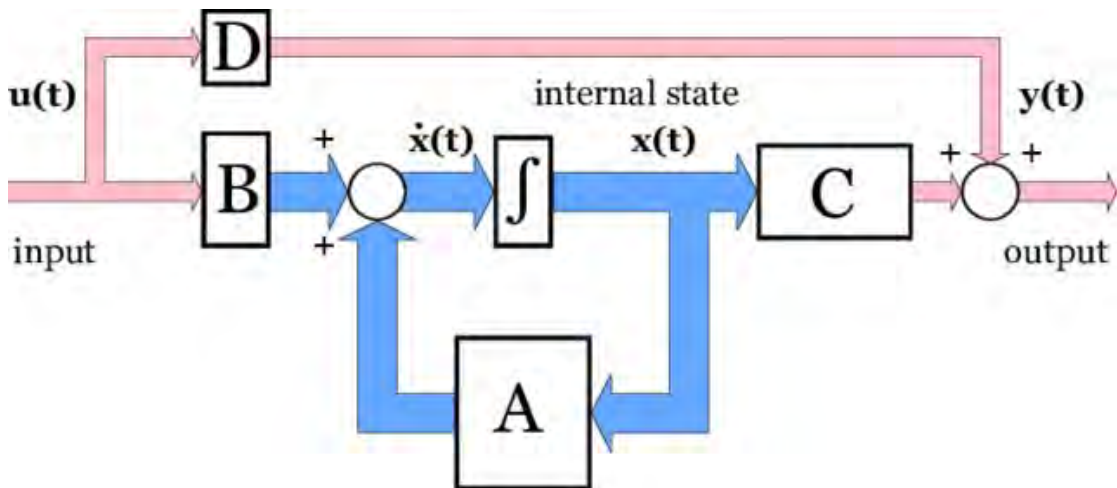


Figure. 2.1. State-space system

2.1.2 Standard and Generalized System

For invertible and symmetric positive definite matrix E , the system (2.5) is said to be generalized.

The system (2.5) is classed as standard if $E = I_n$, is the n -dimensional identity matrix. Because E is invertible, the generalized system may be reduced to the conventional technique of the following form.

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \bar{A}\mathbf{x}(t) + \bar{B}\mathbf{u}(t), \\ \mathbf{y}(t) &= C\mathbf{x}(t) + D\mathbf{u}(t),\end{aligned}\tag{2.6}$$

where $\bar{A} = E^{-1}A$ and $\bar{B} = E^{-1}B$. are the two variables. The conversion (2.6) is not appropriate for real-time practice due to the time-consuming inversion procedure.

2.1.3 Descriptor System

Descriptor systems arise from various physical models and are a particular type of generalized space-state systems with singular matrix E , i.e., $\det(E) = 0$, in current control theory. Singular LTI systems or Differential-Algebraic Equations are other

names for descriptor systems (DAE) [17]. In the modeling of power systems, chemical engineering, and mechanical systems, such methods are regulated .

If the associated matrix pencil is regular,, i.e., $\det(\lambda E - A) \neq 0$. a descriptor system is solvable. According to matrix algebra, non-singular (invertible) transformation matrices T_L and T_R exist for the regular matrix pencil, such that the matrices E and A have the Weierstrass canonical form as follows percent.

$$E = T_L \begin{bmatrix} I_{n_f} & O \\ O & N \end{bmatrix} T_R \quad \text{and} \quad A = T_L \begin{bmatrix} A_1 & O \\ O & I_{n_\infty} \end{bmatrix} T_R, \quad (2.7)$$

where N is nil-potent with nil-potency v , i.e., $N^{v-1} \neq O$ but $N^v = O$, and $n_f + n_\infty = n$. The number v is referred as the algebraic index. The details of the descriptor systems and their derivation is narrated in [18].

This thesis aims to look into specific structured descriptor systems and their applications in engineering. The following are the block matrix form descriptor methods that we are interested in:

$$\underbrace{\begin{bmatrix} E_{11} & E_{12} \\ O & O \end{bmatrix}}_E \underbrace{\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix}}_{\dot{\mathbf{x}}(t)} = \underbrace{\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}}_A \underbrace{\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}}_{\mathbf{x}(t)} + \underbrace{\begin{bmatrix} B_1 \\ B_2 \end{bmatrix}}_B \mathbf{u}(t), \quad (2.8)$$

$$\mathbf{y}(t) = \underbrace{\begin{bmatrix} C_1 & C_2 \end{bmatrix}}_C \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + D\mathbf{u}(t),$$

Model reduction of a class of structured index-2 descriptor systems of the form

$$\underbrace{\begin{bmatrix} E_{11} & 0 \\ O & O \end{bmatrix}}_E \underbrace{\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix}}_{\dot{\mathbf{x}}(t)} = \underbrace{\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & 0 \end{bmatrix}}_A \underbrace{\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}}_{\mathbf{x}(t)} + \underbrace{\begin{bmatrix} B_1 \\ B_2 \end{bmatrix}}_B \mathbf{u}(t), \quad (2.9)$$

$$y(t) = \underbrace{\begin{bmatrix} C_1 & C_2 \end{bmatrix}}_C \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + D\mathbf{u}(t),$$

where $x_1(t) \in \mathbb{R}^{n_1}$, $x_2(t) \in \mathbb{R}^{n_2}$ with $n_1+n_2 = n$ are state vectors and $A_{11}, A_{12}, A_{21}, A_{22}$ are A block matrices with the correct dimensions. E_{11} and A_{11} have full rank in this case. The system (2.8) is termed semi-explicit descriptor system [19]. if the block matrix $E_{12} \neq O$, In some instances, the descriptor system (2.8). must make the assumption that $E_{12} = O$.

Based on physical properties and transitory behavior, the corresponding mathematical models may be developed in various methods. The descriptor systems (2.8) are divided into two categories.

- index-1, if $\det(A_{22}) \neq 0$,
- index-2, if $A_{22} = O$ and $\det(A_{21}A_1) \neq 0$, and
- index-3, if $A_{22} = O$ and $\det(A_{21}A_{12}) = 0$.

2.1.4 Input-Output Relations

In time domain analysis, the step response and frequency response are the two most frequent inputs. The Laplace transformation ¹, the state-space system (2.5) may be represented in the frequency domain.

The system then generates the following form for the complex variable s :

$$\begin{aligned} sEX(s) - x_0 &= AX(s) + BU(s), \\ Y(s) &= CX(s) + DU(s). \end{aligned} \tag{2.10}$$

The Laplace transformations of $\mathbf{x}(t)$, $\mathbf{u}(t)$ and $\mathbf{y}(t)$, are $X(s)$, $U(s)$ and $Y(s)$ respectively. The system (2.10) may be expressed as for $x_0 = 0$,

$$\begin{aligned} X(s) &= (sE - A)^{-1}BU(s), \\ Y(s) &= G(s)U(s). \end{aligned} \tag{2.11}$$

where $G(s) = C(sE - A)^{-1}B + D$. $G(s)$ is the $p \times m$ matrix in MIMO systems, and it may be described as

$$G(s) = \begin{bmatrix} G_{11}(s) & G_{12}(s) & \cdots & G_{1m}(s) \\ G_{21}(s) & G_{22}(s) & \cdots & G_{2m}(s) \\ \vdots & \vdots & \cdots & \vdots \\ G_{p1}(s) & G_{p2}(s) & \cdots & G_{pm}(s) \end{bmatrix} \tag{2.12}$$

¹The Laplace transformation of the function is used to convert data. For any $t \geq 0 \in \mathbb{R}$, the Laplace transformation of the function $f(t)$ is defined as $F(s) = \mathcal{L}[f(t)] = \int_0^\infty f(t)e^{-st}dt$ for the number $s \in \mathbb{C}$.

where $G_{ij} = C(i, :)(sE - A)B(:, j) + D(i, j)$ with the indices $i = 1, 2, \dots, p$ and $j = 1, 2, \dots, m$.

2.1.5 Transfer Function

The transfer function of the system (2.5) is defined by the function $G(s) = C(sE - A)^{-1}B + D$, which was introduced in (2.11) and defined in (2.12). The input-output relationship of state-space systems is represented by the transfer function. The error bound of the reduced order model is established using the transfer function in control theory. If $\lim_{x \rightarrow \infty} G(s) < \infty$ the transfer function $G(s)$ is termed proper, and rigorously proper if $\lim_{x \rightarrow \infty} G(s) = 0$, otherwise $G(s)$ is called improper. The pole of the system is the point s_p where $G(s_p) \rightarrow \infty$.

2.1.6 System Rank

The number of leading entries in a row lowered from R for A is the rank of a matrix A . In R , this is also the number of non-zero rows. The number of leading variables is $\text{rank}[A]$ for any system with A as a coefficient matrix. A matrix's rank is one of its most fundamental characteristics [20]. The rank is commonly denoted by $\text{rank}(A)$. If the maximum conceivable rank for a matrix of the same size is smaller than the number of rows and columns, the matrix is said to be full rank. If a matrix does not have a full rank, it is a rank deficiency. The limited number of rows and columns and the discrepancy in rank cause a matrix to insufficiency rank. The rank of a linear map or operator Φ is defined as the dimension of its image $\text{rank}(\Phi) = \mathbf{dim}(\text{image}(\Phi))$, where \mathbf{dim} is the dimension of a vector space, and \mathbf{image} is the image of a map.

The numeral of solutions to a system of linear equations is one practical use of computing the rank of a matrix. If the position of the augmented matrix is larger than the rank of the coefficient matrix, the system is inconsistent, according to the Rouché–Capelli theorem. If the rankings of these two matrices, on the other hand, are equal, the system must have at least one solution. If the rank equals the number of variables, the solution is unique. Otherwise, there are k free parameters in the general solution, where k is the difference between the number of variables and the rank. The system of equations has an endless number of solutions in this scenario (provided the equations are supernatural or complex numbers). The rank

of a matrix can be used in control theory to assess if a linear system is controllable or observable.

The rank of a function's communication matrix limits the amount of communication required for two persons to compute the process in the field of communication complexity.

2.1.7 Reduced-Order Model

Large-scale space-state systems are regulated by complicated three-dimensional real-world engineering models with sophisticated components. The dimensions of the differential coefficient and system matrices become pretty significant in this circumstance. Simulation methods for these systems necessitate extremely costly time-dealing and are plagued with an infeasible rate of convergence.

In large-scale state-space systems, the size of the matrices is the most challenging component to store in computational tools. Despite the availability of quicker technology and effective modeling methodologies for extensive dimensional systems, The computations are infeasible due to computational complexity and many memory requirements.

As a result, large-scale real-world models must be transformed to Reduced-Order Models (ROM) using iterative approaches such as ADI, RKSM, and IRKA [21], which have numerous applications in engineering systems.

The ROM for the system (2.5) may be calculated as follows:

$$\begin{aligned}\hat{E}\dot{\hat{\mathbf{x}}}(t) &= \hat{A}\hat{\mathbf{x}}(t) + \hat{B}\mathbf{u}(t), \\ \hat{\mathbf{y}}(t) &= \hat{C}\hat{\mathbf{x}}(t) + \hat{D}\mathbf{u}(t),\end{aligned}\tag{2.13}$$

The reduced-order matrices may be produced using the appropriate transformation of the simulation techniques. However, the system design is preserved as invariant as feasible in the methods, and the size of the ROMs should be allocable in terms of memory and time .

The method must be resilient, and the global error bound must be reduced to a given margin, as by an appropriate norm. Also, (2.6) and (2.13) must have the same transfer functions.

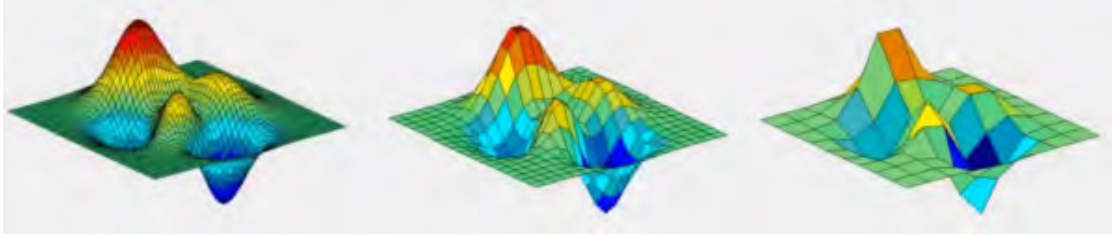


Figure. 2.2. Technique of reduced-order modeling

2.2 Matrix Equations

Some linear and quadratic matrix equations, which have significant applications in control theory, will be introduced in this section. In addition, The problem of a linear quadratic regulator will be briefly discussed.

2.2.1 Riccati Equation

An algebraic Riccati equation is a nonlinear equation that arises in infinite-horizon optimal control theory problems in Discrete or Continuous:

The generalized continuous-time algebraic Riccati equation (CARE) can be written as

$$A^T X E + E^T X A - E^T X B B^T X E + C^T C = 0. \quad (2.14)$$

Here, E , A , B , C are known real coefficient matrices defined by (2.5), while $X \in \mathbb{R}^{n \times n}$ is an unknown symmetric matrix. When a solution is used to regulate the related LQR system, the closed-loop system becomes stable. Using the solution X of the CARE (2.14) purpose of finding the feedback gain K_0 as

$$K_0 = B^T X E, \quad (2.15)$$

and the closed-loop system is

$$A_1 - B_1 K_0 = A_1 - B_1 B^T X E, \quad (2.16)$$

Which is stable for the eigenvalues have a strictly negative real part.

Finding the eigen-decomposition of a big system can also be used to locate the solution [22]. The Hamiltonian matrix is defined for the CARE as

$$Z = \begin{bmatrix} A & -BB^T \\ -C^T C & -A^T \end{bmatrix} \quad (2.17)$$

Let U is comfortably partitioned into four $n \times n$ blocks as

$$U = \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix}, \quad (2.18)$$

Because Z is a Hamiltonian, half of its eigenvalues have a negative real component if there are no eigenvalues on the imaginary axis. In block matrix notation, $2n \times n$ whose columns constitute the basis of the appropriate subspace is denoted as

$$\begin{bmatrix} U_{11} \\ U_{21} \end{bmatrix} \quad (2.19)$$

where U_{11} and U_{21} result from the decomposition

$$X = \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix} \begin{bmatrix} \Lambda_{11} & \Lambda_{12} \\ 0 & \Lambda_{22} \end{bmatrix} \begin{bmatrix} U_{11}^T & U_{12}^T \\ U_{21}^T & U_{22}^T \end{bmatrix} \quad (2.20)$$

Hence we get $X = U_{21}U_{11}^{-1}$ is a Riccati equation solution. Moreover, the eigenvalues of $(A_1 - BB^T X)$ are the eigenvalues of Z with a negative real component.

2.2.2 Lyapunov Equation

Lyapunov equations are the fundamental instruments for large scale state-space systems in MOR approaches. Controllability and observability analysis depend heavily on the Continuous-time Algebraic Lyapunov Equations (CALE) [23].

The generalized CALEs can be organized in the following way:

$$APE^T + EPA^T + BB^T = 0 \quad (2.21)$$

$$A^T QE + E^T QA + C^T C = 0. \quad (2.22)$$

Here P and Q are the controllability and observability Gramians, respectively, and can be defined as

$$P = \int_0^{\infty} e^{At} B B^T e^{A^T t} dt \quad (2.23)$$

$$Q = \int_0^{\infty} e^{A^T t} C^T C e^{At} dt \quad (2.24)$$

Lyapunov equations are the essential part of controllability and observability analysis.

2.2.3 Sylvester Equation

The efficient solution of specially structured Sylvester equations is a crucial ingredient for our H_2 -model order reduction algorithm. Therefore we will discuss a strategy that exploits the particular structure of these equations. We call a Sylvester equation as follows.

$$AXF + EXH + M = 0, \quad (2.25)$$

where $A \in \mathbb{R}^{n \times n}$, $E \in \mathbb{R}^{n \times n}$, and $F \in \mathbb{R}^{r \times r}$, $H \in \mathbb{R}^{r \times r}$, $M \in \mathbb{R}^{n \times r}$. We assume that all requirements for the unique solvability of Equation (2.25) are met in the works [24]. The Sylvester equation defined in (2.25) is of generalized case, which will be used to find the solution in the next section.

2.2.4 Solution of Sylvester Equation

We design an approach that uses this structure after recognizing the need for an efficient solution for sparse-dense Sylvester equations. The resolution of shifted linear systems $(A + pE)$ and the matrix-vector product are permissible operations on matrix A ; element-wise access is prohibited. Furthermore, the eigenvectors and eigenvalues are unknown [25]. Sorensen and Antoulas presented the main idea of solving the desired shifted linear system in [26]. Still, it was rejected because the authors considered the (direct) solution of large-scale linear systems and complex arithmetics to be infeasible.

We can not handle the generalized equation (2.25) directly as it requires more complex approaches with complicated matrix-vector operations. Since the matrix

F is to the right of the first X in the equation (2.25), the generalized case does not operate directly in this fashion (2.25). To solve this problem, we use their generalized Schur decomposition $(Q\hat{F}Z^H; Q\hat{H}Z^H)$ to replace F and H . The QZ algorithm [27]. By using this in (2.25), we have

$$AXQ\hat{F}Z^H + EQ\hat{H}Z^H + M = 0 \quad (2.26)$$

We get a comparable form to the conventional equation by multiplying the equation (2.26) from the right with Z . It gives

$$A \underbrace{XQ}_{\tilde{X}} \hat{F} + E \underbrace{XQ}_{\tilde{X}} \hat{H} + \underbrace{MZ}_{\tilde{M}} = 0 \quad (2.27)$$

The fact that \hat{F} and \hat{H} are upper triangular matrices leads to the following expression for \tilde{X} 's first column:

$$\begin{aligned} A\tilde{x}_1\hat{F}_{11} + E\tilde{x}_1\hat{H}_{11} + \tilde{M}_1 &= 0 \\ (\hat{F}_{11}A + \hat{H}_{11}E)\tilde{x}_1 &= -\tilde{M}_1. \end{aligned} \quad (2.28)$$

All other columns may be generated using a substitution strategy similar to the usual case, but modifying the linear combinations with A and F is necessary. As a result, for each column j of \tilde{X} , we find

$$A(\tilde{X}\hat{F}_{jj} + \sum_{i=1}^{j-1} \hat{F}_{ij}\tilde{X}_i) + E(\tilde{X}\hat{H}_{jj} + \sum_{i=1}^{j-1} \hat{H}_{ij}\tilde{X}_i) + \tilde{M}_j = 0 \quad (2.29)$$

which can be rearranged to

$$(\hat{F}_{jj}A + \hat{H}_{jj}E)\tilde{X}_j = -\tilde{M}_j - \sum_{i=1}^{j-1} (\hat{F}_{ij} \underbrace{A\tilde{X}_i}_{\tilde{X}_i^A} + \hat{H}_{ij} \underbrace{E\tilde{X}_i}_{\tilde{X}_i^E}) \quad (2.30)$$

We must multiply \tilde{X} with Q^H from the right to get the original system's answer. If we precompute the matrix-vector products, \tilde{X}_i^A and \tilde{X}_i^E , as soon as we compute column i of \tilde{X} . We can more quickly evaluate the linear combination on the right-hand side. The estimation of complexity is the same as in the usual case. Even though we require more operations in general, even though we need more functions in general, Algorithm 1 is dominated by the solution of the linear system.

Algorithm 1: Solution of the Sparse-Dense Generalized Sylvester Equation.

Input : $A \in \mathbb{R}^{n \times n}$, $E \in \mathbb{R}^{n \times n}$, $F \in \mathbb{R}^{r \times r}$, $H \in \mathbb{R}^{r \times r}$ and $M \in \mathbb{R}^{n \times r}$, defining (2.25),

Output: $X \in \mathbb{R}^{n \times r}$ solution of (2.25).

- 1 Compute the Schur decomposition $(F, H) = (Q\hat{F}Z^H; Q\hat{H}Z^H)$.
 - 2 Define $\tilde{M} := MZ$
 - 3 **for** $j := 1, \dots, r$ **do**
 - 4 Compute $\hat{M}_j = -\tilde{M}_j - \sum_{i=1}^{j-1} (\hat{F}_{ij}\hat{X}_i^A + \hat{H}_{ij}\hat{X}_i^E)$
 - 5 Solve $(\hat{F}_{jj}A + \hat{H}_{jj}E)\tilde{X}_j = \hat{M}_j$.
 - 6 **if** $j < p$ **then**
 - 7 $\hat{X}_j^A := A\hat{X}_j$
 - 8 $\hat{X}_j^E := E\hat{X}_j$
 - 9 **end if**
 - 10 **end for**
 - 11 $X = \tilde{X}Q^H$
-

2.3 Stability and Related Topics

Stability is another important characteristic of the LTI system. It has many applications in control theory. For example, in the solution of matrix equation and model reduction of the large-scale, In this Thesis, we will exploit this characteristic of some of the basic concepts of stability of a system are discussed elaborately below [28].

2.3.1 Stable and Unstable System

If the output of a system is under control, it is considered stable. It is thought to be unstable otherwise. Stability is defined in control theory as the tendency of a system's response to revert to zero after being perturbed. For example, if all of the matrix pair (A, E) eigenvalues are in the open left half of the complex plane, i.e., C , it is stable; otherwise, it is unstable. The matrix pair (A, E) is semi-stable if a few eigenvalues of the matrix pair (A, E) reside in the open right half of the complex plane yet are highly near the imaginary axis.

A system's reaction to inputs or shocks determines its stability. Stability is a system that remains in a constant state until it is impacted by an external action and returns to that condition when the external activity is eliminated.

