Interpolation Based Parametric Model Order Reduction

von Nguyen Thanh Son

Dissertation

zur Erlangung des Grades eines Doktors der Naturwissenschaften

-Dr. rer. nat.-

Vorgelegt im Fachbereich 3 (Mathematik & Informatik)
der Universität Bremen
im Januar 2012

Datum des Promotionskolloquiums: 27.01.2012

Gutachter: Prof. Dr. Angelika Bunse-Gerstner (Universität Bremen)
Prof. Dr. Peter Benner (Max-Planck-Institut für Dynamik komplexer technischer Systeme)

Acknowledgments

Foremost, I would like to express my sincere gratitude to my advisor, Professor Angelika Bunse-Gerstner. She introduced the topic to me and handed me the freedom to work on it. She always provided me with the best support that she could, gave me advice and encouragements without which I have not been able to overcome some desperation time during the PhD project.

My sincere thanks go to Dr. Ulrike Baur for her suggestions and comments that helped me to improve the presentation of this thesis.

I am also grateful to Dr. Dorota Kubalińska and Diane Wilzeck for their helps. Dorota answered me many questions on the topic and corrected many early pages of my English writing, while Diane helped me in doing administration stuff during my first days in Bremen.

I own thanks to my colleagues: Andreas Bartel, Kanglin Chen, Bastian Kanning, Quy Muoi Pham, and Majid Salmani. Thank you guys for useful discussions (not only about mathematics,) friendship and fun we had together.

This work was within the Scientific Computing in Engineering (SCiE) program at Zentrum für Technomathmatik (ZeTeM) funded by Universität Bremen to which I would like to acknowledge.

I am especially grateful to my mother who has raised my brothers with incredible efforts. She always supported my pursuits and hoped for my success. This thesis partly shows that all her sacrifice is not in vain.

Last but not least, my dearest thanks go to my fiancée for her constant understanding and encouragement. Her love for me and her trust on me are always my endless and indispensable source of motivation which helped me to accomplish this thesis.

Zusammenfassung

Diese Arbeit befasst sich mit der Ordnungsreduktion parameterabhängiger, großer dynamischer Systeme. Ziel ist es, eine Methodik zu entwickeln, um die Ordnung des Modells zu reduzieren und gleichzeitig die Parameter-Abhängigkeit zu erhalten.

Wir nutzen zunächst die Methode des Balancierten Abschneidens in Verbindung mit Spline-Interpolation, um das Problem zu lösen. Kern dieser Methode ist die Interpolations der reduzierten Übertragungsfunktion, basierend auf einer zuvor berechneten Übertragungsfunktion eines Samples des Parameter-Domains. Sowohl lineare, als auch kubische Splines werden getestet. Wie erwartet verbessert die Verwendung letzterer den Fehler der Methode. Es wird gezeigt, dass diese Kombination die Robustheit des Balancierten Abschneidens sowie dessen Stabilitäts-Erhaltung und, basierend auf einer neuen Schranke für die Unendlich-Norm der inversen Matrix, Fehlerschranken aufweist.

Die Ordnungsreduction kann in einem Projektionen-Rahmen formuliert werden, und im Falle eines Parameter-abhängigen Systems hängen die jeweiligen Projektions-Unterräume auch von Parametern an. Man kann diese Parameter-abhängigen Projektions-Unterräume nicht explizit bestimmen, jedoch durch Interpolation auf der Grundlage einer Reihe im Voraus berechneter Unterräume approximieren. Es stellt sich heraus, dass dies das Problem der Interpolation auf Grassmann Mannigfaltigkeiten ist. Die Interpolation wird auf Tangentialräumen der zugrunde liegenden Mannigfaltigkeiten durchgeführt. Um dies zu erreichen, muss man die Exponential- und Logarithmus-Abbildung, inklusive einer Singulärwertzerlegung anwenden. Der Prozess wird in eine Offline- und eine Online-Phase unterteilt. Die Rechenzeit der Online-Phase ist ein entscheidender Punkt. Durch die Untersuchung der Formulierung der Exponential- und Logarithmus-Abbildung, und die Analyse der Struktur von Summen von Singulärenwertzerlegungen, gelingt es uns, die rechnerische Komplexität der Online-Phase zu reduzieren, und damit ist die Nutzung dieses Algorithmuses in Echtzeit möglich.

Interpolation Based Parametric Model Order Reduction

Abstract: In this thesis, we consider model order reduction of parameter-dependent large-scale dynamical systems. The objective is to develop a methodology to reduce the order of the model and simultaneously preserve the dependence of the model on parameters.

We use the balanced truncation method together with spline interpolation to solve the problem. The core of this method is to interpolate the reduced transfer function, based on the pre-computed transfer function at a sample in the parameter domain. Linear splines and cubic splines are employed here. The use of the latter, as expected, reduces the error of the method. The combination is proven to inherit the advantages of balanced truncation such as stability preservation and, based on a novel bound for the infinity norm of the matrix inverse, the derivation of error bounds.

Model order reduction can be formulated in the projection framework. In the case of a parameter-dependent system, the projection subspace also depends on parameters. One cannot compute this parameter-dependent projection subspace, but has to approximate it by interpolation based on a set of pre-computed subspaces. It turns out that this is the problem of interpolation on Grassmann manifolds. The interpolation process is actually performed on tangent spaces to the underlying manifold. To do that, one has to invoke the exponential and logarithmic mappings which involve some singular value decompositions. The whole procedure is then divided into the offline and online stage. The computation time in the online stage is a crucial point. By investigating the formulation of exponential and logarithmic mappings and analyzing the structure of sums of singular value decompositions, we succeed to reduce the computational complexity of the online stage and therefore enable the use of this algorithm in real time.

Keywords: Parametric model order reduction, spline, interpolation, Grassmann manifolds, real time.

Contents

Chapter 1

Introduction

Contents

1.1 Motivation

Nowadays, numerical simulation is vital to manufactures. This step, on the one hand, helps designers to create prototypes that fulfill requirements of producers. On the other hand, it can serve as cheap and/or time-saving surrogate for experiments, since real tests are usually expensive and time-consuming.

As the first step of a simulation, one has to search for a mathematical model which describes the behavior of the device, or a single unit of it. Forming such a model is based on laws in physics, chemistry, etc. One ends up with a set of partial differential equations (PDEs) and typically has to solve these equations numerically. To this end, PDEs must be discretized in space by means of some numerical method such as the finite element method (FEM) or finite difference method (FDM). By appropriate linearization and expansion, this generally results in a linear, time-invariant system of the form

$$\begin{aligned} E\dot{x}(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + Du(t), \end{aligned} \tag{1.1}$$

where $E, A \in \mathbb{R}^{N\times N}, B \in \mathbb{R}^{N\times m}, C \in \mathbb{R}^{l\times N}, D \in \mathbb{R}^{l\times m}$ are real or complex matrices; $x(t)$ is a vector representing the state of the system which depends on time t; $u(t)$ represents the input given by the user or determined by a process, called input or control function, which affects the system behavior; $y(t)$ is some information extracted from the state $x(t)$ and the input $u(t)$. System (1.1) is the mathematical clarification of the input-output correspondence. One inserts an input $u(t)$ and observes how interesting information comes out. This action is repeated many times in the design process.

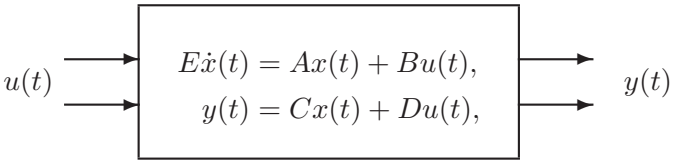

Figure 1.1: Input-output correspondence

Of course, a non-linear system may be generated. For mechanical systems, the resulting differential equations are usually of second order. If, in addition, discretization with respect to the time variable is performed, one derives difference equations. Such cases are beyond the aim of this thesis.

With modern computers, obtaining a numerical solution appears to be simple. But it is not as easy as it seems to be. In industry, for many reasons such as the manufacture cost and/or the users's convenience, one tends to integrate more components into tiny units. This results in the so-called micro-electro-mechanical-systems (MEMS), which are compositions of electric circuits and mechanical elements in micro scale. Simulation of such complex microsystems is complicated. On the one hand, knowledge about phenomena happening in normal scale cannot be applied to micro scale; one needs to take into account the effects that only occur in small scale. On the other hand, in order to understand the relation between different parts of the system, all of them must be simulated in the dynamics of interaction.

Let us consider a solid propellant microthruster [104, 145, 150] as an instance of electro-thermal MEMS. It is used to produce propulsion for nanospacecrafts, microrockets, and microsatellites. A single microthruster is principally composed of four parts: the reservoir or the chamber, the igniter, the seal or the diaphragm, and the nozzle (see Figure 1.2 (left).) The solid fuel is contained in the chamber. The combustion is ignited by a resistor, which is heated by letting an electric current go through. This will increase pressure in the chamber and break the diaphragm when the pressure approaches the critical point. Gas generated from the combustion is led through a hole on the nozzle part and results in propulsion. This electro-thermal process is approximately modeled by

$$\nabla(\mathcal{K}\nabla T) + \frac{I^2 R}{V} - \rho C \frac{\partial T}{\partial t} = 0, \tag{1.2}$$

where $\mathcal{K}$ is the (isotropic) thermal conductivity, C is the heat conductivity, ρ is the mass density, T is the unknown temperature. Furthermore, I is the total current through the resistor, R and V are its total resistance and volume, respectively. Under some more mild assumptions [150], we can consider (1.2) linear. By the finite element method, the equation (1.2) is discretized as

$$\begin{aligned} E\dot{T}(t) &= AT(t) + b\frac{I^2(t)R}{V}, \\ y(t) &= CT(t), \end{aligned} \tag{1.3}$$

in which E, A, b, C are constant matrices and T is now the vector of temperature at grid points. In the above equations, one is not intersted in the temperature in the whole domain but only at some nodes, which is expressed through the function $y(t)$. By the demand of thorough examination, the order of the system (1.3) tends to be high, especially for 3D simulations. It easily reaches thousands or even millions. The situation becomes extreme when one wants to integrate numerous microthrusters into an array (see Figure 1.2 (right)) and model the operation of them simultaneously, not to mention the simulation of the circuit whose output is used to compute the input $I^2(t)R/V$.

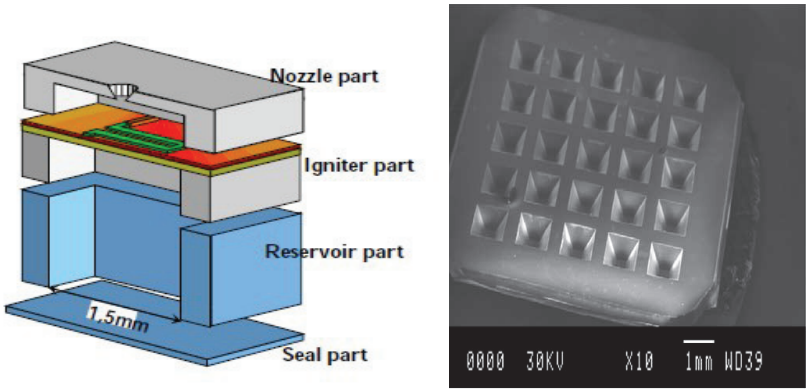

Figure 1.2: A single solid propellant microthruster and an integrated array [104, 146]

Electro-thermal MEMS simulation is not the only issue where one encounters large scale systems. Large systems may appear in many other fields such as simulation of computer microchips which may contain millions of details, data assimilation for weather forecast, modeling and simulation of microfluidic systems.

With such large data, simulation takes unaffordably long time. There obviously is a need to reduce the order of the mathematical models for these systems. More precisely, system (1.1) will be replaced with another system of the same form

$$\begin{aligned} \hat{E}\dot{\hat{x}}(t) &= \hat{A}\hat{x}(t) + \hat{B}u(t), \\ \hat{y}(t) &= \hat{C}\hat{x}(t) + Du(t), \end{aligned} \tag{1.4}$$

where $\hat{E}, \hat{A} \in \mathbb{R}^{r\times r}, \hat{B} \in \mathbb{R}^{r\times m}, \hat{C} \in \mathbb{R}^{l\times r}, r \ll N$ and (1.4) shares important properties with and approximates (1.1) in some sense. This task is widely known as *model order reduction* (MOR). Since the input-output coupling term D is not involved in MOR, we can omit this term from now on.

The MOR problem has been investigated for more than half a century and still attracts the attention of many applied mathematicians as well as engineers. Several methods have been proposed to solve various large scale problems in practice. The earliest known method is most probably Proper Orthogonal Decomposition. It started with the works of Kosambi, Loève and Karhunen in the middle of the last

century [96, 114, 93]. This method shares the same idea with Principal Component Analysis introduced by Pearson in 1901 [133]: using singular value decompositions (SVD), it extracts the most representative information from given data to construct the projection subspace. Also using SVD, balanced truncation was introduced in 1981 by Moore [121]. This method is based on the two notions: observability and reachability, which are of interest in control theory. Balanced truncation first finds a balancing transformation which balances the degree of observability and reachability and then truncates all states that have low degrees of those. In contrast with these two methods, the Krylov subspace method, which was likely to be first proposed for MOR in 1980s by Gragg and Villemagne [69, 173], requires no matrix decomposition. It constructs the reduced models that match some moments of the original transfer function about some point(s). A comprehensive description of MOR methods can be found in [10].

Each method has its own strength and weakness. Depending on the problem and the purpose, the user can choose a suitable one. However, all of them can be formulated under the Petrov-Galerkin projection framework. That is to seek two full-rank projection matrices $V, W \in \mathbb{R}^{N\times r}$ and then construct the reduced system as

$$\begin{aligned} W^T EV\dot{\hat{x}}(t) &= W^T AV\hat{x}(t) + W^T Bu(t), \\ \hat{y}(t) &= CV\hat{x}(t). \end{aligned}$$

There are, however, always new challenges in seemingly solved problems. Let us turn our attention back to the simulation of the solid propellant microthruster array mentioned above. The propulsion production of each individual thruster is independent of the others. It is controlled by an array of resistors which ignite the combustion of the solid propellant. During the combustion, the heat may transfer from one thruster to its neighbors and result in unwanted ignition. On the other hand, the loss of heat, besides to its neighbors, to the outside which is inconsiderable in normal scale, may stop the combustion if it exceeds the provided heat. Therefore, the integration of microthrusters into an array requires a thorough temperature control, specially the heat exchange.

Mathematically, the heat exchange is modeled by convection boundary condition, namely

$$\frac{\partial T}{\partial n} = \sigma(T - T_b), \tag{1.5}$$

where T_b is the bulk temperature and σ is the convection coefficient depending on the surroundings. It is noteworthy that the discretization of (1.5) contributes to the formulation of matrix A in (1.3). Due to the structure of the integrated thruster array, the heat flux through the wall is different from part to part. That means, (1.5) is imposed but with different σ for different parts of the boundary. As a

consequence, (1.3) becomes

$$\begin{aligned} E\dot{T}(t) &= \Big(A_0 + \sum_{i=1}^{k} \sigma_i A_i\Big)T(t) + b\frac{I^2(t)R}{V}, \\ y(t) &= CT(t), \end{aligned}$$

where σ_i can vary during the use of the model. Conventional MOR methods are only applicable for fixed σ_i. Each time one of them changes, the model reduction must be performed again. This solution has two disadvantages. First, the repetition of the computation will multiply the consumed time which should be avoided in modeling and simulation. Second, the outcome model is not convenient to the users. Certainly, the users, who are often engineers, do not want to learn and use model reduction; they prefer a compact model which can be adapted to the change of parameters easily and directly. This fact raises a desire to reduce the order of parameter-dependent systems (PDS)

$$\begin{aligned} E(p)\dot{x}(t;p) &= A(p)x(t;p) + B(p)u(t), \\ y(t;p) &= C(p)x(t;p), \end{aligned} \tag{1.6}$$

where $p \in \Omega \subset \mathbb{R}^d$, while still preserving the dependence on the parameters p. The state and the output of this system depend on, in addition to the time variable t, the parameters p. However, for the sake of simplification of notations, henceforth we will merely denote them as functions of time. The problem is widely known as *parametric model order reduction* (PMOR). Dealing with this problem, the aim of this thesis is to seek either (I) a parameter-dependent reduced order model for (1.6),

$$\begin{aligned} \hat{E}(p)\dot{\hat{x}}(t) &= \hat{A}(p)\hat{x}(t) + \hat{B}(p)u(t), \\ \hat{y}(t) &= \hat{C}(p)\hat{x}(t), \end{aligned} \tag{1.7}$$

where $p \in \Omega \subset \mathbb{R}^d$ or (II) a procedure to produce a reduced model of the form (1.7) for any $p \in \Omega$. One can observe that a method in trend I can be used as a method in trend II. However, the converse is, in general, not available.