In control theory, stability is defined as a measure of the tendency of a system's response to return to zero after being disturbed. The stability of a control system is an indicator of the ability of any system to provide a bounded output when a bounded input is involved in it. Stability is an important characteristic of the LTI system. Therefore, it can be utilized to investigate the characteristics in the solution of matrix equations and model reduction of large-scale systems.

Stability permits the system to reach the steady-state and remain in that state for that particular input even after variation in the system's parameters.

As it is a crucial characteristic thus, the control system's performance shows a high dependency on stability. A stable routine gives your life structure and makes you feel in control. It is considered Hurwitz-stable if all of the eigenvalues of the matrix pair $(A; E)$ are equal. Furthermore, it moreover, it is stated to be in the open left half of the complex plane, i.e., $\lambda \in C^-$. On the other hand, the system is unstable if one of the eigenvalues of the matrix pair $(A; E)$ is in the open right half, i.e., $\lambda \in C^+$, in the complex plane. Similarly, the system's stability is determined according to the matrix pairs stability $(A; E)$. Furthermore, the matrix pair $(A; E)$ is said to be semi-stable if a few eigenvalues of the matrix pair $(A; E)$ are in the open right half of the complex plane but exceptionally near to the imaginary axis while the rest of the eigenvalues lie in the open left half of the complex plane [29].

The idea of detectability is the same as that of stabilized. For example, if the matrix triple $(A^T; C^T; E)$ is stabilized, the matrix triple $(C; A; E)$ is said to be detectable. In short, the following theorems show the stability and detectability requirements [30].

Theorem 2.1 (Stability theorem [28]). *For a given system (2.5) with $x(0) = x_0$, the systems trajectory (i.e., solution) is $\mathbf{x}(t) \rightarrow 0$ as $t \rightarrow \infty$. If all of the matrix's eigenvalues are penciled $P(\lambda) = (\lambda E - A)$.*

Theorem 2.2 (Characterization of Stability). *The statements below are equivalent,*

- $(A, B; E)$ is stabilizable,
- $\text{Rank}[\lambda E - A, B] = n$ for all $\text{Re}(\lambda) \geq 0$,
- For all λ and $x \neq 0$ such that $x^* A = \lambda x^* E$ and $\text{Re}(\lambda) \geq 0$, provided $x^* B \neq 0$.

Theorem 2.3 (Characterization of Detect-ability). *The statements below are equivalent,*

- $(C, A; E)$ is detectable,
- The matrix $\begin{bmatrix} A - \lambda E \\ C \end{bmatrix}$ for all columns has sufficient column rank $\text{Re}(\lambda) \geq 0$,
- For all λ and $\mathbf{x} \neq 0$ such that $A\mathbf{x} = \lambda E\mathbf{x}$ and $\text{Re}(\lambda) \geq 0$, provided $C\mathbf{x} \neq 0$,
- $(A^T, C^T; E)$ is stabilizable.

2.3.2 Feedback Stabilization

When constructing a control system, this is a frequent and powerful technique. Feedback stabilization is a notion that includes a feedback element that provides information on the system's current condition and subsequently makes changes to the system's everyday functioning, as shown in Figure 2.3. The output of any control system is impacted by changes in ambient circumstances or any disruption. As a result, the feedback element changes from the output regularly and returns to the input [31]. However, the essential argument remains because feedback's ability to resist uncertainty makes it so helpful for control purposes.

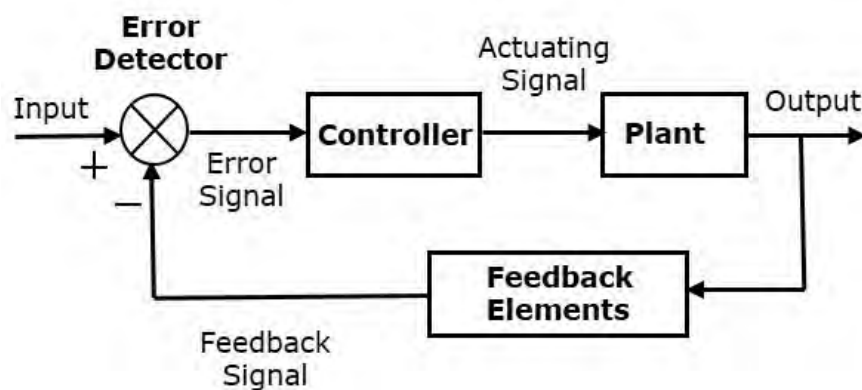


Figure. 2.3. Feedback approach in a system

2.3.3 Riccati Stabilization

In large-scale system simulations, Riccati-based feedback stabilization is the most used technique for stabilization. Determination the convergence of the computationally obtained feedback matrix for the real models is given in [32]. The stabilization of unstable systems around a stationary solution using a Riccati-based feedback matrix has received a lot of interest in recent study, both in terms of control theory and numerical techniques. The challenge in using the LQR method for the target model under consideration is calculating the feedback matrix K_f in such a way that the stabilized system takes on some specified forms. The most difficult job in the Riccati-based feedback stabilization approach is to solve the CARE (2.14) originating from the large-scale model. As a result, the LQR method will be used to compute an approximation to the optimum feedback matrix of the whole system using the reduced-order model (2.13)

The generalized CARE can be expressed in the form (2.13), based on the ROM .

$$\tilde{A}^T \tilde{X} \tilde{E} + \tilde{E}^T \tilde{X} \tilde{A} - \tilde{E}^T \tilde{X} \tilde{B} R^{-1} \tilde{B}^T \tilde{X} \tilde{E} + \tilde{C}^T \tilde{C} = O, \quad (2.31)$$

In terms of matrix dimensions, the generalized CARE (2.31) is feasible and may be solved efficiently for \tilde{X} using any standard solver, such as the MATLAB `care` command.

The stabilizing feedback matrix for the ROM (2.13) may then be calculated using the formula $\tilde{K}_f = R^{-1} \tilde{B}^T \tilde{X} \tilde{E}$ [33]. The full order model's ROM-based approximation to the stabilizing feedback matrix is now $K_f = \tilde{K}_f V^T E$, where V is the transformation matrix used to construct the ROM (2.13).

2.4 Background of Linear Algebra

Basic concepts of linear algebra are required to comprehend the theoretical notion of system and control theory. We will go over some basic linear algebra concepts in this part.

2.4.1 Formation of the Matrices

The matrix' structure has a significant influence on system adaptability and computational convergence. Different sorts of matrix structures can be generated by the governing models. The matrices of a target system can be transformed to a user-defined form to use handy simulation techniques [34].

2.4.2 Sparse and Dense Matrix

If several of a matrix's coefficients are zero and there is no memory allocation for those coefficients, the matrix is sparse. Because its exploitation can result in substantial computing savings, interest in sparsity emerges.

The matrix is dense if the majority of the entries are nonzero. For the non-informative sections, this is a waste of memory resources.

2.4.3 Applications of Sparse and Dense Matrices

If several of a matrices coefficients are zero and there is no memory allocation for those coefficients, the matrix is sparse. The fascination in sparsity stems from the fact that its use can result in considerable computational savings, as well as the fact that many major matrix issues encountered in engineering applications are sparse. Most big matrices are sparse in practice, meaning that virtually all of the entries are zeros. The sparsity of a matrix is defined as the number of zero values in the matrix divided by the total number of elements in the matrix, which is useful in combinations and application fields such as network theory [35].

Large sparse matrices are common in practical machine learning, such as in neural networks. Counts, data encoding that maps categories to counts, and even whole sub-fields of machine learning like natural language processing all contain counts. On computer simulations, sparse matrices are practical for memory allocation and calculation performance. Using specific algorithms and data structures that take use of the sparse structure of the matrices is helpful and frequently required. If a matrix is not sparse, it is said to be dense. In other terms, the matrix is dense if the majority of the entries are nonzero. Many members in dense matrices have zero

values, which is a waste of memory because such zero values carry no information. Standard dense matrix methods are unable to manage some large matrix systems.

2.4.4 Matrix pencil

Let, $A, E \in C^{n \times n}$ the expression $(A - \lambda E)$ with in-determinant, where $\lambda \in C$ is called matrix pencil (or eigen pencil). We denote this by $P(\lambda)$. The terms matrix pair or matrix pencil are used more or less interchangeably, i.e., if any non-zero vector x is an eigenvector of the pencil $(A - \lambda E)$, it is also called an eigenvector of the pair $(A; E)$ [36]. Note that, $\lambda \in C$ is an eigenvalue of $(A; E)$ for $(A - \lambda E)$ is singular that is to say

$$\det(A - \lambda E) = 0. \quad (2.32)$$

This is known as the characteristic equation of the pair $(A; E)$, where the function $\Delta(\lambda) = \lambda E - A$ is the characteristic polynomial with a degree equal to or less than n .

2.4.5 Eigenvalue Problem

For the matrix pair (A, E) , where $A, E \in C^{m \times n}$, an eigenvalue $\lambda \in \mathbb{C}$ and its right eigenvector $\mathbf{x} \in \mathbb{C}^n \setminus \{0\}$ and the left eigenvector $\mathbf{y} \in \mathbb{C}^n \setminus \{0\}$ together form an eigen-triple $(\lambda, \mathbf{x}, \mathbf{y})$ of the matrix pair (A, E) , which satisfies the generalized Eigenvalue Problem (EVP) [37] is defined as

$$A\mathbf{x} = \lambda E\mathbf{x}, \quad \mathbf{y}^* A = \lambda \mathbf{y}^* E. \quad (2.33)$$

The eigenvalues are the roots of the characteristic polynomial $p(\lambda) = \det(A - \lambda E)$ and the spectrum is the set of all eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_n$ corresponding to the matrix pair (A, E) , denoted by $\Lambda(A, E)$ [38]. If E is singular, $\Lambda(A, E)$ contains eigenvalues at infinity and the finite spectrum $\Lambda_f(A, E)$ denotes the set of all finite eigenvalues of the matrix pair (A, E) .

A matrix pair (A, E) is called singular if $A - \lambda E$ is singular for all $\lambda \in \mathbb{C}$, otherwise it called regular [39].

Theorem 2.4 (Eigenvalue Criteria). *Let the matrix pair (A, E) with $A, E \in C^{n \times n}$ and $\lambda \in \mathbb{C}$. Then the following statements are equivalent,*

- λ is an eigenvalue of (A, E) if and only if $\frac{1}{\lambda}$ is an eigenvalue of (E, A) ,
- ∞ is an eigenvalue of (A, E) if and only if E is singular matrix,
- ∞ is an eigenvalue of (A, E) if and only if 0 is an eigenvalue of (E, A) ,
- If E is non-singular, the eigenvalues of (A, E) are exactly the eigenvalues of AE^{-1} and $E^{-1}A$.

The algebraic multiplicity $\alpha(\lambda)$ of a particular eigenvalue is the number of times λ appears as the root of $p(\lambda)$. The number of linearly independent right and left eigenvectors x, y associated to λ is called the geometric multiplicity and denoted by $\zeta(\lambda)$, which satisfies $1 \leq \zeta(\lambda) := \dim\{\ker(A - \lambda E)\} \leq \alpha(\lambda)$. If $\zeta(\lambda) = \alpha(\lambda)$ then λ is called the simple eigenvalue and the corresponding matrix pair (A, E) is called diagonalizable. The following lemmas illustrate the properties of the diagonalizable matrix pair.

Lemma 2.5. *A matrix pair (A, E) with $A, E \in C^{n \times n}$ is diagonalizable if and only if there exists a non-singular matrix $X \in C^{n \times n}$ and $\lambda_1, \lambda_2, \dots, \lambda_n$ are the eigenvalues of the matrix pair (A, E) such that $X^{-1}AX = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$, where the columns of X are eigenvectors of the matrix pair (A, E) .*

Lemma 2.6. *Let the matrix pair (A, E) with $A, E \in C^{n \times n}$ be diagonalizable having distinct eigenvalues $\Lambda(A, E) = \{\lambda_1, \lambda_2, \dots, \lambda_{\hat{n}}\}$ with $n \leq \hat{n}$. Then for the all $i = 1, 2, \dots, \hat{n}$, the relation $\zeta(\lambda_i) = \alpha(\lambda_i)$ holds.*

A matrix pair (A, E) , where $A, E \in C^{n \times n}$ with non-singular E , is called normal matrix pair if it is diagonalizable and its left eigenvectors coincide with the right eigenvectors. The following theorem depicts the properties of the normal matrix pair [40].

2.4.6 Matrix Definiteness

If the scalar $z^T M z$ is strictly positive for every non-zero column vector z of n real numbers, the symmetric matrix $M \in \mathbb{R}^{n \times n}$ is said to be positive definite. z

is acting on an input. when interpreting Mz as the output of an operator M . Positive definiteness refers to the fact that the product always has a positive inner effect with the input, as seen in many physical processes.

A Hermitian matrix $M \in \mathbb{C}^{n \times n}$ is said to be positive definite if the scalar $z^* Mz$ is strictly positive for any non-zero column vector z of n complex numbers, where z^* represents the conjugate transpose of z .

Positive semi-definite matrices are defined identically to negative semi-definite matrices, except that the scalars $z^T Mz$ or $z^* Mz$ must be non-negative. The definitions of negative definite and negative semi-definite matrices are the same. Indefinite refers to a matrix that is neither positive nor negative semi-definite [41].

2.4.7 Hessenberg matrix

A Hessenberg matrix is a kind of square matrix that is "almost" triangular into linear algebra. A lower Hessenberg matrix has zero entries preceding the first super diagonal, whereas an upper Hessenberg matrix has zero entries under the first sub diagonal [42]. When applied to triangular matrices, many linear algebra procedures require substantially less computing effort, and this benefit frequently extends to Hessenberg matrices as well. If the restrictions of a linear algebra issue prevent a general matrix from being reduced to a triangular one, the next best thing is generally reduction to Hessenberg form. Any matrix may be decreased to a Hessenberg form in a limited number of steps. Iterative approaches, such as shifting QR-factorization, can then be used to reduce the Hessenberg matrix to a triangular matrix. Regarding eigenvalue algorithms, Shifted QR-factorization paired with deflation stages can further decrease the Hessenberg matrix to a triangular matrix. Instead of immediately reducing an accessible matrix to a triangular matrix, reducing a general matrix to a Hessenberg matrix and then lowering further to a triangular matrix typically saves arithmetic in the QR approach for eigenvalue issues. A tridiagonal matrix is both upper and lower Hessenberg, of which symmetric or Hermitian Hessenberg matrices are instances.

2.4.8 Projection Matrix

The projection on a vector space V is a linear operator $P : V \mapsto V$ with the property $P^2 = P$. The projection matrix [43]. is thus the square matrix P . If $P^2 = P = P^T$ for a real matrix and $P^2 = P = P^*$ for a complex matrix, the projection matrix P is termed an orthogonal projection matrix. P^* represents the Hermitian transpose of P . A projection matrix P is idempotent by definition, and its eigenvalues must be 0 or 1. The qualities of an orthogonal projector are represented by the lemma .

Lemma 2.7. *When a projector projects onto a subspace S_1 along a subspace S_2 , it is said to be an orthogonal projector if and only if $S_1, S_2 \in \mathbb{C}^n$ are orthogonal sub-spaces such that $S_1 \cap S_2 = \{0\}$ and $S_1 + S_2 = \mathbb{C}^n$, where $S_1 + S_2$ represents the span of S_1 and S_2 , i.e., the set of vectors $s_1 + s_2$ with $s_1 \in S_1$ and $s_2 \in S_2$, respectively.*

If P is a projector, $I - P$ is a complementary projector that fulfills the condition $(I - P)^2 = I - P$.

2.4.9 Matrix Decomposition Techniques

Matrix decomposition (factorization) is a computational method used in control theory to derive ROMs for large-scale systems. There are many ways to matrix decomposition, and some of the most popular will be covered in this section.

2.4.10 Singular-Value Decomposition

One of the most helpful matrix decomposition methods used in control systems, signal processing, and statistics is Singular-Value Decomposition (SVD). It's a powerful tool for creating ROMs. It is an extension of the polar decomposition. That extends the eigenvalue decomposition of a positive semi-definite standard matrix to any $m \times n$ matrix.

The SVD of A as the matrix factorization may be defined as $A \in \mathbb{C}^{m \times n}; m, n \in \mathbb{R}$,

$$A = U\Sigma V^*, \tag{2.34}$$

where $U \in \mathbb{C}^{m \times m}$ and $V \in \mathbb{C}^{n \times n}$ are unitary matrices and $\Sigma \in \mathbb{R}^{m \times n}$ is a diagonal matrix. The singular values of A , which are non-negative and in decreasing order, are represented by the diagonal elements $\sigma_j; j = 1, 2, \dots, k$ of Σ i.e., $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_k \geq 0$, where $k = \min(m, n)$.

The SVD obtained by taking only the first m singular values of A is the thin SVD. The following theorem represents the properties of the SVD [44].

Theorem 2.8 (Properties of SVD). *The following assertions are valid for the singular-value decomposition of a matrix A :*

- *The square roots of the eigenvalues of the symmetric positive semi-definite matrix $A^T A$, are the singular-values σ_j of A*
- *The eigenvectors of the matrix AA^T , are the right singular-vectors, while the eigenvectors of the matrix $A^T A$, are the left singular-vectors.*
- *If the rank of A is r , the number of non-zero singular values, and A is the sum of rank-one matrices,*
- *If $A = A^*$, then the singular-values of A are the absolute values of the eigenvalues of A ,*
- *For $A \in \mathbb{C}^{n \times n}$, $\det(A) = \prod_{j=1}^n \sigma_j$,*
- *$\|A\|_2 = \sigma_1$ and $\|A\|_F = \sqrt{\sum_{k=1}^r \sigma_k^2}$.*

2.4.11 Eigenvalue Decomposition

Eigenvalue decomposition is the process of factoring a matrix into a canonical form, and the matrix is characterized using eigenvalues and eigenvectors. Eigenvalue decomposition is also known as spectral decomposition.

The eigenvalue decomposition of a square matrix A may be described as if the columns $V \in \mathbb{C}^{n \times n}$ contain linearly independent eigenvectors of a square matrix $A \in \mathbb{C}^{n \times n}$,

$$A = V \Lambda V^{-1} \tag{2.35}$$

where $\Lambda \in \mathbb{C}^{n \times n}$ is a diagonal matrix whose elements are the eigenvalues of A .

2.4.12 Schur Decomposition

The Schur decomposition permits an arbitrary matrix to be represented as unitarily equivalent to an upper triangular matrix with diagonal components equal to the original matrix's eigenvalues [45]. Another name for it is the Schur triangulation. For a square matrix $A \in \mathbb{C}^{n \times n}$, the Schur decomposition can be fined as

$$A = UTU^*, \quad (2.36)$$

Where $U \in \mathbb{C}^{n \times n}$ is a unitary matrix, and T is an upper triangular matrix, often known as the Schur form of A . T has the same spectrum as A since it is triangular and similar, and the eigenvalues of A are the diagonal elements of T .

2.4.13 QR Decomposition

QR decomposition is a technique for determining an orthogonal matrix with a set of matrices. QR decomposition is the foundation for a specific eigenvalue method and is frequently used to solve linear least square problems.

A QR decomposition is when a matrix $A \in \mathbb{C}^{m \times n}$ is factored into a matrix product defined as

$$A = QR \quad (2.37)$$

where Q is an upper triangular matrix and R is an orthogonal matrix, i.e., $QQ^T = I = Q^TQ$. The factorization is unique for an invertible matrix A , and the diagonal elements of R are positive .

The modified Gram-Schmidt process and Householder transformations are two techniques to compute QR decomposition.

2.4.14 Cholesky Decomposition

The factorization of a Hermitian, positive-definite matrix into the product of a lower triangular matrix and its conjugate transpose is known as Cholesky decomposition, and it is useful in many computer applications, including Monte-Carlo simulations [46].

Let $A \in \mathbb{C}^{m \times n}$ be a Hermitian, positive definite matrix. The decomposition of A by Cholesky is defined as

$$A = LL^* \tag{2.38}$$

where L^* is the conjugate transpose of a lower triangular matrix L with definite and positive diagonal elements, Every positive-definite Hermitian matrix has its Cholesky decomposition .

Cholesky decomposition may be used to solve the linear system of equations $Ax = b$, The real symmetric and positive-definite matrix A is used here.

2.4.15 Arnoldi Decomposition

The Arnoldi decomposition is a useful iterative solution [47] for constructing the basis for the Krylov subspace. It's usually an effective sparse matrix method that uses the matrix map vectors instead of directly executing the matrix components. The Krylov-based Arnoldi method is one of the most powerful tools for computing the eigenvalues of huge sparse matrices.

Choose an orthogonal projector $V \in \mathbb{R}^{n \times p}$ and a $A \in \mathbb{R}^{n \times n}$. The m -th-dimensional Krylov matrix based on A and V is therefore defined as

$$\mathcal{K}_m(A, V) = [V, AV, A^2V, \dots, A^{m-1}V]. \tag{2.39}$$

Matrix-vector products play the key role in generating the Krylov subspace (2.39) by a recursive technique. The orthogonal columns of the matrix $V_{m+1} = [V_m, v_{m+1}]$ the Krylov subspace has an orthogonal basis \mathcal{K}_m . Also, there exists an unreduced upper Hessenberg matrix $\hat{H}_m \in \mathbb{R}^{(m+1) \times m}$ such that $AV_m = V_{m+1}\hat{H}_m$. By a suitable partition of \hat{H}_m , we can write

$$\begin{aligned} AV_m &= \begin{bmatrix} V_m & v_{m+1} \end{bmatrix} \begin{bmatrix} H_m \\ h_{m+1,m}e_m^T \end{bmatrix}, \\ &= V_m H_m + h_{m+1,m}v_{m+1}e_m^T. \end{aligned} \tag{2.40}$$

Here, H_m can be obtained from \hat{H}_m by removing the last row and e_m in the matrix of the last p columns of the mp -th order identity matrix $I_{m \times p}$ and after m steps $h_{m+1,m}$ will be vanished. So that, after a certain number of iterations, the second

Algorithm 2: Arnoldi decomposition (Modified Gram-Schmidt).