If $E(p)$ is non-singular for all p, the differential algebraic equation (DAE) in (1.6) is an ordinary differential equation (ODE) and can be written as

$$\begin{aligned} \dot{x}(t;p) &= \tilde{A}(p)x(t;p) + \tilde{B}(p)u(t), \\ y(t;p) &= \tilde{C}(p)x(t;p). \end{aligned} \tag{1.8}$$

The system (1.8) is also viewed as a special case of (1.6), where $E(p) = I$. In fact, depending on the applications, while some approaches are devoted to systems whose input-to-state equations are DAEs, the others are developed only for systems that are composed of ODEs. And in order to keep tracking the original results, the presentation will be shifted between two kinds of these systems for different approaches. Moreover, whenever DAEs as in (1.6) are involved, we always assume that the corresponding matrix pencil (A, E) is regular for all p.

1.2 Overview of Existing Approaches

Undoubtedly, reduction of the order of PDSs must be based on usual MOR methods, i.e., methods for parameter-independent systems. The first effort to deal with PMOR was proposed in [176]. It was then followed by various extensions and developments, e.g., [43, 55, 110]. The core of these approaches is based on matching the so-called generalized moments by projecting the original system on either a union of standard Krylov subspaces or newly defined generalized Krylov subspaces. They are therefore referred to as Krylov subspace based methods.

The method proposed in [16, 17] combined balanced truncation with interpolation. It starts with constructing reduced transfer functions at given grid points in the parameter domain. Then, the reduced transfer function on the whole domain is derived by interpolation using Lagrange, Hermite and sinc functions.

Also using interpolation, the authors of [130] chose to interpolate the reduced system matrices. Following this direction, they succeeded in finding a way to sum up different reduced state vectors that have the same physical meaning.

The aforementioned methods follow the trend I: symbolically preserve the dependence on parameters. These solutions give theoretically compact models and are convenient to the users. However, one can easily observe that (1.7) will be utilized in a computational manner, i.e., given $p \in \Omega$, one has to compute system matrices of (1.7) in order to really be able to perform further steps. Forming an explicit expression for the reduced transfer function like (1.7) may encounter some technical difficulties. Instead of this, one may provide an algorithm that allows the computation of (1.7) for each given p. The important point is to achieve an algorithm to compute the reduced system in *real time*. In MOR context, an algorithm is considered to be usable in real time if its computational complexity is independent of the original order.

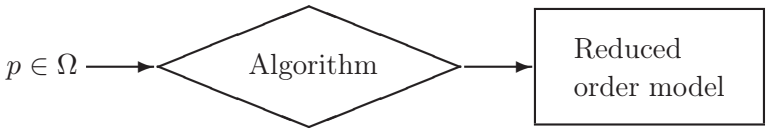

Figure 1.3: Trend II in solutions to MOR of PDSs

Following the work in [7], the authors of [6, 45, 8] make use of the interpolation of reduced system matrices in another style. They observed that the original system matrices usually own some special properties such as symmetric positive definiteness or non-singularity. The reduced models constructed at a parameter sample (should) inherit these properties. The direct interpolation of such matrices, in general, does not preserve such properties and therefore may lead to a completely meaningless reduced system. The key observation is that such structured matrices belong to some differential manifolds. The interpolation generally cannot be carried out on

manifolds, but it can be approximated by interpolating on tangent spaces to this manifold. To this end, one has to invoke a logarithmic mapping to map data from the manifold to a tangent space such that one can manipulate them as in a vector space. Then, the interpolated data are mapped back to the manifold by an exponential mapping. In comparison with the interpolation based methods mentioned above, the procedure of this approach is rather complicated. This is the reason why an explicit expression for the reduced systems of the form (1.7) is not available. The approach is therefore classified to be in the trend II of the solutions to the order reduction of PDSs.

1.3 Contribution and Outline of Thesis

Inspired by the work of Baur and Benner [16, 17], our first result combines the use of balanced truncation and spline interpolation for PMOR. Splines have been widely used in science and mathematics [26]. Their advantage is that they require only low-degree polynomials while it still yields a quite smooth approximation. The application of linear splines to the problem is almost straightforward due to the simplicity. However, invoking cubic splines, which gives better approximation, faces some obstacles. In order to derive an error bound for the method and construct a state space representation for the reduced system, which has been computed by interpolation of reduced transfer function, we have to choose an appropriate end condition and estimate as well as prove an upper bound for the norm of the inverse of the collocation matrix. Besides showing that the stability is preserved in the proposed method, we also give a hint, using the derived error bound, on which cases this method should not be applied.

Our second result, belonging to the trend II, develops an algorithm that computes the reduced order model based on a set of pre-computed projection subspaces. This is done through interpolation of these subspaces. In [7], Amsallem and Farhat suggested interpolating projection subspaces which have been computed at some chosen parameter values. This article, to our knowledge, for the first time proposed the framework of interpolation on a manifold of structured matrices for MOR of PDSs. However, one cannot use it in real time, since the computational complexity depends on the original order. In [6, 45, 8], the online computation issue was addressed but the approach used is interpolation of reduced system matrices, not the projection subspaces. By exploiting the formulations of the logarithmic and exponential mappings on Grassmann manifold, developing a strategy to structure the SVD of sums of SVDs, and by appropriately decomposing the computation procedure into offline and online stage, we propose an improved version of the algorithm that allows real time computation.

The thesis is organized as follows. In Chapter 2, we spend the first section to recall some basic facts of linear dynamical systems. Some widely used MOR methods are presented in detail in order to prepare the reader for a deep understanding on MOR. This chapter also provides a clear explanation for the Riemann structure of

Grassmann manifolds.

Chapter 3 summarizes existing approaches for MOR of PDSs. It starts with a systematic presentation of Krylov subspace based methods. We first collect and give a unified generalization of results in [70, 176, 43] and then show other approaches in this framework. All methods based on interpolation are presented as well: the interpolation of reduced transfer function, the direct and indirect interpolation of system matrices. Besides explanation, the strength and the weakness of each method are also analyzed.

Our main results are presented in Chapter 4. The spline interpolation based method is given in the first part. The method is then tested with a PDS resulting from the discretization of a convection diffusion equation. In the second part, we present a real time procedure for producing reduced order models using interpolation on Grassmann manifolds. The effectiveness of our method is then illustrated through a numerical example with a real-world model.

Finally, the conclusion, possible improvements and open problems are given in Chapter 5.

Chapter 2

Preliminaries

Contents

The first section of this chapter gives an overview of dynamical systems. This includes general definitions, properties that help to have an insight into MOR methods presented later on. In the second section, some common MOR methods for dealing with (parameter-independent) large-scale systems are presented. They are the basis on which MOR methods for PDSs will be built in the succeeding chapters. In the third section, we provide basic facts on some manifolds in linear algebra.

2.1 Brief Theory of Dynamical Systems

There are various textbooks on this subject, e.g., [163, 31, 39, 65, 10, 84]. This part is written, however, mainly based on the last three ones.

2.1.1 Mathematical Formulation of Dynamical Systems

Dynamical systems is a term for all *systems*, in which the present state of some components is determined by not only the present but also past state of other components [125]. Many phenomena in science and life can be modeled as dynamical systems. They appear frequently in physics, economics, biology, mechanics and transportations [1].