Input : A, C , orthogonal matrix V_m .

Output: Matrix $Z_m \in \mathbb{R}^{n \times m}$ such that $X_m \approx Z_m Z_m^T$.

- 1 Compute $C^T = QR$ (QR factorization);
 - 2 Assume $V_1 = Q = v_1$;
 - 3 **for** $j \leftarrow 1$ to m **do**
 - 4 Compute $w_j = Av_j$;
 - 5 **for** $i \leftarrow 1$ to j **do**
 - 6 Compute $h_{i,j} = v_i^T w_j$;
 - 7 Update $w_j = w_j - h_{i,j} v_i$;
 - 8 **end for**
 - 9 Compute $h_{j+1,j} = \|w_j\|_2$;
 - 10 Update $v_{j+1} = \frac{w_j}{h_{j+1,j}}$;
 - 11 Compute $H_j = \begin{bmatrix} H_{j-1} & h_j \\ O & h_{j+1,j} \end{bmatrix}$;
 - 12 Update $V_{j+1} = [V_j, v_{j+1}]$;
 - 13 Partition $\hat{H}_j = \begin{bmatrix} H_j \\ h_{j+1,j} e_j^T \end{bmatrix}$;
 - 14 **end for**
-

term of (2.40) will be converted to zero. Thus, by the orthogonality property of v_{m+1} , (2.40) provides the following projection

$$H_m = V_m^T A V_m. \quad (2.41)$$

Hence, the term H_m represents the projection A onto the Krylov subspace $\mathcal{K}_m(A, V)$ [48]. The Arnoldi decomposition is summarized in the Algorithm-2.

The eigenvalues λ_i of a projection matrix H_m in the Krylov subspace $\mathcal{K}_m(A, V)$, are known as the Ritz values and if χ is an eigenvector of H_m associate with λ [49]. then $V_m \chi$ is called the Ritz vector belong to λ

2.5 Existing Methods

The CARE is a crucial component in analyzing system stability and structural phenomena in control theory. The solution of CAREs regulated by the matrices of the relevant systems is necessary to study the transient behaviors of various disciplines of engineering professions [50]. Simulation approaches improve over time due to the steady growth in system size and complexity. For large-scale

systems, projection-based iterative techniques have shown to be highly successful since they allow for sparsity patterns and offer low-rank approximated systems while retaining the characteristics of the original systems [51]. The Galerkin projection technique is used in several iterative approaches to obtain a viable solution for the CAREs. This section will offer several basic and newly found techniques for dealing with CAREs.

2.5.1 Schur Decomposition Method

Schur decomposition method based on Real Schur Factorization (RSF) of the Hamiltonian matrix for the CARE is one of the fundamental and oldest methods [52]. Consider the converted to the standard system (2.6) and corresponding CARE can be defined as

$$\bar{A}^T X + X \bar{A} - X \bar{B} \bar{B}^T X + C^T C = O, \quad (2.42)$$

where $\bar{A} \in \mathbb{R}^{n \times n}$, $\bar{B} \in \mathbb{R}^{n \times p}$, $C \in \mathbb{R}^{m \times n}$ and $R \in \mathbb{R}^{p \times p}$.

Assuming (A, B) is a stabilizable pair and (C, A) is detectable pair, whereas both of the pairs have full rank. So, the CARE (2.42) has a unique non-negative definite solution X . For the CARE (2.42) the Hamiltonian matrix can be written as

$$H = \begin{bmatrix} \bar{A} & -\bar{B} \bar{B}^T \\ -C^T C & -\bar{A}^T \end{bmatrix} \in \mathbb{R}^{2n \times 2n}. \quad (2.43)$$

To find the finite solution X of (2.42), it should be ensured that H has no pure imaginary eigenvalues [53]. Thus an orthogonal transformation matrix $U \in \mathbb{R}^{2n \times 2n}$ need to be found that puts H in ordered RSF as

$$S = U^T H U = \begin{bmatrix} S_{11} & S_{12} \\ O & S_{22} \end{bmatrix}, \quad (2.44)$$

where $S_{ij} \in \mathbb{R}^{n \times n}$. The eigenvalues of H with negative real parts have been stacked in S_{11} and those with positive real parts are stacked in S_{22} . Let U is comfortably partitioned into four $n \times n$ blocks as

$$U = \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix}, \quad (2.45)$$

Algorithm 3: Schur decomposition method.

Input : \bar{A} , \bar{B} , and C .

Output: The unique stabilizing solution X of the CARE.

- 1 Form the Hamiltonian matrix $H = \begin{bmatrix} \bar{A} & -\bar{B}\bar{B}^T \\ -C^T C & -\bar{A}^T \end{bmatrix}$;
 - 2 Transform H to the RSF $S = U^T H U = \begin{bmatrix} S_{11} & S_{12} \\ O & S_{22} \end{bmatrix}$;
 - 3 Partition U conformably $U = \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix}$;
 - 4 Compute $X = U_{21} U_{11}^{-1}$.
-

where the following relation is true

$$H \begin{bmatrix} U_{11} \\ U_{21} \end{bmatrix} = \begin{bmatrix} U_{11} \\ U_{21} \end{bmatrix} S_{11}. \quad (2.46)$$

Then, $X = U_{21} U_{11}^{-1}$ is the unique stabilizing solution of the CARE (2.42). The Schur decomposition method is summarized in the Algorithm-3.

2.5.2 Iterative Rational Krylov Algorithm

Iterative Rational Krylov Algorithm (IRKA) can be applied to construct an r -dimensional ($r \ll n$) reduced-order model

$$\begin{aligned} \hat{E} \dot{\hat{\mathbf{x}}}(t) &= \hat{A} \hat{\mathbf{x}}(t) + \hat{B} \mathbf{u}(t), \\ \hat{\mathbf{y}}(t) &= \hat{C} \hat{\mathbf{x}}(t) + \hat{D} \mathbf{u}(t), \end{aligned} \quad (2.47)$$

such that its transfer function $\hat{G}(s) = \hat{C}(s\hat{E} - \hat{A})^{-1}\hat{B} + \hat{D}$ interpolates the original one, $G(s)$, at selected points in the complex plane along with selected directions. The points are called interpolation points and the directions are called tangential directions. We use the procedure illustrated in [54]. to make this problem more precisely as follows.

Initially, we consider a set of ad-hoc interpolation points $\{\alpha_i\}_{i=1}^r$, right tangential directions $\{b_i\}_{i=1}^r$ and left tangential directions $\{c_i\}_{i=1}^r$ to construct two $n \times r$

projection matrices

$$\begin{aligned} V &= [(\alpha_1 E - A)^{-1} B b_1, \dots, (\alpha_r E - A)^{-1} B b_r], \\ W &= [(\alpha_1 E - A)^{-T} C^T c_1, \dots, (\alpha_r E - A)^{-T} C^T c_r]. \end{aligned} \quad (2.48)$$

Then the interpolation points and those tangential directions need to be updated until the reduced transfer function interpolates the original transfer function reasonably. Since the continuous updates of the interpolation points gradually match the eigenvalues of the system, the initial ad-hoc consideration will not affect the convergence of the approach.

By the Petrov-Galerkin condition [55] we can construct the reduced-order matrices in (2.47) as

$$\hat{E} := W^T E V, \quad \hat{A} := W^T A V, \quad \hat{B} := W^T B, \quad \hat{C} := C V, \quad \hat{D}_a := D_a. \quad (2.49)$$

For $i = 1, 2, \dots, r$, the following Hermite bi-tangential interpolation conditions are satisfied by the above reduced-order model.

$$G(\alpha_i) b_i = \hat{G}(\alpha_i) b_i, \quad (2.50)$$

$$c_i^T G(\alpha_i) b_i = c_i^T \hat{G}(\alpha_i) b_i, \quad (2.51)$$

$$c_i^T G'(\alpha_i) b_i = c_i^T \hat{G}'(\alpha_i) b_i. \quad (2.52)$$

Using the reduced-order matrices defined in (2.49), the reduced-order CARE can be attained as

$$\hat{A}^T \hat{X} \hat{E} + \hat{E} \hat{X} \hat{A} - \hat{E} \hat{X} \hat{B} \hat{B}^T \hat{X} \hat{E} + \hat{C}^T \hat{C} = 0. \quad (2.53)$$

Thus, the reduced-order CARE (2.53) can be solved for \hat{X} by the MATLAB library command `care`. Then, the stabilizing feedback matrix $\hat{K} = \hat{B}^T \hat{X} \hat{E}$ corresponding to the ROM (2.13) can be estimated, and hence for stabilizing the full model (2.5) the approximated optimal feedback matrix K^o can be retrieved by the scheme of reverse projection as

$$K^o = (\hat{B}^T \hat{X} \hat{E}) V^T E = \hat{K} V^T E. \quad (2.54)$$

The Iterative Rational Krylov Algorithm (IRKA) introduced in [56] resolves the problem by iteratively correcting the interpolation points and the directions as summarized in Algorithm 4.

Algorithm 4: IRKA for generalized systems.

Input : E, A, B, C, D_a .

Output: $\hat{A}, \hat{B}, \hat{C}, \hat{D}_a := D_a$.

- 1 Make the initial selection of the interpolation points $\{\alpha_i\}_{i=1}^r$ and the tangential directions $\{b_i\}_{i=1}^r$ and $\{c_i\}_{i=1}^r$.
- 2 Construct

$$V = [(\alpha_1 E - A)^{-1} B b_1, \dots, (\alpha_r E - A)^{-1} B b_r],$$

$$W = [(\alpha_1 E^T - A^T)^{-1} C^T c_1, \dots, (\alpha_r E^T - A^T)^{-1} C^T c_r].$$

- 3 **while** (*not converged*) **do**

- 4 Compute $\hat{E} = W^T E V$, $\hat{A} = W^T A V$, $\hat{B} = W^T B$ and $\hat{C} = C V$.

- 5 Compute $\hat{A} z_i = \lambda_i \hat{E} z_i$ and $y_i^* \hat{A} = \lambda_i y_i^* \hat{E}$

- 6 $\alpha_i \leftarrow -\lambda_i$, $b_i^* \leftarrow -y_i^* \hat{B}$ and $c_i^* \leftarrow \hat{C} z_i^*$, for $i = 1, \dots, r$.

- 7

$$V = [(\alpha_1 E - A)^{-1} B b_1, \dots, (\alpha_r E - A)^{-1} B b_r],$$

$$W = [(\alpha_1 E^T - A^T)^{-1} C^T c_1, \dots, (\alpha_r E^T - A^T)^{-1} C^T c_r].$$

- 8 Construct the reduced-order matrices $\hat{E} = W^T E V$, $\hat{A} = W^T A V$, $\hat{B} = W^T B$ and $\hat{C} = C V$.

- 9 $i = i + 1$.

- 10 **end while**

- 11 Construct the reduced-order matrices $\hat{E} = W^T E V$, $\hat{A} = W^T A V$, $\hat{B} = W^T B$ and $\hat{C} = C V$.

2.5.3 Alternative Direction Implicit Method

Peaceman *et al.* developed the Alternative Direction Implicit (ADI) iteration in 1955 for solving the linear system $MX = b$, where the matrix $M = M_1 + M_2 \in \mathbb{R}^{n \times n}$ is symmetric, positive definite, and arises in the numerical solution of partial differential equations. If M is a centered finite difference discretization in two-dimensional space, then M_1 and M_2 are finite difference discretizations in the x and y directions, respectively. $i = 1, 2, \dots$ (when converges) is the iteration for ADI (when converges) As a percentage, it's calculated in double steps.

$$(M_1 + \mu_i I_n) X_{i-\frac{1}{2}} = (\mu_i I_n - M_2) X_{i-1} + b,$$

$$(M_2 + \mu_i I_n) X_i = (\mu_i I_n - M_1) X_{i-\frac{1}{2}} + b,$$
(2.55)

where $\mu_i \in \mathbb{R}^+$ are the shift parameters required for the above ADI iteration to converge at a super-linear rate .

Consider the ADI model issue with a generalized CALE [57] defined as

$$A^T X E + E^T X A = -C^T C, \quad (2.56)$$

The original ADI scheme (2.55) for generalized continuous-time LTI system (2.5) can then be redefined as follows:

$$\begin{aligned} (A^T + \mu_i E^T) X_{i-\frac{1}{2}} &= -C^T C - X_{i-1} (A - \mu_i E), \\ (A^T + \mu_i E^T) X_i^T &= -C^T C - X_{i-\frac{1}{2}}^T (A - \mu_i E), \end{aligned} \quad (2.57)$$

The shift parameters $\mu_i \in \mathbb{C}^-$ are acceptable, and the starting iteration is $X_0 = X_0^T \in \mathbb{R}^{n \times n}$. The single-step ADI scheme may be obtained as (2.57) after the equations in (2.57) have been eliminated and simplified.

$$\begin{aligned} X_i &= (A^T + \mu_i E^T)^{-1} (A^T - \bar{\mu}_i E^T) X_{i-1} (A^T - \bar{\mu}_i E^T)^T (A^T + \mu_i E^T)^{-T} \\ &\quad - 2\text{Re}(\mu_i) (A^T + \mu_i E^T)^{-1} C^T C (A^T + \mu_i E^T)^{-T}. \end{aligned} \quad (2.58)$$

Let $Z_I \in \mathbb{R}^{n \times ip}$ be a low-rank Cholesky-factor of $X_I \in \mathbb{R}^{n \times n}$ with the property $X_i = Z_i Z_i^T$ [58]. The matrix Z_i does not have to be a square matrix or have a lower triangular structure. The low-rank ADI scheme may then be determined using Cholesky factorization in (2.58).

$$\begin{aligned} Z_0 &= O_{n \times p}, \\ Z_i Z_i^T &= \{(A^T + \mu_i E^T)^{-1} (A^T - \bar{\mu}_i E^T) Z_{i-1}\} \{((A^T + \mu_i E^T)^{-1} (A^T - \bar{\mu}_i E^T) Z_{i-1})^T \\ &\quad - 2\text{Re}(\mu_i) \{(A^T + \mu_i E^T)^{-1} C^T\} \{(A^T + \mu_i E^T)^{-1} C^T\}^T \in \mathbb{R}^{n \times ip}. \end{aligned} \quad (2.59)$$

So, on the left side of (2.58) Z_i may be derived by combining two elements on the right as follows:

$$Z_i = \left[\sqrt{-2\text{Re}(\mu_i)} (A^T + \mu_i E^T)^{-1} C^T \quad (A^T + \mu_i E^T)^{-1} (A^T - \bar{\mu}_i E^T) Z_{i-1} \right]. \quad (2.60)$$

As a result, the ADI technique may be rewritten in terms of the Cholesky factor Z_i of X_i , with no need to estimate or store X_i at each iteration because only Z_i is required [59]. The initial Cholesky-factor version of low-rank ADI, which

computes the Cholesky-factor Z_i of X_i as

$$\begin{aligned} Z_1 &= \sqrt{-2\text{Re}(\mu_1)}(A^T + \mu_1 E^T)^{-1} C^T \in \mathbb{R}^{n \times p}, \\ Z_i &= \left[\sqrt{-2\text{Re}(\mu_i)}(A^T + \mu_i E^T)^{-1} C^T \quad (A^T + \mu_i E^T)^{-1}(A^T - \bar{\mu}_i E^T)Z_{i-1} \right] \in \mathbb{R}^{n \times ip}. \end{aligned} \quad (2.61)$$

Consider the following statement :

$$\begin{aligned} \gamma_i &= \sqrt{-2\text{Re}(\mu_i)}, \\ F_i &= (A^T + \mu_i E^T)^{-1} C^T, \\ G_i &= (A^T + \mu_i E^T)^{-1}(A^T - \bar{\mu}_i E^T), \\ H_{i,j} &= (A^T - \bar{\mu}_i E^T)(A^T + \mu_i E^T)^{-1}. \end{aligned} \quad (2.62)$$

Then, under the preceding method, the low-rank Cholesky-factor Z_i may be expressed as

$$Z_i = \left[\gamma_i F_i, \gamma_{i-1} G_i F_{i-1}, \gamma_{i-2} G_i G_{i-1} F_{i-2}, \dots, \gamma_1 G_i G_{i-1} \dots G_1 F_1 \right]. \quad (2.63)$$

Using the commutative identities in $G_i F_i = H_{i,j} F_i$, $G_i G_j = H_{i,j} H_{j,i}$, $\forall i, j$, the equation (2.63) can be re-written as

$$Z_i = \left[\gamma_i F_i, \gamma_{i-1} H_{i-1,i} F_i, \gamma_{i-2} H_{i-2,i-1} H_{i-1,i} F_i, \dots, \gamma_1 H_{1,2} \dots H_{i-1,i} F_i \right]. \quad (2.64)$$

Reversing the order of the shift parameters provides the LR-ADI iterations as for $i \geq 1$.

$$\begin{aligned} V_1 &= (A^T + \mu_1 E^T)^{-1} C^T, \\ Z_1 &= \gamma_1 V_1 = \sqrt{-2\text{Re}(\mu_1)}(A^T + \mu_1 E^T)^{-1} C^T, \\ V_i &= H_{i-1,i} V_{i-1} = V_{i-1} - (\mu_i + \bar{\mu}_{i-1})(A^T + \mu_i E^T)^{-1} E^T V_{i-1}, \\ Z_i &= \begin{bmatrix} Z_{i-1} & \gamma_i V_i \end{bmatrix} = \begin{bmatrix} Z_{i-1} & \sqrt{-2\text{Re}(\mu_i)} V_i \end{bmatrix}. \end{aligned} \quad (2.65)$$

The above procedure is summarized in Algorithm-(5).

Algorithm 5: LRCF-ADI for generalized systems.

Input : E, A, C, i_{max} (number of iterations) and shift parameters $\{\mu_j\}_{j=1}^{i_{max}}$.

Output: Low-rank Cholesky-factor Z such that $X \approx ZZ^T$.

```

1 Consider  $Z_0 = []$ ;
2 for  $i \leftarrow 1$  to  $i_{max}$  do
3   if  $i = 1$  then
4     | Solve  $V_1 = (A^T + \mu_1 E^T)^{-1} C^T$ .
5   else
6     | Compute  $V_i = V_{i-1} - (\mu_i + \bar{\mu}_{i-1})(A^T + \mu_i E^T)^{-1} E^T V_{i-1}$ .
7   end if
8   Update  $Z_i = [Z_{i-1} \quad \sqrt{-2\text{Re}(\mu_i)} V_i]$ .
9 end for

```

2.5.4 Kleinman-Newton Method

The solution of CARE equations, particularly those originating from large-scale control systems, is a time-consuming and challenging process. The well-known Kleinman-Newton technique is frequently used to solve the CARE. Iterative solutions for the solution of linear systems happening at each Newton step are apparent to minimize simulation time. Controlling the accuracy of linear system solutions and increasing efficiency without sacrificing convergence rate are key achievements of approximation in computing [60]. The Kleinman-Newton iterative technique achieves the necessary results while maintaining the appropriate inner iteration termination, which is time consuming .

The generalized CARE (2.14) must be transformed to the following generalized CALE in the Klenman-Newton technique.

$$\hat{A}^T X E + E^T X \hat{A} = -W W^T, \quad (2.66)$$

where $\hat{A} = A - B B^T X E$ and $W = \begin{bmatrix} C^T & E^T X B \end{bmatrix}$. The generalized CALE (2.66) can be solved by any existing Lyapunov solvers, i.e., Bartels-Stewart's method [61].

The simulations of the generalized CALE (2.66) are not cheap for direct solvers due to the constantly increasing size of the control system matrices. As a result, iterative solvers with MOR methods must be implemented . The ADI approach is one of the most commonly used iterative strategies for solving the generalized CALE (2.66).

2.5.5 Rational Krylov Subspace Method

The eminent Russian naval engineer and applied mathematician **A. N. Krylov** (1863–1945) [62]. developed the Krylov subspace and the methods based on it. This approach aims to approximate a large-scale system with a lower-dimensional system that is invariant when compared to the original system. The ROMs can replace the original system or as a component of more significant in real-time simulations [63]. The Rational Krylov Subspace Method (RKSM) is one of the most efficient MOR algorithms for state-space systems recently created.

RKSM is a projection-based iterative method using the block Arnoldi or Lanczos processes. Due to recent advancements and expansions, this iterative approach is now competitive with ADI-based techniques. In compared to traditional methods for solving linear matrix equations.

RKSM has superior results with the flexibility of interpolation point selection in real-world applications [64].

For $X \in \mathbb{R}^{n \times m}$, $V \in \mathbb{R}^{n \times m}$, an acceptable solution to the CARE is derived in the form of $X \approx V\tilde{X}V^T$. The columns of V , which span an orthonormal basis for the mp -dimensional Krylov subapace described by, must first be identified.

$$\mathcal{K}_m(A, B) = \text{span}(B, AB, A^2B, \dots, A^{m-1}B). \quad (2.67)$$

The Arnoldi method, which is based on a modified Gram-Schmidt process described in Algorithm-(2), can calculate the orthogonal basis $V_m = [v_1, v_2, \dots, v_m]$ from the Krylov subspace \mathcal{K}_m . The primary goal now is to derive a Riccati equation with a lower order.

$$H_m^T Y_m G_m + G_m^T Y_m H_m - G_m^T Y_m V_m^T B B^T V_m Y_m G_m + V_m^T C^T C V_m = O, \quad (2.68)$$

It has a unique solution if and only if $\lambda_i + \lambda_j \neq 0$ for any combination of eigenvalues λ_i and λ_j for the real matrix H_m .