In order to describe a dynamical system by a mathematical model, all its components must be characterized by quantities, which are functions of time and called *variables*. Depending on their functionalities in the system, they are divided into *external variables* and *internal variables* [84]. External variables quantify the relationship between the system and the surroundings. They are, again, categorized as either *input* $u(t)$ which represents the effect of the outer objects on the system or *output* $y(t)$ which describes the influence of the system on the outer world. We will only consider the *controlled inputs and measured outputs* in this dissertation. Internal variables or the *state* $x(t)$, on the other hand, stands for the state of the system.

Before approaching the definition, we need to specify some other concepts. The first one is the *time domain*. It is the set of time instants T within which the act of the system is examined. The time domain may be a discrete set $T = \mathbb{N}, \mathbb{Z}$ or a continuous set $T = \mathbb{R}_+$, the set of positive real numbers $\mathbb{R}$. The resulting dynamical system will be said to be *discrete-time* or *continuous-time*, respectively. The sets U, Y, X that the input $u(t)$, the output $y(t)$ and the state $x(t)$ vary on are called the *input value space, output value space, state space*, respectively. They are usually subsets of some Cartesian product of $\mathbb{R}$ or the set of complex numbers $\mathbb{C}$. In addition, only input functions $u(t)$ in some set of functions $\mathcal{U}$ are accepted. This set is referred to as the *input function space*. The set of functions from T to U is denoted by U^T. We follow [84] to define dynamical systems and the related concepts.

Definition 2.1 *A dynamical system is a structure* $\Sigma := (T, U, \mathcal{U}, X, Y, \varphi, \eta)$*, where* $T, U, \mathcal{U}, X, Y$ *are non-empty sets,* $T \subset \mathbb{R}$*,* $\mathcal{U} \subset U^T$*,* $\varphi : D_\varphi \subset T^2 \times X \times \mathcal{U} \longrightarrow X, \eta : T \times X \times U \longrightarrow Y$ *such that the following properties are satisfied:*

Interval property: *For each* $t_0 \in T, x_0 \in X, u(\cdot) \in \mathcal{U}$*, the life span of* $\varphi(\cdot, t_0, x_0, u(\cdot))$*,*

$$T_{t_0,x_0,u(\cdot)} := \{t \in T \mid (t; t_0, x_0, u(\cdot)) \in D_\varphi\},$$

is an interval in T *containing* t_0*.*

Consistency property: *For each* $t_0 \in T, x_0 \in X, u(\cdot) \in \mathcal{U}$

$$\varphi(t_0; t_0, x_0, u(\cdot)) = x_0.$$

[1] Unlike [84], where *dynamical system* stands for a mathematical model, we use this term (or sometimes only *system*) for systems in general. The mathematical model of a system will be referred to as a *model*. Nevertheless, *system* is also used to name a system of linear equations or differential equations. Such uses, we suppose, do not cause any serious ambiguity.

Causality property: *For all* $t_0 \in T, x_0 \in X, u(\cdot), v(\cdot) \in \mathcal{U}$, $t_1 \in T_{t_0,x_0,u(\cdot)} \cap T_{t_0,x_0,v(\cdot)}$

$$\Big(u(t) = v(t)\ \forall t \in [t_0, t_1)\Big) \Rightarrow \varphi(t_1; t_0, x_0, u(\cdot)) = \varphi(t_1; t_0, x_0, v(\cdot)).$$

Cocycle property: *If* $t_1 \in T_{t_0,x_0,u(\cdot)}$ *and* $x_1 = \varphi(t_1; t_0, x_0, u(\cdot))$ *for some* $t_0 \in T, x_0 \in X, u(\cdot) \in \mathcal{U}$, *then* $T_{t_1,x_1,u(\cdot)} \subset T_{t_0,x_0,u(\cdot)}$ *and*

$$\varphi(t; t_0, x_0, u(\cdot)) = \varphi(t; t_1, x_1, u(\cdot)),\ t \in T_{t_1,x_1,u(\cdot)}.$$

φ *is called the state transition map,* η *the output map,* $T \times X$ *the event space,* $x(t) = \varphi(t; t_0, x_0, u(\cdot))$ *the trajectory of* Σ *determined by initial* $x(t_0) = x_0$ *and control* $u(\cdot)$, *its graph* $\varphi(t; t_0, x_0, u(\cdot)), t \in T_{t_0,x_0,u(\cdot)}$ *an orbit of* Σ.

We consider only real dynamical systems, i.e., $X \subset \mathbb{R}^N$,$\mathcal{U} = \{f : T \to U \subset \mathbb{R}^m, Y \subset \mathbb{R}^l\}$; specification will be made for the situations that relate to the set of complex number $\mathbb{C}$ whenever it is necessary to avoid confusion.

Definition 2.2 *The dynamical system* Σ *is said to be time-invariant if the following requirements are fulfilled*

i) $0 \in T \subset \mathbb{R}$ *and* $T + T \subset T$.

ii) $\mathcal{U}$ *is invariant under shift with arbitrary length* $0 \leq \tau \in T$, *i.e.,* $u(t) \in \mathcal{U}$ *implies* $u(t - \tau) \in \mathcal{U}$.

iii) For all $t_0, t, \tau \in T, \tau \geq 0$ *and* $x_0 \in X, u(\cdot) \in \mathcal{U}$

$$\varphi(t + \tau; t_0 + \tau, x_0, u(\cdot - \tau)) = \varphi(t, t_0, x_0, u(\cdot)).$$

iv) Output map η *does not depend on time.*

Definition 2.3 *The dynamical system* Σ *is said to be linear if*

i) U,$\mathcal{U}, X, Y$ *are vector spaces on* $\mathbb{R}$,

ii) for all $t, t_0 \in T, t \geq t_0$, *the mappings*

$$\varphi(t; t_0, \cdot, \cdot) : X \times U \longrightarrow X \text{ and } \eta(t, \cdot, \cdot) : X \times U \longrightarrow Y$$

are linear.

An important property of linear systems is that the *superposition principle* holds for the state transition map φ (and output map η), i.e.,

$$\varphi(t; t_0, x_0, u(\cdot)) = \varphi(t; t_0, x_0, 0) + \varphi(t; t_0, 0, u(\cdot)).$$

In words, each trajectory of linear systems is the sum of the *free motion* $\varphi(t; t_0, x_0, 0)$ and the *forced motion* $\varphi(t; t_0, 0, u(\cdot))$. It also implies that without control $u(\cdot)$, 0 is an equilibrium point of every linear system.

Definition 2.4 *The dynamical system Σ is said to be differentiable if the following conditions hold*

i) $T \subset \mathbb{R}$ *is an open interval.*

ii) $U \subset \mathbb{R}^m, Y \subset \mathbb{R}^l$, *and* X *is an open subset in* $\mathbb{R}^N$.

iii) There exists a function $f : T \times X \times U \longrightarrow \mathbb{R}^N$ *such that for all* $t_0 \in T, x_0 \in X, u(\cdot) \in \mathcal{U}$, *the initial value problem*

$$\begin{aligned} \dot{x}(t) &= f(t, x(t), u(t)), t \geq t_0, t \in T, \\ x(t_0) &= x_0 \end{aligned}$$

has a unique solution $x(\cdot)$ *on the maximal open time interval* $T_{t_0,x_0,u(\cdot)}$ *and* $x(t) = \varphi(t; t_0, \cdot, \cdot), t \in T_{t_0,x_0,u(\cdot)}$.

iv) $\eta : T \times X \times U \longrightarrow Y$ *is continuous.*

Remark A nonlinear differentiable dynamical system

$$\begin{aligned} \dot{x}(t) &= f(t, x(t), u(t)), t \in T, \\ y(t) &= \eta(x(t), u(t)), \end{aligned} \tag{2.1}$$

where $T \subset \mathbb{R}, U \subset \mathbb{R}^m, X \subset \mathbb{R}^N$ are open, $Y \subset \mathbb{R}^l$, $\mathcal{U} = \mathcal{C}(T, U)$ - the set of continuous functions from T to U - can be linearized in a neighborhood close to a given trajectory as follows. Let $z(t) = \varphi(t; t_0, z_0, v(\cdot))$. Accordingly

$$\begin{aligned} \dot{z}(t) &= f(t, z(t), v(t)), t_0 < t \in T, \\ z(t_0) &= z_0. \end{aligned}$$