If A is sparse, the most challenging component of the algorithm is finding the orthogonal columns of V_m using modified Gram-Schmidt. The residual R_m corresponding determines the Arnoldi method's convergence for solving CARE to Y_m

which is defined by

$$R_m = A^T V_m Y_m V_m^T E + E^T V_m Y_m V_m^T A - E^T V_m Y_m V_m^T B B^T V_m Y_m V_m^T E + C^T C. \quad (2.69)$$

The solution Y_m of (2.68) may be calculated using RKSM methods, ensuring that the Galerkin condition $V_m^T R_m V_m = O$ is satisfied.

2.6 System norms

The norms of vectors and matrices are helpful in the stability analysis and applications of iterative techniques, such as halting criteria and convergence analysis.

To justify the accuracy of desired computations, we need to apply suitable system norms. There are several system norms to define, measure, and imply the system properties. Out of those system norms H_2 and H_∞ norm are well established and widely implemented. In the model reduction of large-scale dynamical systems, both of the norms are instrumental. The H_2 and H_∞ norms are used to calculate the deviation of the original and reduced-order models. These two standards are briefly introduced in the subsections that follow.

2.6.1 Matrix Norms

If $A = [a_{ij}] \in \mathbb{R}^{n \times n}$, then the matrix norm $\|\cdot\| : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}$ is defined as induced by a vector p -norm.

$$\|A\|_p = \sup_{x \neq O} \frac{\|Ax\|_p}{\|x\|_p}. \quad (2.70)$$

The matrix norms induced by vector p -norms are sometimes called generated p -norms satisfying the properties of the vector norm. In particular, the column-sum

norm, spectral norm, and row-sum norm [65] can be defined as

$$\begin{aligned}\|A\|_1 &= \max_j \sum_{i=1}^n |a_{ij}|; \\ \|A\|_2 &= \sqrt{\lambda_{\max}(A^*A)}, \\ \|A\|_\infty &= \max_i \sum_{j=1}^n |a_{ij}|.\end{aligned}$$

The Frobenius norm is an essential matrix norm that is not induced by a vector norm. The Frobenius norm of a matrix $A = [a_{ij}] \in \mathbb{R}^{n \times n}$ is defined as $\|A\|_F$

$$\|A\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2} = \sqrt{\text{tr}(AA^*)} = \sqrt{\text{tr}(A^*A)}. \quad (2.71)$$

2.6.2 Vector Norms

Let $X \in \mathbb{R}^n$ be a vector space. A real valued function $\|\cdot\| : \mathbb{R}^n \rightarrow \mathbb{R}$ is said to be norm of X if for any $\mathbf{x}, \mathbf{y} \in X$ and $\alpha \in \mathbb{R}$ it satisfies the following properties

- $\|\mathbf{x}\| \geq 0$ and $\|\mathbf{x}\| = 0$ if and only if $x = O$,
- $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$,
- $\|\alpha\mathbf{x}\| = |\alpha|\|\mathbf{x}\|$.

The vector p -norm of $\mathbf{x} \in \mathbb{C}$ is defined as

$$\|\mathbf{x}\|_p = \left(\sum_{i=1}^n |x_i|^p \right)^{\frac{1}{p}}; \quad 1 \leq p \leq \infty \quad (2.72)$$

In particular, when $p = 1, 2, \dots, \infty$, the norm can be defined as

$$\begin{aligned}\|\mathbf{x}\|_1 &= \sum_{i=1}^n |x_i|, \\ \|\mathbf{x}\|_2 &= \sqrt{\sum_{i=1}^n |x_i|^2} = \sqrt{\mathbf{x}^T \mathbf{x}}, \\ \|\mathbf{x}\|_\infty &= \max_{1 \leq i \leq n} |x_i|.\end{aligned}$$

2.6.3 H_2 -Norm

The H_2 norm is frequently referred to as the system's energy derived from the impulse input [66]. The H_2 norm is defined as for an asymptotically stable system

$$\|G\|_{H_2} = \sqrt{\frac{1}{2\pi} \int_{-\infty}^{\infty} \text{tr}(G(jw^*)G(jw))dw}, \quad (2.73)$$

where $G(jw)$ is the frequency response of an LTI system. This relation can be obtained from (2.83) by applying parseval's theorem.

Rewriting relation above equation we attained

$$\|G\|_{H_2} = \sqrt{\text{tr}(B^TQB)} = \sqrt{\text{tr}(CPC^T)}, \quad (2.74)$$

where Q is the solution of the observability Lyapunov equation

$$A^TQE + E^TQA + C^TC = 0, \quad (2.75)$$

and P as the solution of the controllability Lyapunov equation

$$APE^T + EPA^T + BB^T = 0. \quad (2.76)$$

2.6.4 H_∞ -Norm

The highest feasible output of a stable system may be measured by $|G(jw)|$ for a given sinusoidal input with a certain frequency w and unit magnitude. Over the whole range of frequencies of a single sinusoidal input, the H_∞ norm effectively

gives the highest feasible amplification of a stable system [67]. As a result, the H_∞ norm defined as follows

$$\|G\|_{H_\infty} = \sup_{w \in \mathbb{R}} \alpha_{max}(G(jw)), \quad (2.77)$$

with α_{max} denoting the maximum singular value of $G(jw)$. To compute the H_∞ norm we can follow the procedure given below.

Pick a set of frequencies $w_1, w_2, w_3, \dots, w_N$, then for $k = 1, 2, 3, \dots, N$ calculate $N_k = \alpha_{max}(G(jw_k))$. Here α_{max} expresses the maximum singular value, and λ_{max} represents the maximum eigenvalue. Estimate the H_∞ norm as

$$\|G\|_{H_\infty} = \max_{1 \leq k \leq N} N_k. \quad (2.78)$$

2.7 Error System

The disparity between the desired and actual value of a system output in the limit as time advances to infinity is known as state error (i.e., when the control system's response has reached steady-state). The state error of a linear system is a characteristic of the input-output response. A sound control system, in general, will have a low steady-state error. A variety of circumstances can cause control system errors. For example, during the transient phase, changes in the reference input will produce an inevitable error and may also induce steady-state error. At a steady state, mistakes are caused by imperfections in the system components such as static friction, backlash (a sudden strong response or reaction), amplifier drift, and aging or degradation. H_2 -norm of the error system Now, by preserving the form (2.1.1), the error system connected with the ROM (2.13) of the subjected system (2.5) has the condition.

$$G_{err} = G(s) - \hat{G}(s) - C_{err}(sE_{err} - A_{err})^{-1}B_{err} \quad (2.79)$$

where the transfer functions $G(s)$ and $\hat{G}(s)$ are respectively coupled to systems (2.5) and (2.13). The designed error system (2.79) are formed with the matrices

$$E_{err} = \begin{bmatrix} E & 0 \\ 0 & \hat{E} \end{bmatrix}, A_{err} = \begin{bmatrix} A & 0 \\ 0 & \hat{A} \end{bmatrix}, B_{err} = \begin{bmatrix} B \\ \hat{B} \end{bmatrix}, C_{err} = \begin{bmatrix} C & -\hat{C} \end{bmatrix}, \quad (2.80)$$

The controllability and observability Lyapunov equations corresponding to the Gramians P_{err} and Q_{err} , respectively, connecting to the error system (2.79) are

$$\begin{aligned} A_{err}P_{err}E_{err}^T + E_{err}P_{err}A_{err}^T + B_{err}B_{err}^T &= 0, \\ A_{err}^TQ_{err}E_{err} + E_{err}^TQ_{err}A_{err} + C_{err}^TC_{err} &= 0. \end{aligned} \quad (2.81)$$

In the previous work an efficient way to estimate the H_2 -norm for the error system (2.79) is introduced as

$$\begin{aligned} \|G\|_{H_2} &= \sqrt{\text{trace}(B_{err}^TQ_{err}B_{err})} \\ &= \sqrt{\|G(s)\|_{H_2}^2 + \|G(s)\|_{H_2}^2 + 2\text{trace}(B^TQ_s\hat{B})} \end{aligned} \quad (2.82)$$

Here, $\|G(s)\|_{H_2}$ is the H_2 norm of the full model which must to assess at once in the total computing, but this is not possible by any direct solvers due to the system structure of large-scale systems. Suppose Z_q is the low-rank Gramian factor that can be attained by rearranging the LRCF-ADI technique as provided in [68], resulting in $Q = Z_qZ_q^T$ then $\|G(s)\|_{H_2}$ can be written as

$$\|G(s)\|_{H_2}^2 = \text{trace}(B^TQB) = \text{trace}((B^TZ_q)(B^TZ_q)^T). \quad (2.83)$$

Again, $\|\hat{G}(s)\|_{H_2}$ is the H_2 -norm of the ROM (2.13), can be enumerated by the Gramian \hat{Q} of the low-rank Lyapunov equation

$$\hat{A}^T\hat{Q}\hat{E} + \hat{E}^T\hat{Q}\hat{A} + \hat{C}^T\hat{C} = 0. \quad (2.84)$$

that consists of reduced-order matrices. Due to the small size of these matrices, the following Lyapunov equation is solvable by the MATLAB library command `lyap`.

Finally, $\text{trace}(B^TQ_s\hat{B})$ can be measured by the low-rank Gramian Q_s of the sparse-dense Sylvester equation

$$A^TQ_s\hat{E} + \hat{E}^TQ_s\hat{A} + C^T\hat{C} = 0. \quad (2.85)$$

that can be efficiently solved by a modified form of the techniques presented in Algorithm 1.

2.8 Shift Parameters

The numerical weapon for compensating for system disturbances is shift parameters. They are the random constants that have been pre-conditioned and are system-oriented. Adjustable shift selection is critical for the simulation process's rapid and smooth convergence. Traditional Penzl's heuristic shifts and Wachspress's optimum shifts are widely employed for large-scale descriptor systems. For advanced and more significant descriptor systems, the adaptive ADI shift [69] selection method is now used. The ADI min-max problem is the most crucial instrument for producing shift parameters, as stated by.

$$\min_{\mu_1, \dots, \mu_j \in \mathbb{C}^-} \left(\max_{1 \leq l \leq n} \left| \prod_{i=1}^J \frac{\overline{\mu_i} - \lambda_l}{\mu_i + \lambda_l} \right| \right); \quad \lambda_l \in \Lambda(A, E), \quad (2.86)$$

where $\Lambda(A, E)$ represents the spectrum of the matrix pencil $\lambda E - A$. We'll go through some of the methods for locating relevant shift parameters in this section.

2.8.1 Adaptive ADI Shifts

The eigenvalues of the matrix pencil $\lambda E - A$ projected to the span of C^T are initially examined in the situation of adaptive shifts, where E , A and C are sparse and of acceptable dimensions. Once all of the shifts in the set have been utilized, the pencil must be projected throughout the current basis V_i with the current eigenvalues serving as the next set of shifts. The procedure had to be recursive, and at each step, the subspace to all the bases V_i produced with the current setting of shifts had to be extended [70]. It should be noted that the presence of all predicted eigenvalues in \mathbb{C}^- cannot be guaranteed. Consider W to be the extended subspace's orthogonal basis, and calculate the eigenvalues of $(\lambda W^T E W - W^T A W)$. By solving the min-max issue (2.86) in a similar way to the heuristic procedure, select several optimum shifts $\{\mu_i\}_{i=1}^J$. The iterative process will be repeated until the algorithm reaches the specified tolerance.

2.9 Existing data models

Generating real-world data from the physical model one of the toughest part of the branches of system and control theory. In general, the physical models are sparse and of the types of descriptor systems. In this thesis, In-compressible Navier-Stokes model is mainly adapted for the validation of the proposed techniques.

2.9.1 Finite element method

A mathematical model is created to characterize an engineering system to study it. Some assumptions are used for simplification while building the mathematical model. Finally, the system's behavior is described using the controlling mathematical statement. Differential equations with provided conditions are commonly used in mathematical expressions. The solutions to these differential equations that explain the behavior of the given engineering system are frequently quite challenging to come by. Solving such differential equations has become achievable with the introduction of high-performance computers. To discover approximate solutions to various engineering issues, many numerical solution approaches have been developed and implemented [71]. The finite element approach, in particular, has become a critical numerical solution technique. One of the most significant advantages of the finite element technique is the ease with which general-purpose computer software can be created to examine various issues. The limited element approach, in particular, can handle any complicated shape of the problem domain with defined constraints. The finite element approach divides the problem domain into numerous sub-domains, each of which is referred to as a finite element. As a result, the problem domain comprises a lot of finite element patches.

2.9.2 Navier–Stokes Model

To discuss diverse fluid dynamics issues and engineering applications. Large-scale structural index-2 descriptive systems are produced by spatial dissociation of such equations utilizing finite differences or finite material approaches. Because of the computational complexity and storage needs, controller design, simulation, and design optimization work with these large-scale systems can be difficult. Because

the Navier-Stokes equation is parabolic, it has more excellent analytical characteristics but a less mathematical structure (e.g., they are never fully integrated). Because they represent the mathematics of numerous phenomena of scientific and engineering interest, the Navier-Stokes equations are effective. Weather, ocean currents, and airflow over a wing may all be modeled using these. In their full and simplified version, the Navier-Stokes equations aid in the design of airplanes and vehicles, the research of blood flow, the construction of power plants, the analysis of pollution, and much more. They are may be used to simulate and investigate magneto-hydrodynamics when combined with Maxwell's equations.

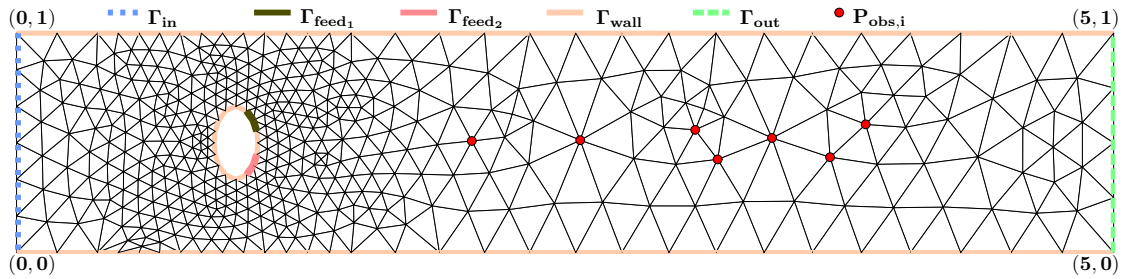


Figure. 2.4. Initial discretization of the von Kármán vortex street with coordinates, boundary parts and observation points

According to the linearization principle, a generic nonlinear model can be stabilized by a linear quadratic regulator (LQR) for linearization of itself in the region of the linearization point. The underlying assumption is that if the regulator functions effectively, the area where linearization is a good approximation of the nonlinear system is never left. The authors used this idea to create a Navier-Stokes model for the von Kármán vortex street in . The linearized Navier-Stokes equations that developed there, and which we analyze in this study, are as follows

$$\begin{aligned} \frac{\partial \mathbf{v}}{\partial t} - \frac{1}{Re} \nabla^2 \mathbf{v} + (\mathbf{w} \cdot \nabla) \mathbf{v} + (\mathbf{v} \cdot \nabla) \mathbf{w} + \nabla p &= 0, \\ \nabla \cdot \mathbf{v} &= 0, \end{aligned} \tag{2.87}$$

where, \mathbf{v} , \mathbf{w} signify velocity vectors, p denotes pressure, and Re indicates Reynolds number. The vector \vec{w} represents the stationary solution of the in-compressible nonlinear Navier-Stokes equations. The divergence of the initial state from the fixed solution is represented by the vector \mathbf{v} gives the boundary and beginning conditions and the model's derivation. A mixed finite element approach based on the well-known Taylor-Hood finite elements is used to find desired LTI systems

[72]. The algebraic differential equations are the result of this

$$\begin{aligned} E_1 \cdot \mathbf{v}(\mathbf{t}) &= A_1 \mathbf{v}(\mathbf{t}) + A_2 p(t) + B_1 \mathbf{u}(\mathbf{t}), \\ A_2^T \mathbf{v}(\mathbf{t}) &= 0 \end{aligned} \tag{2.88}$$

where, $\mathbf{v}(\mathbf{t}) \in \mathbb{R}^{n_1}$ is the nodal vector of discretized velocity deviations, $p(t) \in \mathbb{R}^{n_2}$ is the discretized pressure, $\mathbf{u}(\mathbf{t}) \in \mathbb{R}^m$ are the inputs, and $E_1, A_1 \in \mathbb{R}^{n_1 \times n_1}$ $A_2 \in \mathbb{R}^{n_1 \times n_2}$, $B_1 \in \mathbb{R}^{n_1 \times m}$ are all sparse matrices.

Additionally, the vertical velocities in the observation nodes depicted in Figure 2.4 in the domain are modeled by the output equation

$$\mathbf{y}(\mathbf{t}) = C_1 \mathbf{v}(\mathbf{t}) \tag{2.89}$$

with the output $y(t) \in \mathbb{R}^p$ and the output matrix $C_1 \in \mathbb{R}^{p \times n_1}$

2.9.3 Stokes Model

Let us consider a measure approaching thorough numerical processing of the Le ray projection, which is the key to proper implementations of optimal control for flow problems. We consider a symmetric and linear approaches for stationary, in-compressible flow problems, i.e., the Stokes equations

$$\left. \begin{aligned} \frac{\partial v(t, x)}{\partial t} - v \Delta v(t, x) + \nabla p(t, x) &= 0 \\ \nabla \cdot \Delta v(t, x) &= 0 \end{aligned} \right\} \text{On } (0, \infty) \times \Omega \tag{2.90}$$

including the time $t(0, \infty)$, these spatial variable $x \in \Omega$, the velocity field $v(t, x) \in \mathbb{R}^2$, the pressure $p(t, x) \in \mathbb{R}$, and some viscosity $v \in \mathbb{R}^+$. Additionally, we become $\Omega \subset \mathbb{R}^2$ as a bounded and relevant domain with boundary $= \Delta\Omega$, some Dirichlet boundary conditions, which define an inflow outflow problem. and suitable initial conditions.

We implement the system of lines to the Stokes equations, as is usual in in stationary control problems. which involves we discretize (2.90) in space using a mixed

finite element method. and get the following system of differential-algebraic equations

$$\begin{aligned} M \frac{\partial z(t)}{\partial t} &= Az(t) + Gp(t) + f(t) \\ G^T z(t) &= 0 \end{aligned} \tag{2.91}$$

with the discretized velocity $z(t) \in \mathbb{R}^{n_v}$, and pressure $p(t) \in \mathbb{R}^{n_p}$ the symmetric positive definite mass matrix $M = M^T \succ 0 \in \mathbb{R}^{n_v \times n_v}$ and the symmetric negative definite system matrix $A = A^T \prec 0 \in \mathbb{R}^{n_v \times n_v}$.

Since, in general, one can recognize the velocity only in members of the domain, we compute the output equation

$$y(t) = Cz(t) \tag{2.92}$$

The output $y(t) \in \mathbb{R}^{n_a}$ and the output operator $C \in \mathbb{R}^{n_a \times n_b}$ that picks the part of the domain where we require to measure the velocity, which is a part of the outflow boundary.

There are several special forms of the Navier-Stokes models. Among them some are very important for the real-life engineering purposes, namely, Oseen model and Stokes model.

2.9.4 Oseen Model

Oseen model is a particular type of data model directed by the Stokes equation [73] written for incompressible fluid flow as

$$\begin{aligned} \frac{\partial \mathbf{v}}{\partial t} &= \Delta \mathbf{v} - \nabla p + f \\ \Delta \mathbf{v} &= 0 \end{aligned} \tag{2.93}$$

with initial and boundary situations

$$\begin{aligned} \mathbf{v}(\mathbf{x}, \mathbf{0}) &= v_0(x), x \in \Omega \\ \mathbf{v}(\mathbf{x}, \mathbf{t}) &= g(x, t), (x, t) \in \partial\Omega \times (0, t_f) \end{aligned} \tag{2.94}$$

where, \mathbf{v} is the velocity vector, p is the fluid pressure, f is the external force, Ω is a bounded open domain with boundary $\partial\Omega$ and t_f is the final time interval.

This model is related to the simple channel flow of an in-compressible fluid like water, whose meshing structure can be represented by Figure 2.5. Figure 2.6 and Figure 2.7 sequentially represent velocity and pressure profiles of simple channel flow must initial velocity and pressure as zero with no-slip boundary condition.

We get a specific sort of descriptor system of the form (2.9) after extracting the data model of the oseen equation from the physical model, which is known as the index-II descriptor system. After completing a comprehensive investigation utilizing the data models given above, the analytical outcomes of this thesis were identified.