Let us assume moreover that φ and η are continuously differentiable with respect to (x, u). Denote by

$$\begin{aligned} A(t) &= \Big(\frac{\partial f_i}{\partial x_j}\big(t, z(t), v(t)\big)\Big)_{ij} \in \mathbb{R}^{N\times N}, \\ B(t) &= \Big(\frac{\partial f_i}{\partial u_j}\big(t, z(t), v(t)\big)\Big)_{ij} \in \mathbb{R}^{N\times m}, \\ C(t) &= \Big(\frac{\partial \eta_i}{\partial x_j}\big(t, z(t), v(t)\big)\Big)_{ij} \in \mathbb{R}^{l\times N}, \\ D(t) &= \Big(\frac{\partial \eta_i}{\partial u_j}\big(t, z(t), v(t)\big)\Big)_{ij} \in \mathbb{R}^{l\times m}. \end{aligned}$$

It is shown in [84] that the solution of the state equation of

$$\begin{aligned} \dot{x}(t) &= A(t)x(t) + B(t)u(t), \\ y(t) &= C(t)x(t) + D(t)u(t) \end{aligned} \tag{2.2}$$

is a first order approximation to that of (2.1). Hence, (2.2) is called the *linearization* of (2.1). □

In this thesis, we only consider continuous-time linear time-invariant (LTI) systems, which take the form

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + Du(t), \end{aligned} \tag{2.3}$$

where $A \in \mathbb{R}^{N\times N}, B \in \mathbb{R}^{N\times m}, C \in \mathbb{R}^{l\times N}, D \in \mathbb{R}^{l\times m}$. Some of the following arguments are still valid for descriptor systems, non-linear systems, or systems whose state equation is a second-order differential equation. Such cases will be specified clearly during the presentation. For the sake of brevity, we write $\Sigma = \left(\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right)$ for system (2.3). If $l = m = 1$, the system has only one input and one output. It is therefore called *single-input-single-output* (SISO) system, otherwise if $m > 1, l > 1$, it is called to be *multi-input-multi-output* (MIMO). Suppose that the state equation in (2.3) is coupled with the initial condition $x(t_0) = x_0$, by the *variation of constants*, the state $x(t)$ can be written as

$$x(t) = e^{A(t-t_0)}x_0 + \int_{t_0}^{t} e^{A(t-\tau)}Bu(\tau)d\tau, t \in \mathbb{R}.$$

The ultimate goal of the dissertation, however, involves systems that depend on parameters $p \in \Omega \subset \mathbb{R}^d$. These parameters are not the usual control input that only affects the second term in the state equation but appear in the whole system,

$$\begin{aligned} \dot{x}(t) &= A(p)x(t) + B(p)u(t), \\ y(t) &= C(p)x(t) + D(p)u(t). \end{aligned}$$

We also assume moreover that the system matrices depend, at least, continuously on the parameters.

2.1.2 Input-output Behavior Formulation

In many applications, one is only interested in the response of the system to the given inputs. It may also be the case in which the full state vector is not completely accessible. One has to define the system without the presence of the state. Assume that $T = \mathbb{R}_+, t_0 = 0, x_0 = 0$. Then the output of (2.3) associated with input $u(\cdot)$ is

$$y(t) = Du(t) + \int_{0}^{t} Ce^{A(t-\tau)}Bu(\tau)d\tau.$$

Recall that the Dirac delta function is a generalized function satisfying

$$\delta(x) = \begin{cases} +\infty, x = 0, \\ 0, x \neq 0. \end{cases} \qquad \text{such that} \quad \int_{-\infty}^{\infty} \delta(x)dx = 1.$$

The output $y(t)$ can be rewritten as

$$\begin{aligned} y(t) &= \int_0^t D\delta(\tau - t)u(\tau)d\tau + \int_0^t Ce^{A(t-\tau)}Bu(\tau)d\tau \\ &= \int_0^t D\delta(t - \tau)u(\tau)d\tau + \int_0^t Ce^{A(t-\tau)}Bu(\tau)d\tau \\ &= \int_0^t \left(D\delta(t-\tau) + \int_0^t Ce^{A(t-\tau)}Bu \right) u(\tau)d\tau \\ &= (G * u)(t), \end{aligned} \tag{2.4}$$

where $G(t) = D\delta(t) + Ce^{At}B$. Since $G(t)$ is the response of system Σ to the impulse δ, it is called the *impulse response*. Accordingly, we define

$$\begin{array}{rcl} L : \mathcal{L}^q(\mathbb{R}_+, \mathbb{R}^m) & \longrightarrow & \mathcal{L}^q(\mathbb{R}_+, \mathbb{R}^l), 1 \leq q \leq \infty \\ u & \longmapsto & y(t) = Du(t) + \int_0^t Ce^{A(t-\tau)}Bu(\tau)d\tau, \end{array} \tag{2.5}$$

where $\mathcal{L}^q(\mathbb{R}_+, \mathbb{R}^n) := \left\{ f : \mathbb{R}_+ \to \mathbb{R}^n, \left(\int_{\mathbb{R}_+} \|f(t)\|_q^q dt \right)^{\frac{1}{q}} < \infty \right\}$. L is called the *input-output operator*. This approach is referred to as the *input-output behavior approach in time domain*, since only functions of time are involved. It is demonstrated [84] that if A is a Hurwitz matrix, i.e., $Re(\lambda) < 0, \forall \lambda \in \Lambda(A)$ - the set of eigenvalues of A, L is a bounded linear operator from $\mathcal{L}^q(\mathbb{R}_+, \mathbb{R}^m)$ to $\mathcal{L}^q(\mathbb{R}_+, \mathbb{R}^l), q \geq 1$. In that case, L is said to be $\mathcal{L}^q$*-stable*. Accordingly, its corresponding system in the state space form and its transfer function, defined later, are also said to be stable. The eigenvalues of A are sometimes referred to as the *poles* of the system. With this property, one can define L on $T = \mathbb{R}$ and with any initial time t_0 and initial condition x_0. The details can be found in [84].

The analysis of the input-output behavior in the frequency domain is derived from applying Laplace transform to L in $\mathcal{L}^1(\mathbb{R}_+, \mathbb{R}^n)$.

Definition 2.5 *Let $f(t) \in \mathcal{L}^1(\mathbb{R}_+, \mathbb{R}^n)$, the Laplace transform of f is*

$$\hat{f}(s) = (\mathcal{L}f)(s) := \int_0^\infty f(t)e^{-st}dt, s \in \mathbb{C}.$$

The crucial advantage of using Laplace transform is that it turns the convolution into the normal product of two functions. Note that the integral does not always converge. This happens when $f(t)e^{-\alpha t} \in \mathcal{L}^1(\mathbb{R}_+, \mathbb{R}^n)$, where $Re(s) \geq \alpha$. Taking Laplace transform on both sides of (2.4) yields

$$\hat{y}(s) = \hat{G}(s)\hat{u}(s). \tag{2.6}$$

In frequency domain, $\hat{G}(s)$ allows us to determine the system's output directly through the usual product with the input. It is called the *transfer function* of system (2.3).

Another way to formulate the transfer function of system (2.3) is to take the Laplace transform directly on both sides of this system

$$s\hat{x}(s) = A\hat{x}(s) + B\hat{u}(s),$$
$$\hat{y} = C\hat{x}(s) + D\hat{u}(s).$$

Thus,

$$\hat{y}(s) = (D + C(sI - A)^{-1}B)\hat{u}(s) =: H(s)\hat{u}(s). \tag{2.7}$$

Comparing (2.6) and (2.7), one has $\hat{G}(s) = H(s) = D + C(sI - A)^{-1}B$.

Let ϕ be a coordinate transformation from x to $\tilde{x}$ in the state space X, i.e., $x = \phi\tilde{x}$. The state space description of system Σ becomes $\tilde{\Sigma} = \left(\begin{array}{c|c} \tilde{A} & \tilde{B} \\ \hline \tilde{C} & \tilde{D} \end{array}\right)$, where $\tilde{A} = \phi^{-1}A\phi, \tilde{B} = \phi^{-1}B, \tilde{C} = C\phi, \tilde{D} = D$. By some simple computations, it follows that

$$\tilde{H}(s) \equiv H(s).$$

That is, transfer functions are basis independent.

2.1.3 Reachability and Observability

The structure of the system $\Sigma = \left(\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right)$ may be such that not all states of the state space X can be reached from a fixed state with a reasonable control. The reachable ones constitute a subspace of X. We formulate this concept in the following

Definition 2.6

- *A state $\bar{x} \in X$ is said to be reachable from zero if there exists a finite energy control $u(\cdot) \in \mathcal{U}$, a finite time $\bar{t}$ such that*

$$\bar{x} = \varphi(\bar{t}; t_0, 0, u(\cdot)).$$

- *The reachable subspace $X^r \subset X$ is defined as the set of all reachable states.*
- *System Σ is said to be reachable if $X^r = X$.*
- *The infinite dimensional matrix*

$$\mathcal{R}(A, B) := [B \ AB \ A^2B \ \cdots]$$

is called the reachability matrix of Σ.

The phrase "finite energy" related to control $u(\cdot)$ means that u has a finite energy norm with which $\mathcal{U}$ is equipped. Usually, the standard norm of $\mathcal{L}^2(\mathbb{R}_+, \mathbb{R}^m)$ is used for $\mathcal{U} = \mathcal{L}^2(\mathbb{R}_+, \mathbb{R}^m)$.

The above definition involves only the pair (A, B) of Σ. We however want to attach this concept to a concrete dynamical system.

The reachability matrix has a close relationship with the *reachability gramian* defined as follow.

Definition 2.7 *The finite reachability gramian at time $t \in \mathbb{R}_+$ of the system $\Sigma = \left(\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right)$ is the matrix*

$$\mathcal{P}(t) := \int_0^t e^{A\tau} B B^T e^{A^T \tau} d\tau,$$

where letter ^{T} means the matrix transpose.

It is shown that the following statements hold.

Theorem 2.1 *([10], Proposition 4.10)*

- $\mathcal{P}(t) = \mathcal{P}^T(t)$ *and is positive semi-definite.*
- *For all* $t \in \mathbb{R}_+, \mathrm{Im}\mathcal{P}(t) = \mathrm{Im}\mathcal{R}(A, B)$.

Based on Theorem 2.1, the pivotal result and its consequence are given.

Theorem 2.2 *([10], Theorem 4.7)*

- $X^r = \mathrm{Im}\mathcal{R}(A, B)$.
- X^r *is an* A*-invariant subspace, i.e.,* $AX^r \subset X^r$.
- Σ *is reachable iff* $\mathrm{rank}(\mathcal{R}(A, B)) = N$.
- X^r *is invariant under coordinate transformations.*

Taking the Cayley-Hamilton theorem into account, the rank of $\mathcal{R}(A, B)$ is determined by $\{A^i B, i = 0, \cdots, N-1\}$.

By Theorems 2.1 and 2.2, $\forall\ \bar{x} \in X^r, \forall\ \bar{t} \in \mathbb{R}_+, \exists\ \bar{\xi}$ such that $\bar{x} = \mathcal{P}(\bar{t})\bar{\xi}$. Then

$$\bar{u}(t) = B^T e^{A^T(\bar{t}-t)} \bar{\xi}$$

is a control that drives 0 to $\bar{x}$ at time $\bar{t}$. It is shown in [10] that $\bar{u}(t)$ has minimal energy among all controls doing the same task, i.e., $\|\bar{u}\|_2 \leq \|u\|_2, \forall\ u(t) \in \mathcal{L}^2(\mathbb{R}_+, \mathbb{R}^m), \varphi(\bar{t} : 0, 0, u(\cdot)) = \bar{x}$. If Σ is reachable, by some simple symbolic computations, we have

$$\|\bar{u}\|_2^2 = \bar{x}^T \mathcal{P}(\bar{t})^{-1} \bar{x}. \tag{2.8}$$

The concept of observability derives from the state observation problem: given $y(t) = Cx(t)$ for some $t \in [t_1, t_1 + \tau]$, reconstruct state $x(t_1)$.

Definition 2.8

- *A state* $\bar{x} \in X$ *of* $\Sigma = \left(\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right)$ *is unobservable if* $y(t) = C\varphi(t; 0, \bar{x}, 0) = 0, \forall\ t \geq 0$.
- *The unobservable subspace* $X^{uo} \subset X$ *is defined as the set of all unobservable states of* Σ.

- *System Σ is said to be observable if $X^{uo} = \{0\}$.*
- *The infinite dimensional matrix*

$$\mathcal{O}(A, C) := [C^T \ A^T C^T \ (A^T)^2 C^T \ \cdots]^T$$

is called the observability matrix of Σ.

- *The finite observability gramian at $t \in \mathbb{R}_+$ is*

$$\mathcal{Q}(t) := \int_0^t e^{A^T \tau} C^T C e^{A\tau} d\tau.$$

In the following, we summarize properties of observability. They are, in a certain sense, the counterpart of that of reachability.

Theorem 2.3 *([10], Theorem 4.20)*

- *For all $t \in \mathbb{R}_+, X^{uo} = Ker\mathcal{O}(A, C) = Ker\mathcal{Q}(t)$.*
- *X^{uo} is A-invariant.*
- *Σ is observable iff* $\operatorname{rank}(\mathcal{O}(A, C)) = N$
- *Observability is basis independent.*

Similar to (2.8), the energy in $\mathcal{L}^2(\mathbb{R}_+, \mathbb{R}^l)$ of the output function $y(t) = Cx(t)$ caused by state $\bar{x}$ at time $\bar{t}$ is computed by

$$\|y\|^2 = \bar{x}^T \mathcal{Q}(\bar{t}) \bar{x}.$$

For the controllability and observability of second order LTI systems, one can see, e.g., in [115].

By definition, $\mathcal{P}$ and $\mathcal{Q}$ are non-decreasing in $\mathbb{R}_+$. If Σ is reachable, then $\mathcal{P}(t)$ is non-singular and its inverse $\mathcal{P}^{-1}(t)$ is non-increasing. If we fix a state $\bar{x}$ and take (2.8) into account, the longer the time the control $u(\cdot)$ needs to steer 0 to $\bar{x}$, the less energy it consumes. We deduce that the minimal energy for driving 0 to $\bar{x}$ at time $\bar{t}$ is attained as $\bar{t} \to \infty$. Likewise, the longer time the state $\bar{x}$ is active, the larger observation energy it produces. These facts raise the need to define *infinite gramians.*

Definition 2.9 *For a stable system* $\Sigma = \left(\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right)$*, the two (infinite) reachability gramian and observability gramian are defined as*

$$\mathcal{P} := \int_0^\infty e^{A\tau} B B^t e^{A^T \tau} d\tau, \tag{2.9}$$

$$\mathcal{Q} := \int_0^\infty e^{A^T \tau} C^T C e^{A\tau} d\tau. \tag{2.10}$$

Instead of explicit formulations, computation of reachability gramian and observability gramian are usually based on the following result.

Theorem 2.4 *([10], Proposition 4.27) Reachability gramian and observability gramian of the stable system Σ are solutions of Lyapunov equations:*

$$A\mathcal{P} + \mathcal{P}A^T + BB^T = 0, \tag{2.11}$$
$$A^T\mathcal{Q} + \mathcal{Q}A + C^TC = 0. \tag{2.12}$$

By the above arguments and notations, the minimal energy for reaching $\bar{x}$ from 0 is

$$\bar{x}\mathcal{P}^{-1}\bar{x}; \tag{2.13}$$

the largest observation energy produced by $\bar{x}$ is

$$\bar{x}\mathcal{Q}\bar{x}. \tag{2.14}$$

We consider next how a coordinate transformation $x = \phi\tilde{x}$ can affect the gramians.

$$\begin{aligned}\tilde{\mathcal{P}} &= \int_0^\infty e^{\tilde{A}\tau}\tilde{B}\tilde{B}^T e^{\tilde{A}^T\tau}d\tau \\ &= \int_0^\infty e^{\phi^{-1}A\phi\tau}\phi^{-1}BB^T\phi^{-T}e^{\phi^T A^T\phi^{-T}\tau}d\tau \\ &= \phi^{-1}\mathcal{P}\phi^{-T}.\end{aligned} \tag{2.15}$$

Likewise,

$$\tilde{\mathcal{Q}} = \phi^T\mathcal{Q}\phi. \tag{2.16}$$

(2.15) and (2.16) lead to an important observation: eigenvalues of $\mathcal{PQ}$ are invariant under coordinate transformations of the state space.