Table. 2.1. Dimension of Oseen model

Model size	Differential variables	Algebraic variables	Inputs	Outputs
2804	1904	900	4	4

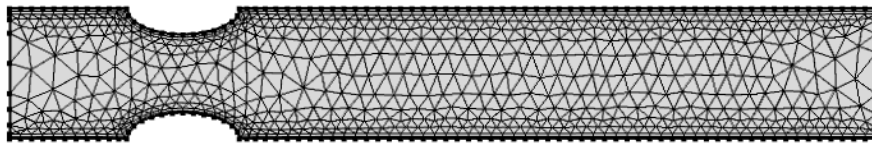


Figure. 2.5. Mesh structure of a pipe flow

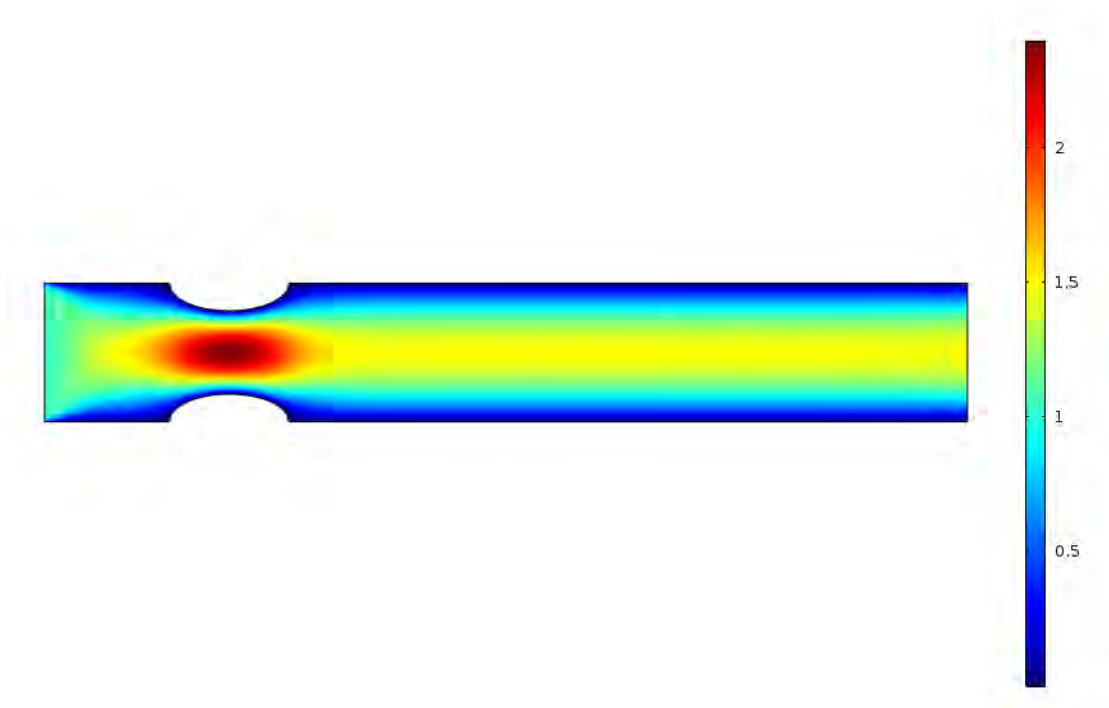


Figure. 2.6. Velocity shape of a pipe flow

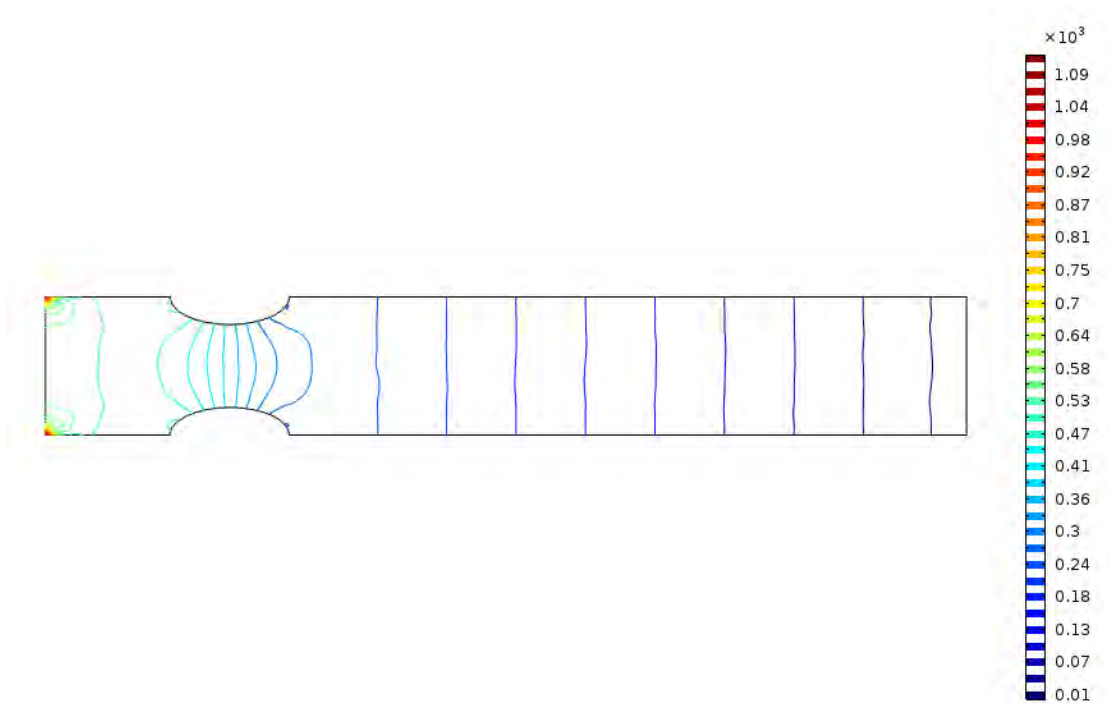


Figure. 2.7. Pressure shape of a pipe flow

Chapter 3

Two Sided Iterative Algorithm for the CARE Arising from Index-2 Descriptor System

The main outcome of this thesis is included in this chapter. Furthermore, conversion of index-2 systems to generalized systems, the sparsity-preserving structure of Krylov subspace for index-2 systems, the Two-sided iterative algorithm for index-2 systems, and the particular configuration of the H_2 -norm estimation are discussed in this chapter in detail.

3.1 Krylov subspace for index-2 descriptor system

In general Krylov subspace is defined and structured for the generalized systems and because of that an initial conversion from index-2 systems to generalized systems is required. But converted generalized systems will be dense and structure destroying. So, explicit converted form is prohibited. Thus, we will not imply the converted generalized systems in practice, instead we will come back to the sparse form of the index-2 systems and accordingly construct the Krylov subspace as the combination of linear systems in sparse forms.

3.1.1 Structure of the incompressible Navier-Stokes model

Incompressible Navier-Stokes flow is one of the most impressive content of the interest of the researchers. It has prominent impact in fluid mechanics, naval engineering, oceanography, and water resource scientists. The the analysis of fluid properties and attributes of undersea atmospheres, incompressible Navier-Stokes flow is one of the prominent issue for the naval architects. It is an essential part of the investigation of the sea-route analysts.

Details of the Navier-Stokes equations with the discretization can be found in [4]. Linearizing the Navier-Stokes equations in space and time variable by mixed finite element method without altering the time variable converts them to the linear time-invariant systems . The incompressible Navier-Stokes flow can be written as the following differential-algebraic equations

$$\begin{aligned} M\dot{v}(t) &= A_1v(t) + A_2p(t) + B_1u(t), \\ 0 &= A_3v(t) + B_2u(t), \\ y(t) &= C_1v(t) + C_2p(t) + Du(t), \end{aligned} \tag{3.1}$$

where $v(t) \in \mathbb{R}^{n_v}$ is the nodal vector of the discretized velocity with $v(0) = v_0$, $p(t) \in \mathbb{R}^{n_p}$ is the discretized pressure, and $u(t) \in \mathbb{R}^{n_u}$ is the input with the output $y(t) \in \mathbb{R}^{n_y}$. The symmetric positive definite matrix $M \in \mathbb{R}^{n_v \times n_v}$ represents the mass, whereas the matrices $A_1 \in \mathbb{R}^{n_v \times n_v}$, $A_2 \in \mathbb{R}^{n_v \times n_p}$, and $A_3 \in \mathbb{R}^{n_p \times n_v}$ are for system components including discretized gradient. $C_1 \in \mathbb{R}^{n_y \times n_v}$ and $C_2 \in \mathbb{R}^{n_y \times n_p}$ are the output matrices, which quantify the velocity pattern at the inner nodes. $D \in \mathbb{R}^{n_y \times n_u}$ is the direct transform matrix [74]. Considering M is invertible, A_2 and A_3 have both full column rank and $A_3M^{-1}A_2$ is non-singular. Hence the system (3.1) is of the index-2 differential-algebraic system having the dimension n_v+n_p [75]. A system equivalent to the system (3.1) can be written in the following form

$$\begin{aligned} \bar{E}\dot{x}(t) &= \bar{A}x(t) + \bar{B}u(t), \\ y(t) &= \bar{C}x(t) + \bar{D}u(t), \end{aligned} \tag{3.2}$$

where $x(t) = \begin{bmatrix} v(t) \\ p(t) \end{bmatrix}$, $\bar{E} = \begin{bmatrix} M & 0 \\ 0 & 0 \end{bmatrix}$, $\bar{A} = \begin{bmatrix} A_1 & A_2 \\ A_3 & 0 \end{bmatrix}$, $\bar{B} = \begin{bmatrix} B_1 & B_2 \end{bmatrix}$, $\bar{C} = \begin{bmatrix} C_1 \\ C_2 \end{bmatrix}$, and $\bar{D} = D$ with appropriate dimensions. The matrix pencil corresponding to the system is defined as (\bar{A}, \bar{E}) , where the matrix \bar{E} is singular and the matrix pencil

(\bar{A}, \bar{E}) has $n_v - n_p$ number of non-zero finite complex eigenvalues and $2n_p$ infinite eigenvalues.

Reynolds number (Re) is one of the key features that instigate the system characteristics. For $Re \geq 300$, the Navier-Stokes flow turns into unstable and analysis of the flow attributes becomes troublesome [76]. Optimal feedback stabilization of the Navier-Stokes flow is indispensable and Riccati-based boundary feedback stabilization is the best apparatus in this situation. Riccati equation corresponding to system (3.2) is of the form

$$\bar{A}^T X \bar{E} + \bar{E}^T X \bar{A} - \bar{E}^T X \bar{B} \bar{B}^T X \bar{E} + \bar{C}^T \bar{C} = 0. \quad (3.3)$$

The solution X of the Riccati equation (3.3) is symmetric positive definite and called stabilizing for a stable closed-loop matrix $\bar{A} - (\bar{B} \bar{B}^T) X \bar{E}$. The optimal feedback matrix K^o can be computed as $K^o = \bar{B}^T X \bar{E}$ and can be implemented for optimally stabilized the target system by replacing \bar{A} by $\bar{A}_s = \bar{A} - \bar{B} K^o$ by $\bar{A}_s = \bar{A} - \bar{B} K^o$ [77]. The stabilized system can be written as

$$\begin{aligned} \bar{E} \dot{x}(t) &= \bar{A}_s x(t) + \bar{B} u(t), \\ y(t) &= \bar{C} x(t) + \bar{D} u(t). \end{aligned} \quad (3.4)$$

With the increasing number of flow components and finer meshes in the discretization process, dimensions of the matrices in the system (3.1) become very large. Due to the size of the system matrices, Riccati-based boundary feedback stabilization of the system (3.1) is infeasible as the associated Riccati equation cannot not be solved by the usual matrix equation solvers [78]. To overcome this adversity, a suitable Model-Order Reduction (MOR) technique needs to be applied. A computationally convenient Reduced-Order Model (ROM) of the system (3.2) can be written as

$$\begin{aligned} \hat{E} \dot{\hat{x}}(t) &= \hat{A} \hat{x}(t) + \hat{B} \hat{u}(t), \\ \hat{y}(t) &= \hat{C} \hat{x}(t) + \hat{D} \hat{u}(t), \end{aligned} \quad (3.5)$$

where $\hat{E} = W^T \bar{E} V$, $\hat{A} = W^T \bar{A} V$, $\hat{B} = W^T \bar{B}$, $\hat{C} = \bar{C} V$, $\hat{D} = \bar{D}$. The projector V and W can be found from any compatible MOR technique. Using the reduced-order matrices given in (3.5), the reduced-order form of Riccati equation (3.3) can be attained as

$$\hat{A}^T \hat{X} \hat{E} + \hat{E} \hat{X} \hat{A} - \hat{E} \hat{X} \hat{B} \hat{B}^T \hat{X} \hat{E} + \hat{C}^T \hat{C} = 0. \quad (3.6)$$

Now, the optimal feedback matrix for the system (3.2) can be approximated by applying the ROM (3.5). The reduced-order Riccati equation (3.6) is to solved for symmetric positive-definite matrix \hat{X} by the MATLAB library command `care`. Then, the stabilizing feedback matrix $\hat{K} = \hat{B}^T \hat{X} \hat{E}$ corresponding to the ROM (3.5) can be estimated, and hence the approximated optimal feedback matrix $K^o = \hat{K} V^T \bar{E}$ [79]. Finally, utilizing K^o stabilization of the system (3.2) can be done as the system (3.4).

3.1.2 Conversion of index-2 descriptor system to generalized system

Let us Assume M is non-symmetric positive definite and $A_2^T \neq A_3$. Initially, we consider $B_2 = 0$. According to the Section 3 of [80], from the algebraic part of system (3.1), $p(t)$ can be expressed as $v(t)$. Then by proper elimination and substitution, system (3.1) can be converted to an equivalent form

$$\begin{aligned} \mathcal{E}\dot{x}(t) &= \mathcal{A}x(t) + \mathcal{B}u(t), \\ y(t) &= \mathcal{C}x(t) + \mathcal{D}u(t), \end{aligned} \quad (3.7)$$

where the converted matrices are structured as

$$\begin{aligned} x &:= x_1, \quad \mathcal{E} := \Pi_l M \Pi_r, \quad \mathcal{A} := \Pi_l A_1 \Pi_r, \quad \mathcal{B} := \Pi_l B_1, \\ \mathcal{C} &:= (C_1 - C_2(A_3 M^{-1} A_2)^{-1} A_3 M^{-1} A_1) \Pi_r, \\ \mathcal{D} &:= D - C_2(A_3 M^{-1} A_2)^{-1} A_3 M^{-1} B_1, \end{aligned} \quad (3.8)$$

with $\Pi_l^T v(0) = \Pi_l^T v_0$. The projectors Π_l and Π_r are composed as

$$\begin{aligned} \Pi_l &:= I - M^{-1} A_2 (A_3 M^{-1} A_2)^{-1} A_3, \\ \Pi_r &:= I - A_2 (A_3 M^{-1} A_2)^{-1} A_3 M^{-1}. \end{aligned} \quad (3.9)$$

For the general case, we assume $B_2 \neq 0$. According to the Section 6 of [36], we consider the velocity $v(t)$ can be decomposed as

$$v(t) = v_0(t) + v_g(t), \quad (3.10)$$

where $v_0(t)$ satisfies $A_3 v_0(t) = 0$ and $v_g(t) = M^{-1} A_2 (A_3 M^{-1} A_2)^{-1} B_2 u(t)$.

Thus, the required converted system can be formed as

$$\begin{aligned} x &:= x_1, & \mathcal{E}\dot{x}(t) &= \mathcal{A}x(t) + \mathcal{B}u(t), \\ y(t) &= \mathcal{C}x(t) + \mathcal{D}u(t) - C_2(A_3M^{-1}A_2)^{-1}B_2\dot{u}(t), \end{aligned} \quad (3.11)$$

with $\Pi_l^T v(0) = \Pi_l^T (v_0 - v_g(0))$ and the converted matrices as

$$\begin{aligned} x &:= x_1, & \mathcal{E} &:= \Pi_l M \Pi_r, & \mathcal{A} &:= \Pi_l A_1 \Pi_r, \\ \mathcal{B} &:= \Pi_l (B_1 - A_1 M^{-1} A_2 (A_3 M^{-1} A_2)^{-1} B_2), \\ \mathcal{C} &:= (C_1 - C_2 (A_3 M^{-1} A_2)^{-1} A_3 M^{-1} A_1) \Pi_r, \\ \mathcal{D} &:= D - C_2 (A_3 M^{-1} A_2)^{-1} A_3 M^{-1} B_1. \end{aligned} \quad (3.12)$$

In addition, if both $B_2 = 0$ and $C_2 = 0$, the converted system will be as the same as system (3.11) but converted matrices will have the form

$$\begin{aligned} x &:= x_1, & \mathcal{E} &:= \Pi_l M \Pi_r, & \mathcal{A} &:= \Pi_l A_1 \Pi_r, \\ \mathcal{B} &:= \Pi_l B_1, & \mathcal{C} &:= C_1, & \mathcal{D} &:= D. \end{aligned} \quad (3.13)$$

3.1.3 Sparsity-preserving Krylov subspace bases for IRKA

For the converted generalized system derived in the previous section, desired projection matrices can be formed as

$$\begin{aligned} V &= [(\alpha_1 \mathcal{E} - \mathcal{A})^{-1} \mathcal{B} b_1, \dots, (\alpha_r \mathcal{E} - \mathcal{A})^{-1} \mathcal{B} b_r], \\ W &= [(\alpha_1 \mathcal{E} - \mathcal{A})^{-T} \mathcal{C}^T c_1, \dots, (\alpha_r \mathcal{E} - \mathcal{A})^{-T} \mathcal{C}^T c_r]. \end{aligned} \quad (3.14)$$

Here the matrices utilized to form the projection matrices V and W are dense and it will exploit the feasibility of the computation. Since keeping the system structure invariant is one of the prime concerns, the dense form of the system matrices is contradictory to the aim of the work. As the remedy of this inconvenience we must not construct the bases of the Krylov subspace explicitly with those dense matrices. Instead, we will generate the bases of the Krylov subspace by solving the following linear systems

$$\begin{aligned} (\alpha_i \mathcal{E} - \mathcal{A})^{-1} \mathcal{B} b_i &= V_i, \\ (\alpha_i \mathcal{E} - \mathcal{A})^{-T} \mathcal{C}^T c_i &= W_i, \end{aligned} \quad (3.15)$$

into the sparse forms as

$$\begin{aligned} \begin{bmatrix} \alpha_i M - A_1 & -A_2 \\ -A_3 & 0 \end{bmatrix} \begin{bmatrix} V_i \\ \Lambda_v \end{bmatrix} &= \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} b_i, \\ \begin{bmatrix} \alpha_i M^T - A_1^T & -A_3^T \\ -A_2^T & 0 \end{bmatrix} \begin{bmatrix} W_i \\ \Lambda_w \end{bmatrix} &= \begin{bmatrix} C_1^T \\ C_2^T \end{bmatrix} c_i. \end{aligned} \quad (3.16)$$

where Λ_v and Λ_w are the truncated parts of the basis vectors. Accordingly, the ROM corresponding to system (3.1) or equivalent systems can be formed as

$$\begin{aligned} \tilde{\mathcal{E}}\dot{x}(t) &= \tilde{\mathcal{A}}x(t) + \tilde{\mathcal{B}}u(t), \\ y(t) &= \tilde{\mathcal{C}}x(t) + \tilde{\mathcal{D}}u(t), \end{aligned} \quad (3.17)$$

where the sparsity-preserving reduced-order matrices

$$\begin{aligned} \tilde{\mathcal{E}} &:= W^T M V, \quad \tilde{\mathcal{A}} := W^T A_1 V, \\ \tilde{\mathcal{B}} &:= W^T B_1 - (W^T A_1) M^{-1} A_2 (A_3 M^{-1} A_2)^{-1} B_2, \\ \tilde{\mathcal{C}} &:= C_1 V - C_2 (A_3 M^{-1} A_2)^{-1} A_3 M^{-1} (A_1 V), \\ \tilde{\mathcal{D}} &:= D - C_2 (A_3 M^{-1} A_2)^{-1} A_3 M^{-1} B_1. \end{aligned} \quad (3.18)$$

Details of the above formulation is derived in the Section 4.4.1 and Section 4.4.1 of [54]. Sparsity-preserving Iterative Rational Krylov Algorithm (IRKA) for the index-2 descriptor systems is summarized in Algorithm 6.

3.2 Two Sided Iterative Algorithm for index-2 descriptor systems

This section includes the main tasks of this work. We derive an improved version of the Two Sided Iterative Algorithm (TSIA) for index-2 descriptor systems for stabilize the incompressible Navier-Stokes flow.

Algorithm 6: Sparsity-preserving IRKA for index-2 descriptor systems.

Input : $M, A_1, A_2, A_3, B_1, B_2, C_1, C_2, D$.

Output: Optimal feedback matrix K^o .

- 1 Make the initial selection of the interpolation points $\{\alpha_i\}_{i=1}^r$ and the tangential directions $\{b_i\}_{i=1}^r$ and $\{c_i\}_{i=1}^r$.
 - 2 Utilizing the sparse linear systems in (3.16), construct the projection matrices as equation (3.14).
 - 3 **while** (*not converged*) **do**
 - 4 Compute the reduced-order matrices as (3.18).
 - 5 Compute $\tilde{A}z_i = \lambda_i \tilde{\mathcal{E}}z_i$ and $y_i^* \tilde{A} = \lambda_i y_i^* \tilde{\mathcal{E}}$ for $\alpha_i \leftarrow -\lambda_i$, $b_i^* \leftarrow -y_i^* \tilde{\mathcal{B}}$ and $c_i^* \leftarrow \tilde{\mathcal{C}}z_i^*$, for $i = 1, \dots, r$.
 - 6 Repeat step 2.
 - 7 Repeat step 4.
 - 8 Solve the Riccati equation (3.6) for \hat{X} .
 - 9 Compute $\hat{K} = \tilde{\mathcal{B}}^T \hat{X} \hat{E}$ and hence $K^o = \hat{K}V^T M$.
 - 10 **end while**
-

3.2.1 Formulation of the generalized sparse-dense Sylvester equation

Initially, we assume a makeshift ROM with the desired dimension in the form (3.17) employing any classical iterative method accomplishing a few iterations. But due to the use of minimum number of iterations, the attained ROM cannot efficiently approximate the original system. So, quality of the ROM needs to be improved by the further techniques such that it satisfies the Wilson conditions for \mathcal{H}_2 optimality [81].

Now, we are to construct two generalized sparse-dense Sylvester equations. To do this, we need to compute the following matrices

$$\begin{aligned}
 B_n &= \begin{bmatrix} M & A_2 \\ A_3 & 0 \end{bmatrix}^{-1} \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, \\
 C_n &= \begin{bmatrix} M^T & A_3^T \\ A_2^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} C_1^T \\ C_2^T \end{bmatrix}.
 \end{aligned} \tag{3.19}$$

Assuming $P_n = MB_n\tilde{\mathcal{B}}^T$ and $Q_n = -MC_n\tilde{\mathcal{C}}$ the required sparse-dense Sylvester equations can be formed as

$$\begin{aligned} AX\tilde{\mathcal{E}} + \mathcal{E}X\tilde{\mathcal{A}} + P_n &= 0, \\ AY\tilde{\mathcal{E}} + \mathcal{E}Y\tilde{\mathcal{A}} + Q_n &= 0. \end{aligned} \quad (3.20)$$

3.2.2 Solving generalized sparse-dense Sylvester equation

Sylvester equations formed in (3.20) can be solved by the techniques given in the Section 3.2 of [82]. Since the simulations of the Sylvester equations in (3.20) are analogous, we will derive the method for the computing X only. To generate the Krylov subspace QZ-decomposition of $\tilde{\mathcal{E}}$ and $\tilde{\mathcal{A}}$ is to be computed, such that $\tilde{\mathcal{E}} = QSZ^T$ and $\tilde{\mathcal{A}} = QTZ^T$, respectively. Here, S and T are the upper triangular matrices. As the techniques provided in the dense form, we have to re-write the basis vector in the sparse form as

$$\begin{bmatrix} S_{jj}A_1 + T_{jj}M^T & S_{jj}A_2 \\ S_{jj}A_3 & 0 \end{bmatrix} \tilde{X}_j = -\hat{P}_n^{(j)}, \quad (3.21)$$

and the successive columns for the solution matrix X can be generated as

$$\hat{P}_n^{(j)} = -\tilde{P}_n^{(j)} - \sum_{i=1}^{j-1} F_{ij}\tilde{X}_i, \quad (3.22)$$

where $\tilde{P}_n^{(j)} = P_n^{(j)}Z$ and $F_{ij} = \begin{bmatrix} S_{ij}A_1 + T_{ij}M^T & S_{ij}A_2 \\ S_{ij}A_3 & 0 \end{bmatrix}$. The updated sparse form of the desired techniques are provided in Algorithm 7.

3.2.3 Two Sided Iterative Algorithm to estimate the optimal feedback matrix

At first, an iterative method with minimum iterations needs to be utilized to find initial makeshift ROM with desired dimension. With the required matrix-vector algebraic operations Sylvester equations defined in (3.20) need to be solved for X and Y , respectively. Then the approximated projector matrices \tilde{V} and \tilde{W} can be computed by the QR-decomposition of the matrices X and Y , respectively. Those

Algorithm 7: Solution of generalized sparse-dense Sylvester equation.

Input : $M, A_1, A_2, A_3, B_1, B_2, \tilde{\mathcal{E}}, \tilde{\mathcal{A}}, \tilde{\mathcal{B}}, j_{\max}$ (number of iterations).

Output: Approximate solution X of the Sylvester equation (3.20).

- 1 Compute the QZ-decomposition $\tilde{\mathcal{E}} = QSZ^T$ and $\tilde{\mathcal{A}} = QTZ^T$.
 - 2 Construct B_n according to equation (3.19) and compute $P_n = MB_n\tilde{\mathcal{B}}^T$.
 - 3 Compute $\tilde{P}_n = P_nZ$.
 - 4 **while** $j \leq j_{\max}$ **do**
 - 5 Compute $\hat{P}_n^{(j)}$ by equation (3.22).
 - 6 Solve the linear system (3.21) for \tilde{X}_j .
 - 7 **end while**
 - 8 Compute $X = \tilde{X}Q^T$.
-

Algorithm 8: TSIA for the optimal feedback matrix of index-2 descriptor systems.

Input : $M, A_1, A_2, A_3, B_1, B_2, C_1, C_2, \tilde{\mathcal{E}}, \tilde{\mathcal{A}}, \tilde{\mathcal{B}}, \tilde{\mathcal{C}}$.

Output: $\tilde{\mathcal{E}}, \tilde{\mathcal{A}}, \tilde{\mathcal{B}}, \tilde{\mathcal{C}}$ satisfying the Wilson conditions.

- 1 Construct the Sylvester equations defined in (3.20) and solve for X and Y , respectively.
 - 2 Compute the QR decomposition $X = \tilde{V}\beta_v$ and $Y = \tilde{W}\beta_w$.
 - 3 Compute $V = \tilde{V}$ and $W = \tilde{W}^{-1}(\tilde{V}^T M^T \tilde{W})$.
 - 4 Construct Wilson conditions satisfying reduced-order matrices defined in (3.18).
 - 5 Form the reduced-order Riccati equation (3.23) and solve for \tilde{X} .
 - 6 Approximate the optimal feedback matrix from equation (3.24).
-

crude projector matrices need to be refined as $V = \tilde{V}$ and $W = \tilde{W}(\tilde{V}^T M^T \tilde{W})^{-1}$, respectively. Implementing V and W the ROM (3.17) can be acquired with the reduced-order matrices defined in (3.18) and hence the reduced-order Riccati equation can be formed as

$$\tilde{A}^T \tilde{X} \tilde{E} + \tilde{E}^T \tilde{X} \tilde{A} - \tilde{E}^T \tilde{X} \tilde{B} \tilde{B}^T \tilde{X} \tilde{E} + \tilde{C}^T \tilde{C} = 0, \quad (3.23)$$

which can be solved for \tilde{X} by MATLAB `care` command and then reduced-order feedback matrix can be estimated as $\tilde{K} = \tilde{B}^T \tilde{X} \tilde{E}$. Finally, the optimal feedback matrix for the system (3.1) can be approximated as

$$K^o = (\tilde{B}^T \tilde{X} \tilde{E})V^T \tilde{E} = \tilde{K}V^T M. \quad (3.24)$$

The whole process is summed up in Algorithm 8.

3.2.4 Stabilization of index-2 descriptor system

Plugging the optimal feedback matrix K^o , the unstable incompressible Navier-Stokes flow given by (3.1) can be optimally stabilized as

$$\begin{aligned} M\dot{v}(t) &= (A_1 - B_1K^o)v(t) + A_2p(t) + B_1u(t), \\ 0 &= A_3v(t) + B_2u(t), \\ y(t) &= C_1z(t) + C_2p(t) + Du(t). \end{aligned} \quad (3.25)$$

3.2.5 \mathcal{H}_2 - norm of the error system

Let us consider $G(s)$ and $\tilde{G}(s)$ are the transfer functions of the converted generalized system (3.11) and (3.17), respectively. Then the associated error system can be formatted as

$$G_{err} = G(s) - \tilde{G}(s) = \mathcal{C}_{err}(s\mathcal{E}_{err} - \mathcal{A}_{err})^{-1}\mathcal{B}_{err}, \quad (3.26)$$

where we have taken the reduced-order matrices of (3.18) into account and the system matrices are formed as

$$\mathcal{E}_{err} = \begin{bmatrix} \mathcal{E} & 0 \\ 0 & \tilde{\mathcal{E}} \end{bmatrix}, \quad \mathcal{A}_{err} = \begin{bmatrix} \mathcal{A} & 0 \\ 0 & \tilde{\mathcal{A}} \end{bmatrix}, \quad \mathcal{B}_{err} = \begin{bmatrix} \mathcal{B} \\ \tilde{\mathcal{B}} \end{bmatrix}, \quad \text{and } \mathcal{C}_{err} = \begin{bmatrix} \mathcal{C} & -\tilde{\mathcal{C}} \end{bmatrix}. \quad (3.27)$$

Rahman *et al.* [83] derived the way of estimating \mathcal{H}_2 norm of the error system (3.26) as

$$\|G_{err}\|_{\mathcal{H}_2} = \sqrt{\|G(s)\|_{\mathcal{H}_2}^2 + \|\tilde{G}(s)\|_{\mathcal{H}_2}^2 + 2\text{trace}(\mathcal{B}^T\mathcal{Q}_s\tilde{\mathcal{B}})}. \quad (3.28)$$

Here, \mathcal{H}_2 norm of the full model is defined as $\|G(s)\|_{\mathcal{H}_2}$ and needs to be evaluate for once, which is inconvenient for the conventional simulation solvers. Assuming Z_q is the low-rank Gramian factor of the Gramian \mathcal{Q} that needs to be determined. In [9], a practically feasible technique is derived to overcome this situation. Then $\|G(s)\|_{\mathcal{H}_2}$ can be written as

$$\|G(s)\|_{\mathcal{H}_2}^2 = \text{trace}((B_1^T Z_q)(B_1^T Z_q)^T + (B_2^T Z_q)(B_2^T Z_q)^T). \quad (3.29)$$

Again, $\|\tilde{G}(s)\|_{\mathcal{H}_2}$ is the \mathcal{H}_2 norm of the ROM that can be evaluated by the Gramian \hat{Q} of the low-rank Lyapunov equation

$$\tilde{\mathcal{A}}^T \tilde{Q} \tilde{\mathcal{E}} + \tilde{\mathcal{E}}^T \tilde{Q} \tilde{\mathcal{A}} + \tilde{\mathcal{C}}^T \tilde{\mathcal{C}} = 0, \quad (3.30)$$

due to involvement of the reduced-order matrices, the Lyapunov equation (3.30) is solvable by the MATLAB library command `lyap`.

Finally, $\text{trace}(\mathcal{B}^T \mathcal{Q}_s \tilde{\mathcal{B}})$ can be computed by the Gramian \mathcal{Q}_s of the sparse-dense Sylvester equation

$$\mathcal{A}^T \mathcal{Q}_s \tilde{\mathcal{E}} + \tilde{\mathcal{E}}^T \mathcal{Q}_s \tilde{\mathcal{A}} + \mathcal{C}^T \tilde{\mathcal{C}} = 0, \quad (3.31)$$

that can be conveniently solved by reforming the techniques provided in Algorithm 7.

Chapter 4

Numerical Result

In this chapter, justification of the adaptability and efficiency of the proposed techniques will be discussed. Computations are done through the MATLAB simulations to observe the betterment of the proposed techniques in comparison to the present methods. The investigation involves both the graphical and tabular approaches. Computation time and accuracy of the approximation, and robustness of the stabilization of the transient behaviors will be the prime concern of this discussion. we have implemented the proposed techniques to the real-world Navier-Stokes models of unstable type.

All the results have been achieved using the MATLAB 8.5.0 (R2015a) on a Windows machine having Intel-Xeon Silver 4114 CPU 2.20 GHz clock speed, 2 cores each, and 64 GB of total RAM.

4.1 Model description

In this work, we have considered some unstable Navier-Stokes models with Reynolds number $Re = 300, 400, 500$, respectively. Dimension of the models varies from 1 to 5. Each of the models are non-symmetric having 2 inputs with 7 outputs. For the target models, in the conventional first-order index-2 descriptor system the sub-matrices B_2 and C_2 both are the sparse zero matrices. The full specifications of the target models are given in Table [4.1](#).

Table. 4.1. Structure of the target Navier-Stokes models

Dimension	Reynolds Number (Re)	States	Algebraic variables	Input	Output
1	300	3142	453	2	7
	400				
	500				
2	300	8568	1123		
	400				
	500				
3	300	19770	2615		
	400				
	500				
4	300	44744	5783		
	400				
	500				
5	300	98054	12566		
	400				
	500				

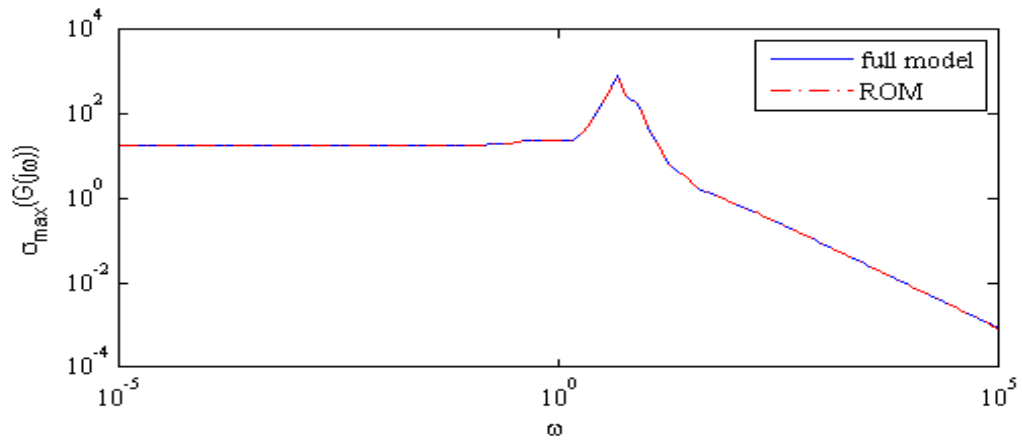
4.2 Approximation of the full models with the reduced-order models

The accuracy of the approximation of the full models with the ROMs will be validated here. We will verify the level of approximation graphically and then evaluate the \mathcal{H}_2 -norm of the error systems for the objective models.

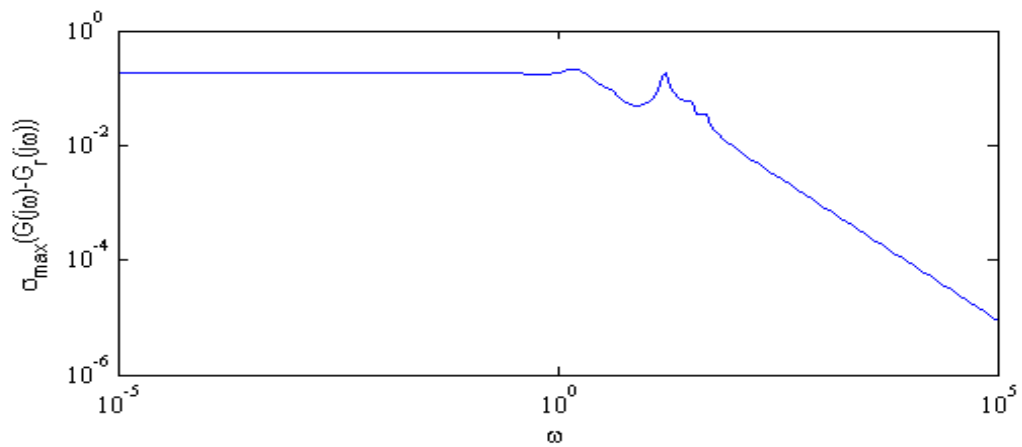
4.2.1 Comparison of the transfer functions

A graphical comparison between the full models and ROMs will be depicted. Since the Navier-Stokes models of various dimensions have the same system structure and transitional properties, in the graphical analysis we will show only the properties of the 3-dimensional model.

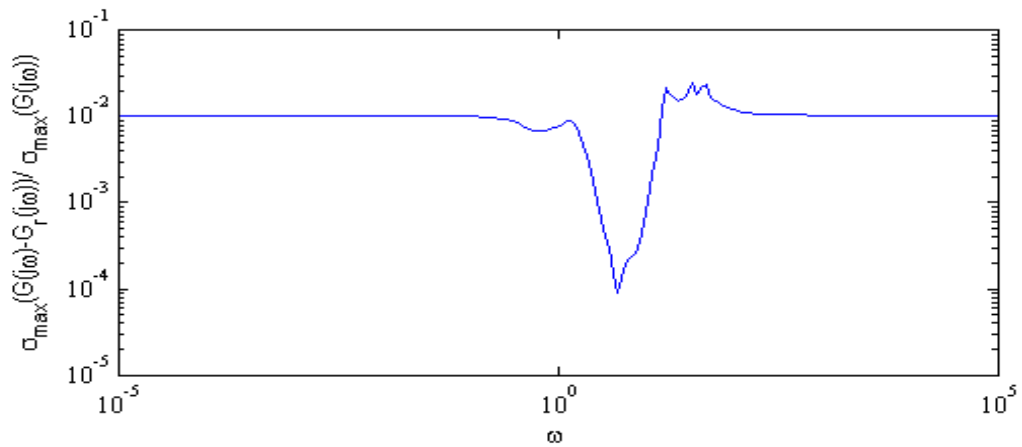
In Figure 4.1, the sub-figure Figure 4.1a shows the comparison of the transfer function (sigma plot) of the full model and corresponding ROM, whereas sub-figures Figure 4.1b and Figure 4.1c evince the absolute and relative errors, respectively, of this reduced-order approximation. From the above-mentioned figures, it can be evident that the by the proposed techniques the objective full models can be properly approximated by the ROMs with a reasonable level of accuracy.



(a) Sigma plot



(b) Absolute error



(c) Relative error

Figure. 4.1. Comparison of full model and ROM of 3-dimensional model for $Re = 500$

4.2.2 \mathcal{H}_2 -norm of the error system for the ROMs

Here, \mathcal{H}_2 -norm of the error system for the ROMs of the target models will be provided.

Table. 4.2. \mathcal{H}_2 error norm of the ROMs of the target models

Reynolds Number	Dimension	\mathcal{H}_2 error norm
300	1	2.11×10^{-03}
	2	7.52×10^{-04}
	3	7.90×10^{-04}
	4	8.12×10^{-04}
	5	8.49×10^{-04}
400	1	1.55×10^{-02}
	2	3.25×10^{-03}
	3	3.57×10^{-03}
	4	3.95×10^{-03}
	5	4.15×10^{-03}
500	1	2.62×10^{-01}
	2	1.22×10^{-02}
	3	1.74×10^{-02}
	4	1.88×10^{-02}
	5	9.46×10^{-03}

Table 4.2 conveys the desired \mathcal{H}_2 error norm of the ROMs. From the measured \mathcal{H}_2 error norm of the ROMs, it is evident that the level of the approximation process is suitably robust and confirms the compatible scale of the accuracy.

4.3 Graphical Comparisons of Stabilization of the Unstable Systems

In this section, stabilization of the unstable Navier-Stokes models will be graphically illustrated. For the compactness of this thesis, we will only demonstrate the stabilization of the transient behaviors of the 3-dimensional model.

4.3.1 Stabilization of the eigenvalues

In the sub-figures of the Figure 4.2 exhibit the eigenvalues of the original (unstable) 3-dimensional systems with Reynolds number $Re = 300, 400, 500$, respectively. In

contrast, the sub-figures of the Figure 4.3 exhibit the stabilized eigenvalues of that of the systems. From those figures, it can be concluded that the eigenvalues of the desired models are properly stabilized.

4.3.2 Stabilization of the step-responses

In the sub-figures of the Figure 4.4 and Figure 4.6 display the step-responses of the 1st-input/1st-output and 2nd-input/7th-output relations, respectively, of the above-mentioned original (unstable) systems. On the contrary, the sub-figures of the Figure 4.4 and Figure 4.6 display the stabilized step-responses of the 1st-input/1st-output and 2nd-input/7th-output relations, respectively, of that of the systems. The scenario of the step-response stabilization duly confirms the efficiency of the proposed techniques.

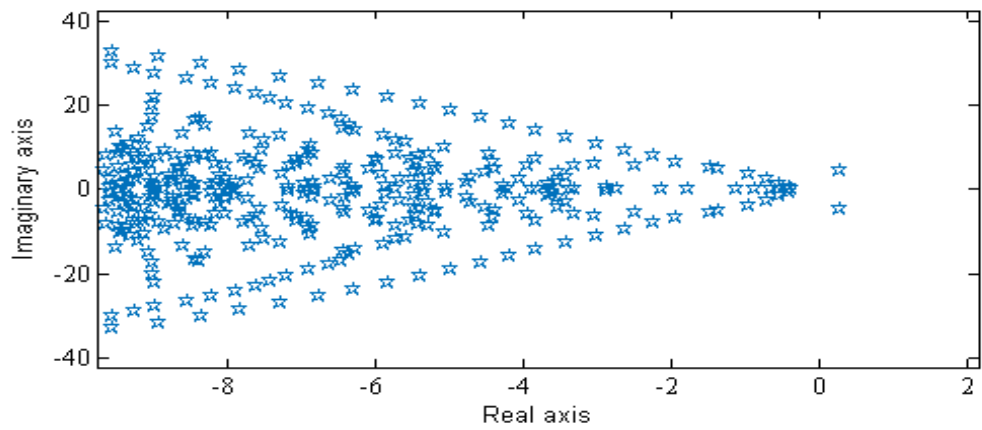
4.4 Comparison of the TSIA with IRKA

In this section, we are going to compare the proposed algorithm with the IRKA approach for reduced-order modelling. Initially, we have aimed to compare with the RKSM approach as well. But due to the non-symmetric structure of the Navier-Stokes models, the RKSM approach is applicable for here. In this comparative analysis, we will investigate both of the computation time and the \mathcal{H}_2 error norm of the ROMs of selected 3-dimensional models. Table 4.2 reveals the required information of computation time and \mathcal{H}_2 error norm.