2.1.4 Norms of Systems

In order to quantify dynamical systems, especially the quality of approximation methods, norms of systems must be defined. Due to the diversity of purposes and situations, various norms were proposed. First of all, we would like to recall here the Schatten norm of matrices. Let $A \in \mathbb{R}^{l\times m}, m \le l$, denote by $\sigma_1(A) \ge \sigma_2(A) \ge \cdots \ge \sigma_m(A)$ the singular values [68, 46] of A. Then

$$\|A\|_{S,p} := \begin{cases} \left(\sum_{i=1}^m \sigma_i^p(A)\right)^{\frac{1}{p}}, 1 \le p < \infty, \\ \sigma_1(A), p = \infty. \end{cases} \tag{2.17}$$

The Schatten norm for $p = 2$ is also called the *Frobenius norm* and is equal to the *trace norm*

$$\|A\|_{S,2} = \|A\|_F := \left(\sum_{i=1}^m \sigma_i^2(A)\right)^{\frac{1}{2}} = (\mathrm{trace}(A^*A))^{\frac{1}{2}} \tag{2.18}$$

To define the *Hankel norm* of a system, we need to formulate its associated *Hankel operator*. Recall the formulation (2.5) of a linear system Σ, the Hankel operator $\mathcal{H}$ is defined as

$$\begin{aligned} \mathcal{H} : \mathcal{L}^2(\mathbb{R}_-, \mathbb{R}^m) &\longrightarrow \mathcal{L}^2(\mathbb{R}_+, \mathbb{R}^l) \\ u_- &\longmapsto y_+(t) := \int_{-\infty}^0 G(t-\tau)u_-(\tau)d\tau, t \geq 0. \end{aligned}$$

Definition 2.10 *The singular values of $\mathcal{H}, \sigma(\mathcal{H})$, are called the Hankel singular values of system Σ. The Hankel norm of the system Σ is defined as the induced $\mathcal{L}^2$-norm of its Hankel operator, i.e.,*

$$\|\Sigma\|_H := \|\mathcal{H}\|_{\mathcal{L}_2} = \sigma_{\max}(\mathcal{H}).$$

It was proven, e.g., in [10] that non-zero Hankel singular values of a reachable, observable and stable system are equal to the positive square roots of the eigenvalues of the product of the two gramians,

$$\sigma_i(\Sigma) = \sqrt{\lambda_i(\mathcal{PQ})}, i = 1, \cdots, m. \tag{2.19}$$

Therefore,

$$\|\Sigma\|_H = \sqrt{\lambda_{\max}(\mathcal{PQ})}.$$

Next, two frequently used norms for transfer functions will be defined through *Hardy spaces*.

Definition 2.11 *For functions $F : \mathbb{C}_+ \longrightarrow \mathbb{C}^{l\times m}$ analytic on the open right complex half-plane $\mathbb{C}_+$, the Hardy norm of F is*

$$\|F\|_{\mathcal{H}_p} := \begin{cases} \left(\sup\limits_{\alpha>0} \int_\infty^\infty \|F(\alpha+i\beta)\|_{S,p}^p dy\right)^{\frac{1}{p}}, 1 \leq p < \infty, \\ \sup\limits_{z\in\mathbb{C}_+} \|F(z)\|_{S,p}, \ p = \infty. \end{cases}$$

In the case $p = 2$ and $p = \infty$, taking (2.17) and (2.18) into account,

$$\begin{aligned} \|F\|_{\mathcal{H}_2} &= \left(\sup_{\alpha>0} \int_\infty^\infty \text{trace}\Big(F^*(\alpha-i\beta)F(\alpha+i\beta)\Big)d\beta\right)^{\frac{1}{2}}, \\ \|F\|_{\mathcal{H}_\infty} &= \sup_{z\in\mathbb{C}_+} \sigma_{\max}(F(z)). \end{aligned} \tag{2.20}$$

The Hardy space $\mathcal{H}_p(\mathbb{C}_+, \mathbb{C}^{l\times m})$ is defined as

$$\mathcal{H}_p(\mathbb{C}_+, \mathbb{C}^{l\times m}) := \{F : \mathbb{C}_+ \longrightarrow \mathbb{C}^{l\times m}, \|F\|_{\mathcal{H}_p} < \infty\}.$$

Remark By the *Maximum modulus theorem* [74], (2.20) turns into

$$\begin{aligned} \|F\|_{\mathcal{H}_2} &= \left(\int_{-\infty}^\infty \text{trace}\Big(F^*(-i\beta)F(i\beta)\Big)d\beta\right)^{\frac{1}{2}}, \\ \|F\|_{\mathcal{H}_\infty} &= \sup_{\beta\in\mathbb{R}} \sigma_{\max}(F(i\beta)). \end{aligned}$$

It is demonstrated that if a stable linear system $L : \mathcal{L}^2(\mathbb{R}, \mathbb{R}^m) \longrightarrow \mathcal{L}^2(\mathbb{R}, \mathbb{R}^l)$ has $F(s)$ as its transfer function, then the $\|\cdot\|_{\mathcal{H}_\infty}$ norm of $F(s)$ is equal to the induced $\mathcal{L}_2$ norm of L.

Another definition of $\mathcal{H}_2$-norm is given in [10]. There, one also can find a comparison between norms. □

2.2 MOR Methods

Recall that MOR involves the approximation of systems of the form (1.1) by a lower order system of the same form (1.4). For the sake of simplicity, we consider almost only the ordinary differential equations, i.e., $E = I$ and remove the input-output coupling term D. The special case will be specified during the presentation of the method.

2.2.1 Balanced Truncation

Balanced truncation was first proposed in [121] during the analysis of principal components of linear systems. It is a popular method because of the robustness, the guarantee of an error bound and the stability preservation. We follow the explanation in [10] to present the method.

In control theory, it is of importance to characterize the energy to reach a specified state. Recall the formula (2.13) that for a reachable, stable system the minimal energy required to reach $\bar{x}$ is $\bar{x}\mathcal{P}^{-1}\bar{x}$. Since $\mathcal{P}$ is symmetric positive definite, it allows an eigenvalue decomposition

$$\mathcal{P} = V^T \Delta V,$$

where V is an orthogonal matrix whose columns V_i are eigenvectors of $\mathcal{P}$ and $\Delta = \operatorname{diag}(d_1, \cdots, d_N)$, $d_1 \geq \cdots \geq d_N > 0$. Then

$$\mathcal{P}^{-1} = V^T \Delta^{-1} V,$$

in which, $\Delta^{-1} = \operatorname{diag}(d_1^{-1}, \cdots, d_N^{-1})$.

For any $\bar{x} \in X(= \mathbb{R}^N)$, assume that $\bar{x}$ has a linear representation through columns of V as

$$\bar{x} = \sum_{i=1}^{N} \alpha_i V_i.$$

The energy needed to reach $\bar{x}$ is

$$\begin{aligned}\bar{x}\mathcal{P}^{-1}\bar{x} &= \Big(\sum_{i=1}^{N} \alpha_i V_i\Big)^T \mathcal{P}^{-1} \Big(\sum_{j=1}^{N} \alpha_j V_j\Big) \\ &= \Big(\sum_{i=1}^{N} \alpha_i V_i\Big)^T \Big(\sum_{j=1}^{N} \alpha_j d_j^{-1} V_j\Big) = \sum_{i=1}^{N} \alpha_i^2 d_i^{-1}.\end{aligned}$$

It turns out that the states having large (significant) components in the subspace spanned by eigenvectors of $\mathcal{P}$ associated with large eigenvalues require small energy to be reached, referred to as *easy to reach*, and conversely, the states having large (significant) components in the subspace spanned by eigenvectors of $\mathcal{P}$ associated with small eigenvalues require large energy to be reached, referred to as *difficult to reach*. Likewise for an observable, stable system, by (2.14), the states having large components in the subspace spanned by eigenvectors of $\mathcal{Q}$ associated with large (small) eigenvalues will produce large (small) observation energy, referred to as *easy (difficult) to observe*.