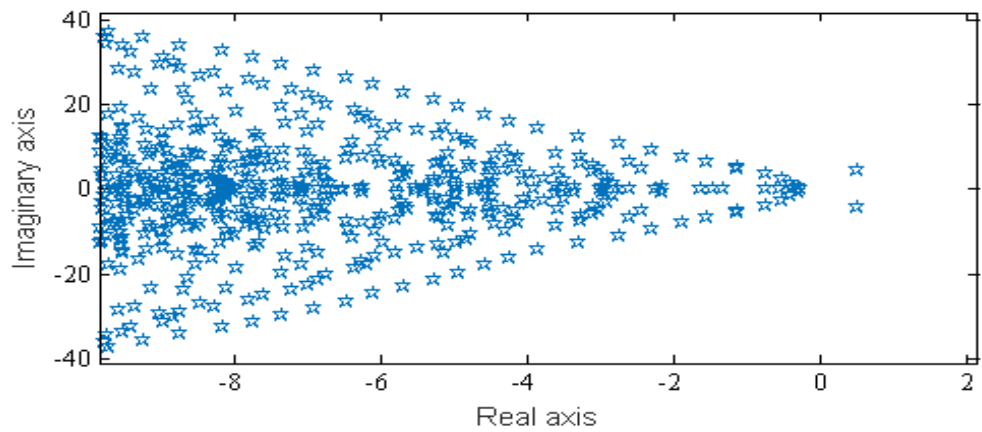
Table. 4.3. Comparison of computation time and \mathcal{H}_2 error norm of the ROMs of 3-dimensional models archived by TSIA and IRKA

Reynolds Number	Time		\mathcal{H}_2 error norm	
	TSIA	IRKA	TSIA	IRKA
300	6.81×10^3	9.83×10^3	7.90×10^{-04}	9.97×10^{-04}
400	6.86×10^3	9.65×10^3	3.57×10^{-03}	6.30×10^{-03}
500	6.94×10^3	9.68×10^3	1.74×10^{-02}	5.13×10^{-02}

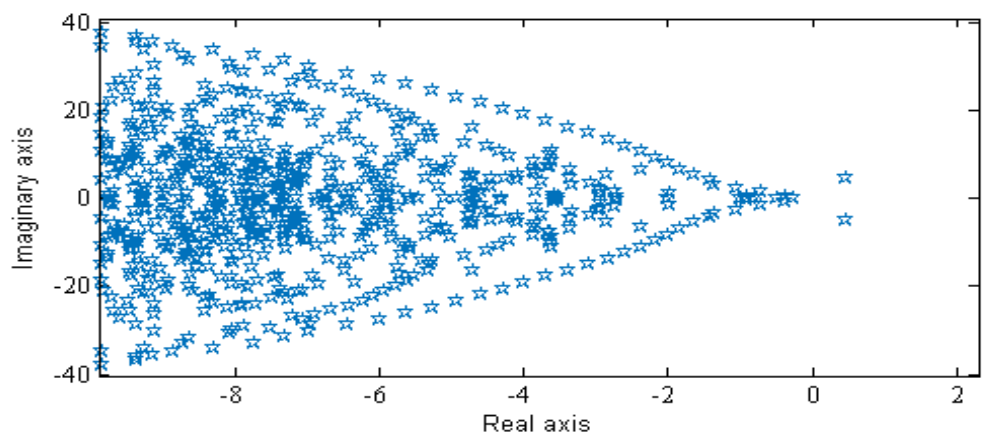
It to be noted that stabilization of the eigenvalues and step-responses by the TSIA and IRKA via the reduced-order modelling are very identical. Thus, we have ignored their graphical comparisons and chosen the tabular comparisons.



(a) Unstable eigenvalues Reynolds number 300

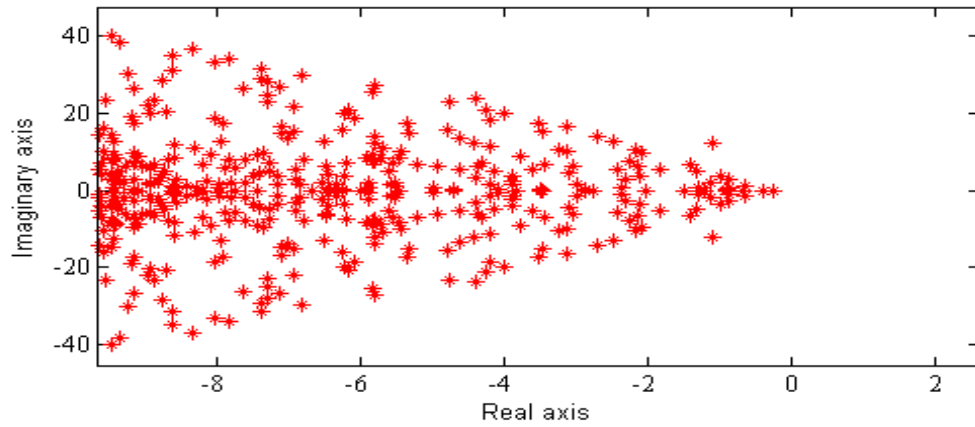


(b) Unstable eigenvalues Reynolds number 400

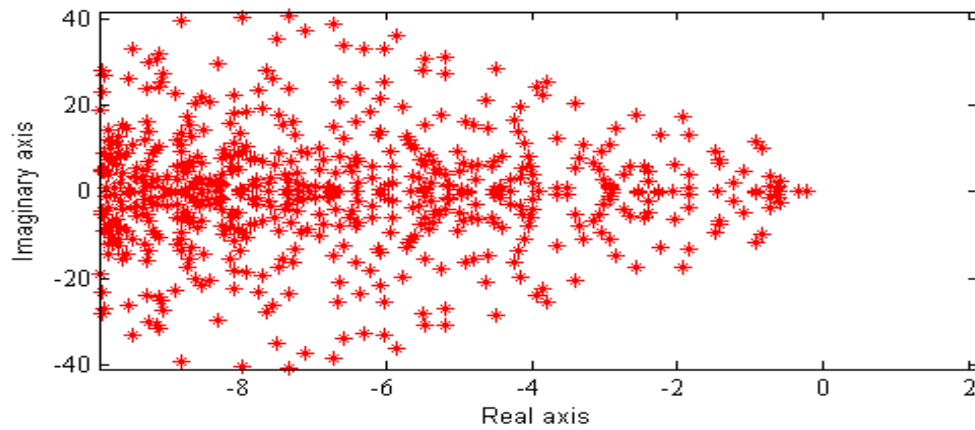


(c) Unstable eigenvalues Reynolds number 500

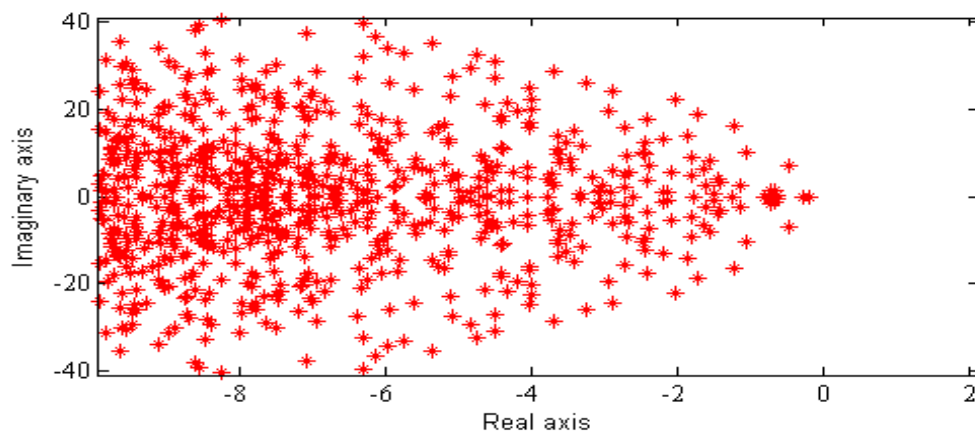
Figure. 4.2. Unstable eigenvalues of 3-dimensional models



(a) Stabilized eigenvalues Reynolds number 300

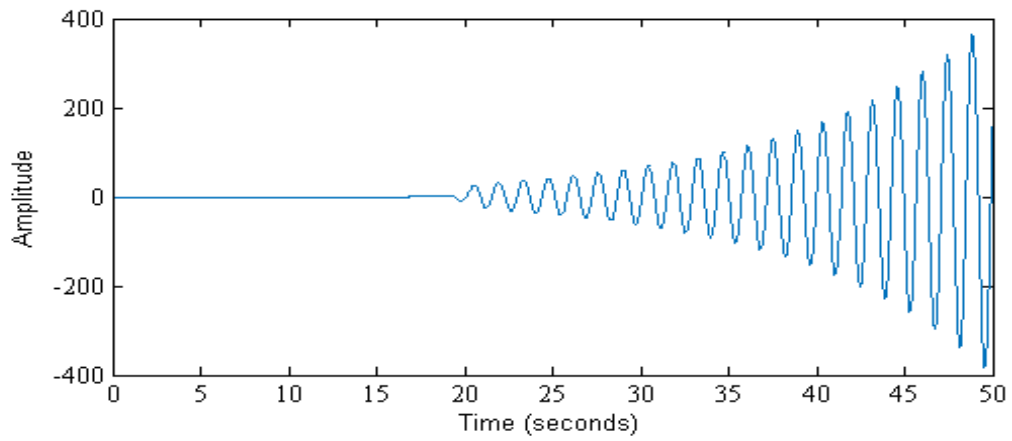


(b) Stabilized eigenvalues Reynolds number 400

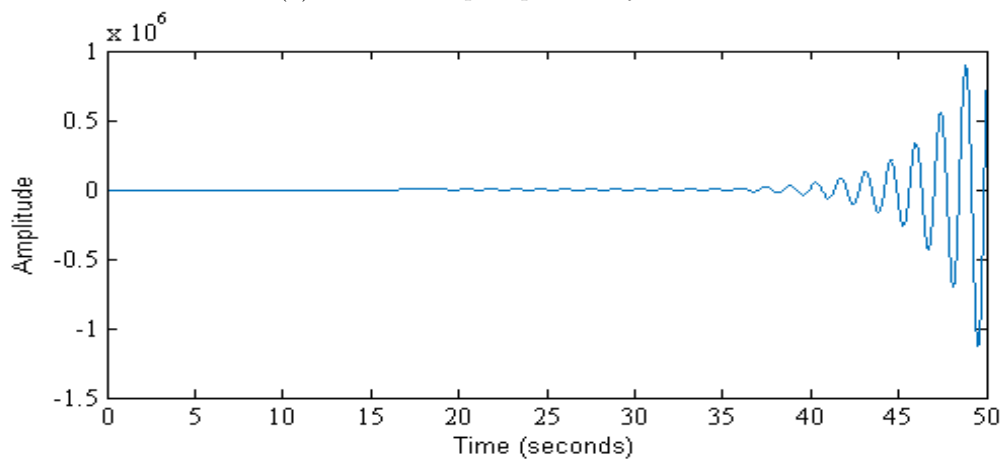


(c) Stabilized eigenvalues Reynolds number 500

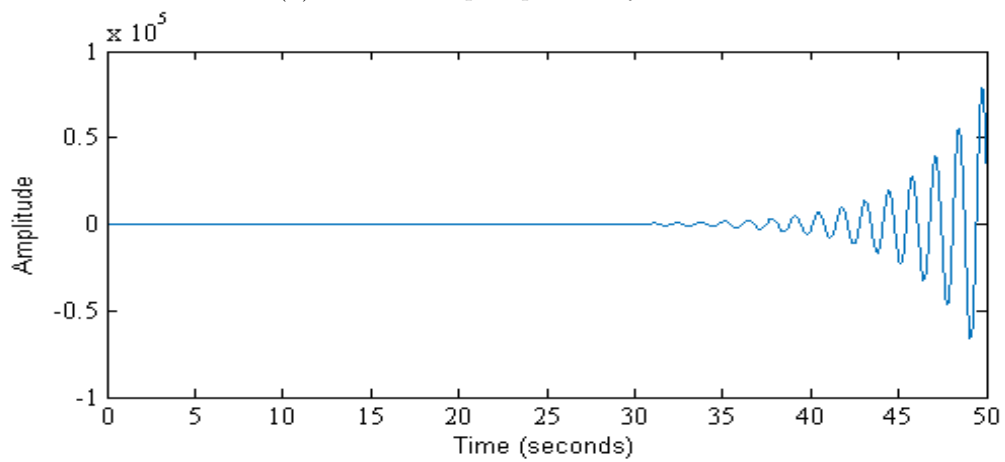
Figure. 4.3. Stabilized eigenvalues of 3-dimensional models



(a) Unstable step response Reynolds 300

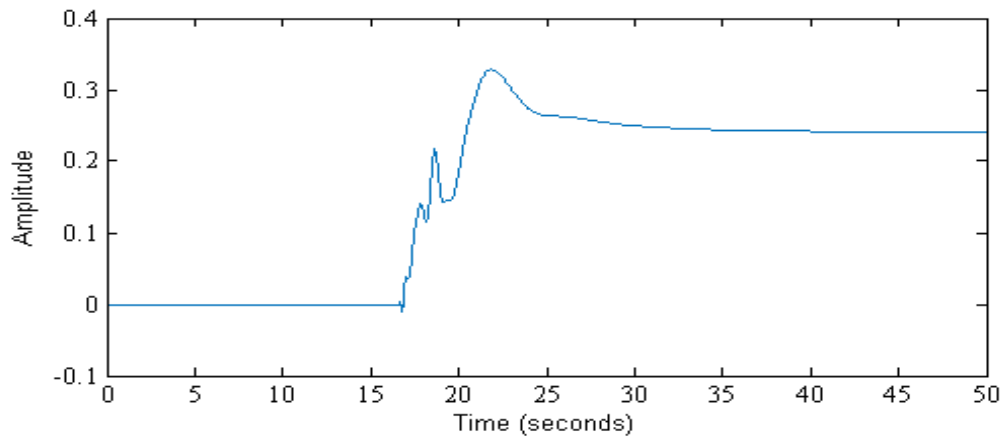


(b) Unstable step response Reynolds 400

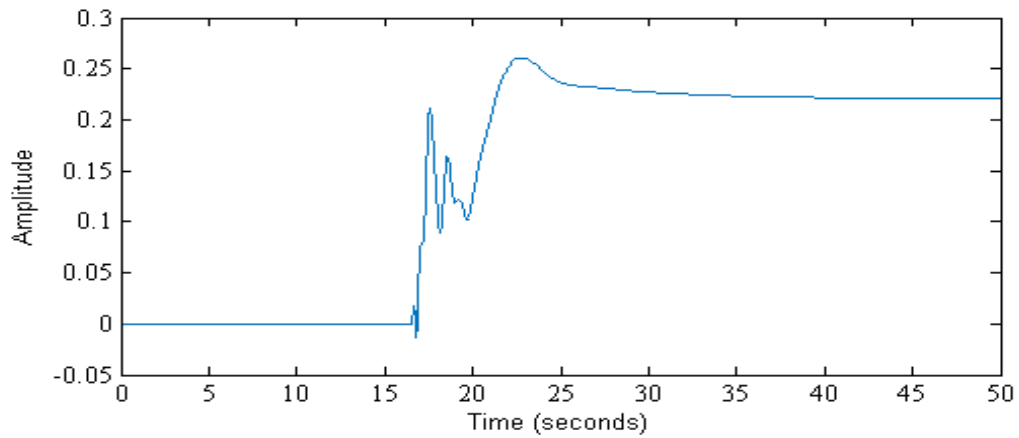


(c) Unstable step response Reynolds 500

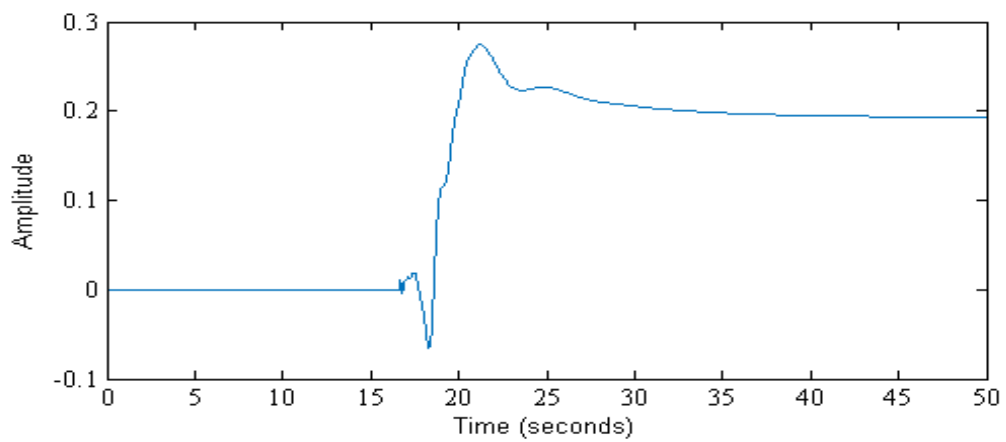
Figure. 4.4. Unstable step response for 1st input and 1st output of 3-dimensional models



(a) Stabilized step response Reynolds 300

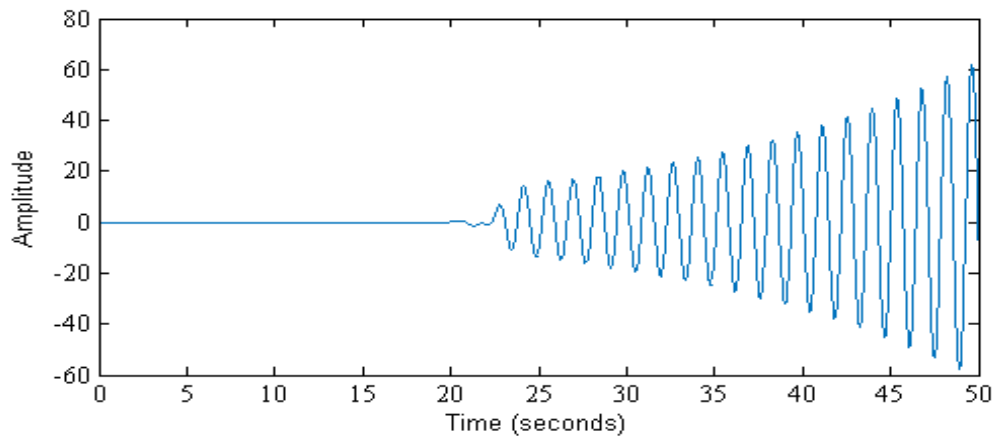


(b) Stabilized step response Reynolds 400

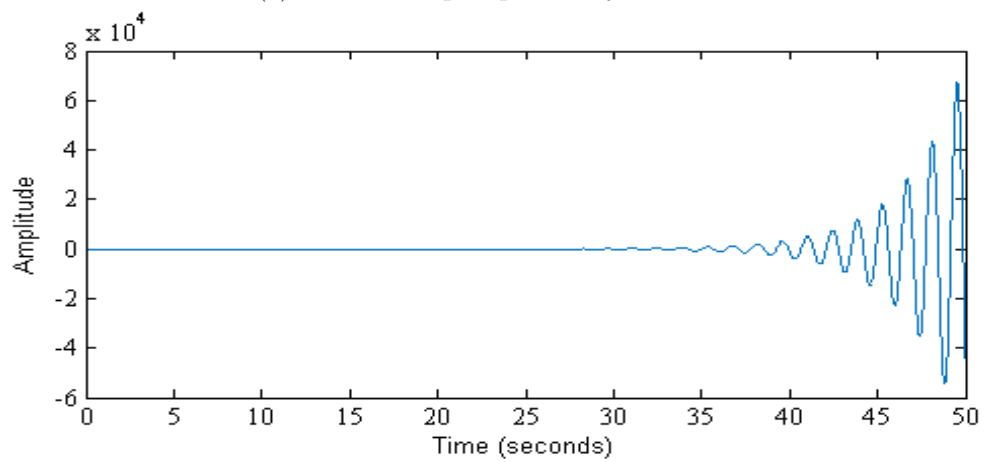


(c) Stabilized step response Reynolds 500

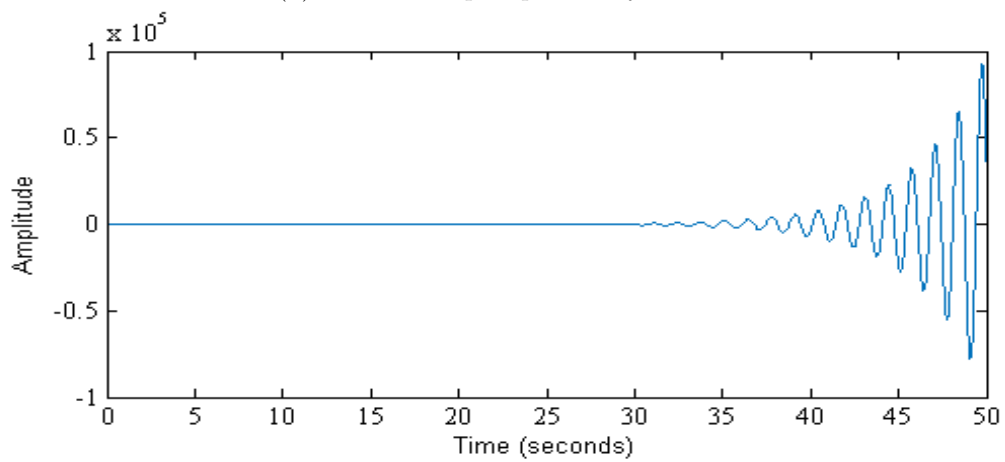
Figure. 4.5. Stabilized step response for 1st input and 1st output of 3-dimensional models



(a) Unstable step response Reynolds 300

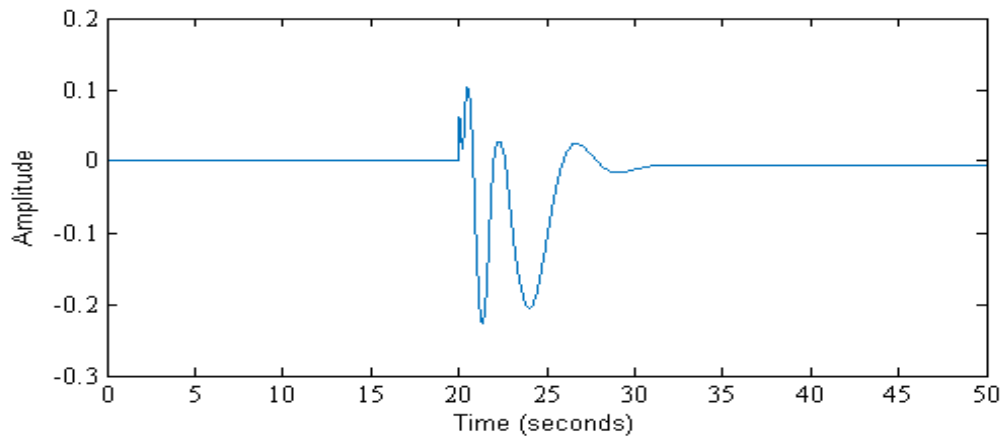


(b) Unstable step response Reynolds 400

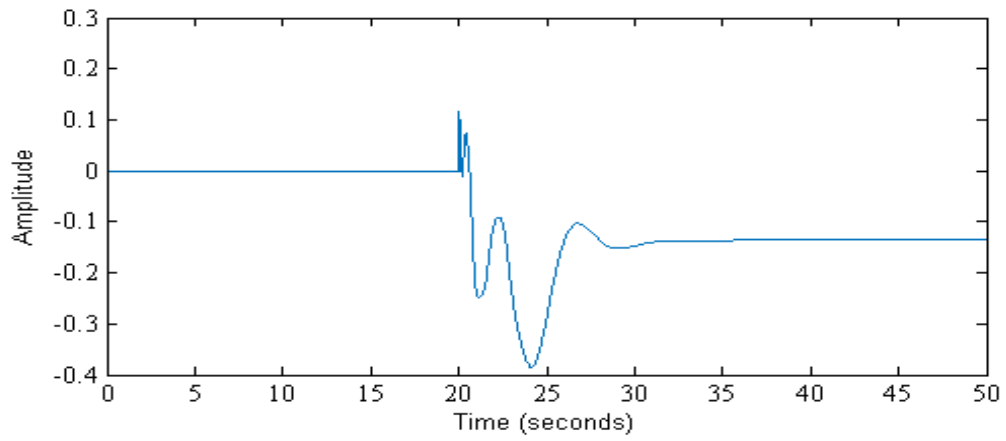


(c) Unstable step response Reynolds 500

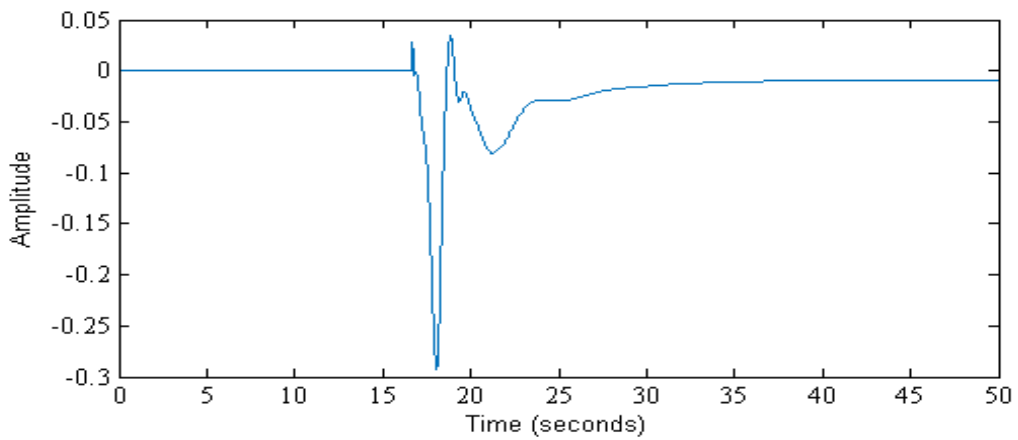
Figure. 4.6. Unstable step response for 2nd input and 7th output of 3-dimensional models



(a) Stabilized step response Reynolds 300



(b) Stabilized step response Reynolds 400



(c) Stabilized step response Reynolds 500

Figure. 4.7. Stabilized step response for 2nd input and 7th output of 3-dimensional models

From the above table, it is apparent that in case of computation time TSIA algorithm requires about two-thirds of that of IRKA. Also, comparison of \mathcal{H}_2 error norm of the ROMs manifests the betterment of TSIA over IRKA.

Chapter 5

Conclusion

In this chapter, a summary of the thesis work will be included consisting of findings in the current research, limitations in the computational activities, and a proposal for future research.

5.1 Summary

The thesis is mainly compacted with the approximation of the full models with the ROMs, finding the reduced-order feedback matrices and deriving the optimal feedback matrices for the unstable Navier-Stokes models.

We have introduced a two-sided projection-based sparsity-preserving reduced-order modeling approach for the stabilization of non-symmetric index-2 descriptor systems explored from unstable Navier-Stokes models with \mathcal{H}_2 optimality. Required Reduced-order models are derived by the two-sided projection techniques implementing the sparse-dense Sylvester equations to minimize the computation time, ensure the requirement of less memory allocation, and enhance the stability of the reduced-order models by satisfying the Wilson conditions. Modified structures of sparsity-preserving Krylov subspaces are introduced to solve the desired Sylvester equations, involving a simplified form system of linear equations solvable by the direct matrix solvers. Reduced-order feedback matrices are estimated from reduced-order models. Finally, the classical inverse projection scheme is deployed to attain the optimal feedback matrices of the full models and stabilize them.

From the numerical analysis, it is ascertained that the proposed techniques can be proficiently applied to stabilize the target models through reduced-order modeling.

From the tabular and graphical comparisons of the results of numerical computations, the findings for the proposed techniques are as follows:

- RKSM is not applicable for the target models due to the non-symmetric structure, whereas TSIA can be suitably applied.
- Sparse-dense Sylvester equations can be implemented to find the projection matrices in the reduced-order modelling.
- Full models can be efficiently approximated by the corresponding ROMs with minimized \mathcal{H}_2 error norm.
- Inverse projection scheme is effective in computing the optimal feedback matrices from the reduced-order feedback matrices.
- Eigenvalues and step-responses of the target models can be properly stabilized.
- The techniques involved in TSIA outplayed that of the IRKA in both the saving computation time and \mathcal{H}_2 error norm.

Thus, the bottom line of the thesis is that the proposed techniques can be utilized to stabilize the unstable Navier-Stokes models with better accuracy and less computing time.

5.2 Limitations

In this thesis, two Sylvester equations is required to find the projector matrices at which initial makeshift ROMs are required, which may exploit the robustness of the computation. A inverse projection technique is used to find the optimal feedback matrices from the reduced-order feedback matrices, this dependency on the ROMs is sometimes infeasible due to the less robustness of the projector matrices.

5.3 Future Research

The research of the thesis can be extended to find the ROMs of the second-order state-space systems.

In future research, we will try to overcome the requirement of the makeshift ROMs. Also, instead inverse projection technique a straight-forward eigen-decomposition technique will be tried to find the optimal feedback matrices. Further investigation is required to enhance the rapid convergence. Moreover, apply TSIA for unstable index-3 descriptor systems will be tried.

References

- [1] P. J. Antsaklis and A. N. Michel, *Linear systems*. Springer Science & Business Media, 2006.
- [2] J. T. Borggaard and S. Gugercin, “Model reduction for daes with an application to flow control,” *Active Flow and Combustion Control 2014*, vol. 127, p. 381, 2014.
- [3] M. M. Rahman, M. Uddin, M. M. Uddin, and Andallah, “Svd-krylov based techniques for structure-preserving reduced order modelling of second-order systems,” *Mathematical Modelling and Control*, vol. 1, no. 2, pp. 79–89, 2021.
- [4] E. Bänsch, P. Benner, J. Saak, and H. K. Weichelt, “Riccati-based boundary feedback stabilization of incompressible navier-stokes flow,” *SIAM Journal on Scientific Computing*, vol. 37, no. 2, pp. A832–A858, 2015.
- [5] P. Benner, “Preface: System reduction for nanoscale ic design,” 2017.
- [6] I. Kose, “Introduction to state-space control theory,” *Department of Mechanical Engineering, Bogazici University*, 2003.
- [7] P. Benner and T. Stykel, “Model order reduction for differential-algebraic equations: A survey,” *Surveys in Differential-Algebraic Equations IV*, p. 107, 2017.
- [8] M. M. Rahman, M. M. Uddin, L. Andallah, and M. Uddin, “Tangential interpolatory projections for a class of second-order index-1 descriptor systems and application to mechatronics,” *Production Engineering*, vol. 15, no. 1, pp. 9–19, 2021.
- [9] M. Uddin, M. M. Uddin, M. A. H. Khan, and M. T. Hossain, “Svd-krylov based sparsity-preserving techniques for riccati-based feedback stabilization

- of unstable power system models,” *Journal of Engineering Advancements*, vol. 2, no. 3, pp. 125–131, 2021.
- [10] V. Simoncini, “Analysis of the rational krylov subspace projection method for large-scale algebraic riccati equations,” *SIAM Journal on Matrix Analysis and Applications*, vol. 37, no. 4, pp. 1655–1674, 2016.
- [11] D. Palitta, “The projected newton-kleinman method for the algebraic riccati equation,” *arXiv preprint arXiv:1901.10199*, 2019.
- [12] I. V. Gosea and A. C. Antoulas, “A two-sided iterative framework for model reduction of linear systems with quadratic output,” in *2019 IEEE 58th Conference on Decision and Control (CDC)*, 2019, pp. 7812–7817.
- [13] A. Sharma, K. K. Paliwal, S. Imoto, and S. Miyano, “Principal component analysis using qr decomposition,” *International Journal of Machine Learning and Cybernetics*, vol. 4, no. 6, pp. 679–683, 2013.
- [14] P. Benner and T. Breiten, “Two-sided projection methods for nonlinear model order reduction,” *SIAM Journal on Scientific Computing*, vol. 37, no. 2, pp. B239–B260, 2015.
- [15] T. Kailath, *Linear systems*. Prentice-Hall Englewood Cliffs, NJ, 1980, vol. 156.
- [16] M. Uddin, “Numerical study on continuous-time algebraic riccati equations arising from large-scale sparse descriptor systems,” Master’s thesis, Bangladesh University of Engineering and Technology, 2020.
- [17] T. Stykel, “Analysis and numerical solution of generalized lyapunov equations,” *Institut für Mathematik, Technische Universität, Berlin*, 2002.
- [18] P. Kunkel and V. Mehrmann, *Differential-algebraic equations: analysis and numerical solution*. European Mathematical Society, 2006, vol. 2.
- [19] S. Gugercin, T. Stykel, and S. Wyatt, “Model reduction of descriptor systems by interpolatory projection methods,” *SIAM Journal on Scientific Computing*, vol. 35, no. 5, pp. B1010–B1033, 2013.
- [20] S. A. Sheldon Axler, “Linear algebra done right,” 2015.

- [21] P. Benner, S. Gugercin, and K. Willcox, “A survey of projection-based model reduction methods for parametric dynamical systems,” *SIAM review*, vol. 57, no. 4, pp. 483–531, 2015.
- [22] B. N. Datta, *Numerical linear algebra and applications*. Siam, 2010, vol. 116.
- [23] Z. Gajic and M. Qureshi, “Lyapunov matrix equation in system stability and control, 1995.”
- [24] A. C. Antoulas, *Approximation of large-scale dynamical systems*. SIAM, 2005.
- [25] M. Köhler and J. Saak, “Efficiency improving implementation techniques for large scale matrix equation solvers,” 2009.
- [26] D. C. Sorensen and A. Antoulas, “The sylvester equation and approximate balanced reduction,” *Linear algebra and its applications*, vol. 351, pp. 671–700, 2002.
- [27] C. F. Van Loan and G. Golub, “Matrix computations (johns hopkins studies in mathematical sciences),” 1996.
- [28] M. M. Uddin, *Computational methods for approximation of large-scale dynamical systems*. CRC Press, 2019.
- [29] P. Benner, J. Saak, and M. M. Uddin, “Balancing based model reduction for structured index-2 unstable descriptor systems with application to flow control,” *Numerical Algebra, Control & Optimization*, vol. 6, no. 1, p. 1, 2016.
- [30] P. Benner, P. Kürschner, and J. Saak, “An improved numerical method for balanced truncation for symmetric second-order systems,” *Mathematical and Computer Modelling of Dynamical Systems*, vol. 19, no. 6, pp. 593–615, 2013.
- [31] C. Cosentino and D. Bates, *Feedback control in systems biology*. Crc Press, 2019.
- [32] P. Benner and J. Heiland, “Convergence of approximations to riccati-based boundary-feedback stabilization of laminar flows,” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 12 296–12 300, 2017.

-
- [33] P. Benner, Z. Bujanović, P. Kürschner, and J. Saak, “A numerical comparison of solvers for large-scale, continuous-time algebraic riccati equations,” *arXiv preprint arXiv:1811.00850*, 2018.
- [34] I. S. Duff, A. M. Erisman, and J. K. Reid, *Direct methods for sparse matrices*. Oxford University Press, 2017.
- [35] J. T. Betts, *Practical methods for optimal control and estimation using nonlinear programming*. SIAM, 2010.
- [36] D. S. Watkins, *The matrix eigenvalue problem: GR and Krylov subspace methods*. SIAM, 2007.
- [37] Y. Saad, *Numerical methods for large eigenvalue problems: revised edition*. Siam, 2011, vol. 66.
- [38] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. Baltimore: Johns Hopkins University Press, 1996.
- [39] J. H. Wilkinson, *The algebraic eigenvalue problem*. Oxford Clarendon, 1965, vol. 662.
- [40] H. Roger and R. J. Charles, *Topics in matrix analysis*. Cambridge University Press, 1994.
- [41] A. Van den Bos, *Parameter estimation for scientists and engineers*. John Wiley & Sons, 2007.
- [42] R. Horn and C. Johnson, “Matrix analysis, cambridge univ,” *Press. MR0832183*, 1985.
- [43] C. R. Johnson and R. A. Horn, *Matrix analysis*. Cambridge university press Cambridge, 1985.
- [44] L. N. Trefethen and D. Bau III, *Numerical linear algebra*. Siam, 1997, vol. 50.
- [45] J. R. Schott, *Matrix analysis for statistics*. John Wiley & Sons, 2016.
- [46] M. C. Seiler and F. A. Seiler, “Numerical recipes in c: the art of scientific computing,” *Risk Analysis*, vol. 9, no. 3, pp. 415–416, 1989.
- [47] K. Jbilou, “Block krylov subspace methods for large algebraic riccati equations,” *Numerical algorithms*, vol. 34, no. 2, pp. 339–353, 2003.

- [48] D. Kressner, “Memory-efficient krylov subspace techniques for solving large-scale lyapunov equations,” in *2008 IEEE International Conference on Computer-Aided Control Systems*. IEEE, 2008, pp. 613–618.
- [49] J. W. Demmel, *Applied numerical linear algebra*. Siam, 1997, vol. 56.
- [50] D. A. Bini, B. Iannazzo, and B. Meini, *Numerical solution of algebraic Riccati equations*. SIAM, 2011.
- [51] P. Benner and Z. Bujanović, “On the solution of large-scale algebraic riccati equations by using low-dimensional invariant subspaces,” *Linear Algebra and its Applications*, vol. 488, pp. 430–459, 2016.
- [52] W. F. Arnold and A. J. Laub, “Generalized eigenproblem algorithms and software for algebraic riccati equations,” *Proceedings of the IEEE*, vol. 72, no. 12, pp. 1746–1754, 1984.
- [53] R. Byers, “A hamiltonian-jacobi algorithm,” *IEEE transactions on automatic control*, vol. 35, no. 5, pp. 566–570, 1990.
- [54] S. Wyatt, “Issues in interpolatory model reduction: Inexact solves, second-order systems and daes,” Ph.D. dissertation, Virginia Tech, 2012.
- [55] D. Palitta and V. Simoncini, “Optimality properties of galerkin and petrov–galerkin methods for linear matrix equations,” *Vietnam Journal of Mathematics*, vol. 48, no. 4, pp. 791–807, 2020.
- [56] S. Gugercin, A. C. Antoulas, and C. Beattie, “H₂ model reduction for large-scale linear dynamical systems,” *SIAM journal on matrix analysis and applications*, vol. 30, no. 2, pp. 609–638, 2008.
- [57] E. Wachspress, *The ADI model problem*. Springer, 2013.
- [58] J. R. Li, “Model reduction of large linear systems via low rank system gramians,” Ph.D. dissertation, Massachusetts Institute of Technology, 2000.
- [59] T. Penzl, “A cyclic low-rank smith method for large sparse lyapunov equations,” *SIAM Journal on Scientific Computing*, vol. 21, no. 4, pp. 1401–1418, 1999.
- [60] K. Morris and C. Navasca, “Approximation of low rank solutions for linear quadratic control of partial differential equations,” *Computational Optimization and Applications*, vol. 46, no. 1, pp. 93–111, 2010.

-
- [61] D. C. Sorensen and Y. Zhou, “Direct methods for matrix sylvester and lyapunov equations,” *Journal of Applied Mathematics*, vol. 2003, pp. 277–303, 2002.
- [62] A. N. Krylov, “On the numerical solution of the equation by which in technical questions frequencies of small oscillations of material systems are determined,” *Izvestija AN SSSR (News of Academy of Sciences of the USSR), Otdel. mat. i estest. nauk*, vol. 7, no. 4, pp. 491–539, 1931.
- [63] C. Lein, M. Beitelschmidt, and D. Bernstein, “Improvement of krylov-subspace-reduced models by iterative mode-truncation,” *IFAC-PapersOnLine*, vol. 48, no. 1, pp. 178–183, 2015.
- [64] S. Gugercin and A. Antoulas, “A comparative study of 7 algorithms for model reduction,” in *Proceedings of the 39th IEEE Conference on Decision and Control (Cat. No. 00CH37187)*, vol. 3. IEEE, 2000, pp. 2367–2372.
- [65] A. Jeffrey and D. Zwillinger, *Table of integrals, series, and products*. Elsevier, 2007.
- [66] A. Bunse-Gerstner, D. Kubalińska, G. Vossen, and D. Wilczek, “h2-norm optimal model reduction for large scale discrete dynamical mimo systems,” *Journal of computational and applied mathematics*, vol. 233, no. 5, pp. 1202–1216, 2010.
- [67] L. Xie and E. de Souza Carlos, “Robust h/sub infinity/control for linear systems with norm-bounded time-varying uncertainty,” *IEEE Transactions on Automatic Control*, vol. 37, no. 8, pp. 1188–1191, 1992.
- [68] J. Saak, P. Benner, and P. Kürschner, “A goal-oriented dual lrcf-adi for balanced truncation,” *IFAC Proceedings Volumes*, vol. 45, no. 2, pp. 752–757, 2012.
- [69] P. Benner, P. Kürschner, and J. Saak, “Self-generating and efficient shift parameters in adi methods for large lyapunov and sylvester equations,” *Electronic Transactions on Numerical Analysis (ETNA)*, vol. 43, pp. 142–162, 2014.
- [70] ———, “Efficient handling of complex shift parameters in the low-rank cholesky factor adi method,” *Numerical Algorithms*, vol. 62, no. 2, pp. 225–251, 2013.

- [71] Y. W. Kwon and H. Bang, *The finite element method using MATLAB*. CRC press, 2018.
- [72] P. Hood and C. Taylor, “Navier-stokes equations using mixed interpolation,” *Finite element methods in flow problems*, pp. 121–132, 1974.
- [73] T. Stykel, “Balanced truncation model reduction for semidiscretized stokes equation,” *Linear Algebra and its Applications*, vol. 415, no. 2-3, pp. 262–289, 2006.
- [74] E. Bänsch, P. Benner, J. Saak, and H. K. Weichelt, “Optimal control-based feedback stabilization of multi-field flow problems,” *Trends in PDE Constrained Optimization*, pp. 173–188, 2014.
- [75] P. Benner, M. Heinkenschloss, J. Saak, and H. K. Weichelt, “Efficient solution of large-scale algebraic riccati equations associated with index-2 daes via the inexact low-rank newton-adi method,” *Applied Numerical Mathematics*, vol. 152, pp. 338–354, 2020.
- [76] F. Gargano, M. Sammartino, V. Sciacca, and K. W. Cassel, “Analysis of complex singularities in high-reynolds-number navier–stokes solutions,” *Journal of fluid mechanics*, vol. 747, pp. 381–421, 2014.
- [77] J.-P. Raymond, “Local boundary feedback stabilization of the navier-stokes equations,” *Control Systems: Theory, Numerics and Applications, Rome*, vol. 30, 2005.
- [78] P. Benner, Z. Bujanović, P. Kürschner, and J. Saak, “A numerical comparison of different solvers for large-scale, continuous-time algebraic riccati equations and lqr problems,” *SIAM Journal on Scientific Computing*, vol. 42, no. 2, pp. A957–A996, 2020.
- [79] R. Altmann and J. Heiland, “Continuous, semi-discrete, and fully discretised navier-stokes equations,” *Applications of Differential-Algebraic Equations: Examples and Benchmarks*, p. 277, 2018.
- [80] M. Heinkenschloss, D. C. Sorensen, and K. Sun, “Balanced truncation model reduction for a class of descriptor systems with application to the oseen equations,” *SIAM Journal on Scientific Computing*, vol. 30, no. 2, pp. 1038–1063, 2008.

- [81] K.-L. Xu and Y.-L. Jiang, “Reduced optimal models via cross gramian for continuous linear time-invariant systems,” *IET Circuits, Devices & Systems*, vol. 12, no. 1, pp. 25–32, 2018.
- [82] P. Benner, M. Köhler, and J. Saak, “Sparse-dense sylvester equations in h_2 -model order reduction,” 2011.
- [83] M. M. Rahman, M. M. Uddin, L. S. Andallah, and M. Uddin, “Interpolatory projection techniques for \mathcal{H}_2 optimal structure-preserving model order reduction of second-order systems,” *Advances in Science, Technology and Engineering Systems Journal*, vol. 5, no. 4, pp. 715–723, 2020.