The above analysis provides an efficient way to quantify the *degree of reachability* and the *degree of observability*. The states that are easy to reach and the states that are easy to observe play dominant roles in the behavior of the system. The others do not have much contribution to the system and are of less importance. They are good candidates to be cut off in order to make the order of the system smaller without considerably affecting the system's behavior. However, the problem is that degree of reachability and degree of observability of the states are two independent concepts. A dilemma can, therefore, happen that a state, which is difficult to reach (prefers to be truncated), is easy to observe (prefers to be retained) and vice versa. Consider the system $\Sigma = \left(\begin{array}{c|c} A & b \\ \hline c & \end{array}\right)$ where

$$A = \begin{bmatrix} -2 & -3 \\ 1 & 1 \end{bmatrix}, b = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, c = \begin{bmatrix} 0 & 1 \end{bmatrix}.$$

Its two gramians are

$$\mathcal{P} = \begin{bmatrix} 2.5 & -1.5 \\ -1.5 & 1 \end{bmatrix}, \mathcal{Q} = \begin{bmatrix} 0.5 & 1 \\ 1 & 2.5 \end{bmatrix}.$$

Eigenvalues and eigenvectors of $\mathcal{P}$, which are given in only four decimal digits, are

$$V = \begin{bmatrix} -0.5257 & -0.8507 \\ -0.8507 & 0.5257 \end{bmatrix}, D = \begin{bmatrix} 0.0729 & 0 \\ 0 & 3.4271 \end{bmatrix}.$$

The first eigenvector of $\mathcal{P}$, $V(:,1)$, corresponds to the small eigenvalue and is therefore difficult to reach. Conversely, the second eigenvector $V(:,2)$ is easy to reach. Now, we compute the observability energy these vectors produce.

$$V(:,1)^T \mathcal{Q} V(:,1) = 2.8416 \quad \text{and} \quad V(:,2)^T \mathcal{Q} V(:,2) = 0.1584.$$

It turns out that $V(:,1)$ is easy to observe while $V(:,2)$ is difficult to observe.

To cope with this, one has to find a basis, if it exists, for the state space on which, the degree of reachability and the degree of observability are balanced. More precisely,

$$\mathcal{P} = \mathcal{Q} = \Lambda = \text{diag}(\sigma_1, \cdots, \sigma_N). \tag{2.21}$$

It is worth to note that, by (2.19), if (2.21) holds, the diagonal elements are nothing else but the Hankel singular values of the system.

Definition 2.12 *A reachable, observable, stable system is called to be balanced if* $\mathcal{P} = \mathcal{Q}$ *and principal-axis balanced if* (2.21) *holds.*

The coordinate transformation that converts a reachable, observable, and stable system into principal-axis balanced form is called a balancing transformation.

The following lemma gives the answers to both questions: the existence of a balancing transformation and its formulation. The proof is due to some direct computations.

Lemma 2.5 *(e.g., [10], Lemma 7.3) Suppose that* $\mathcal{P}, \mathcal{Q}$ *are reachability and observability gramians of a reachable, observable, stable system. Then a balancing transformation is*

$$\phi = UK\Lambda^{-\frac{1}{2}} \quad \text{and} \quad \phi^{-1} = \Lambda^{\frac{1}{2}} K^T U^{-1}, \tag{2.22}$$

where $\mathcal{P} = UU^T, U^T\mathcal{Q}U = K\Lambda^2 K^T$ *are the Cholesky factorization of* $\mathcal{P}$ *and the eigenvalue decomposition of* $U^T\mathcal{Q}U$.

It is demonstrated that if Hankel singular values are pairwise distinct, the balancing transformation is unique up to a factor $S = \text{diag}(\pm 1, \cdots, \pm 1)$. Otherwise, instead of S, the factor is a block diagonal matrix whose blocks are orthogonal matrices.

It remains to apply Lemma 2.5 to reduce the reachable, observable, stable system $\Sigma = \left(\begin{array}{c|c} A & B \\ \hline C & \end{array}\right)$. Suppose that Σ has been set in a balancing coordinate, i.e., $\mathcal{P} = \mathcal{Q} = \Lambda$. Decompose the system matrices as

$$\Lambda = \begin{bmatrix} \Lambda_1 & 0 \\ 0 & \Lambda_2 \end{bmatrix}, A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, B = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, C = \begin{bmatrix} C_1 & C_2 \end{bmatrix}.$$

We define the two "subsystems" as

$$\Sigma_i = \left(\begin{array}{c|c} A_{ii} & B_i \\ \hline C_i & \end{array}\right), i = 1, 2,$$

which are referred to as *the reduced order systems obtained from* Σ *by balanced truncation*. The following theorem was proven in [50, 67, 10].

Theorem 2.6 *(e.g., [10], Theorem 7.9) The reduced systems* Σ_i *constructed from the reachable, observable, stable system* Σ *have the following properties:*

1. Σ_i *are balanced and have no pole in the open right complex half-plane.*
2. *If each diagonal entry of* Λ_1 *is different from all those of* Λ_2, Σ_i *are reachable, observable and stable.*
3. *Suppose that the Hankel singular values of* Σ *are* $\sigma_i, i = 1, \cdots, n$, *with multiplicities* $m_i, i = 1, \cdots, n$, *and* Λ_1 *contains the first* k *values with multiplicities. Then, the difference between* Σ *and the reduced order system* Σ_1 *is bounded from above by twice the sum of neglected Hankel singular values*

$$\|\Sigma - \Sigma_1\|_{\mathcal{H}_\infty} \leq 2(\sigma_{k+1} + \cdots + \sigma_n). \tag{2.23}$$

Note that the multiplicities of neglected Hankel singular values are not included in the upper bound (2.23).

One has a reason to be concerned about the error of the method. Indeed, the worst case is that $\sigma_i, i = k+1, \cdots, n$ are almost the same $\sigma_i, i = 1, \cdots, k$, and the true error may reach the right hand side of (2.23), which is very large, since in MOR the reduced order is much less than the original order. One can ensure a good approximation by balanced truncation method only if the Hankel singular values of the original system decay quickly. Fortunately, as mentioned in [11] and references therein, in most cases Hankel singular values decay very quickly and therefore the error caused by balanced truncation is small in such cases.

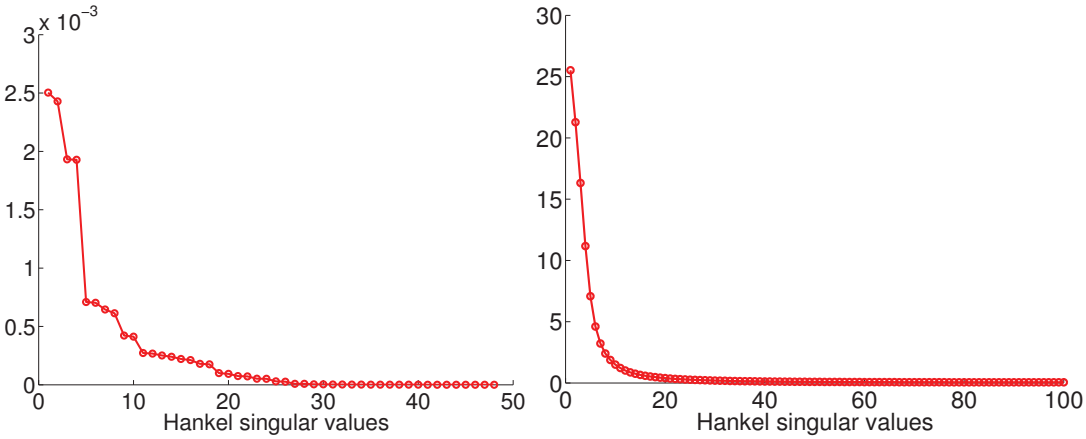

Figure 2.1: The decay of Hankel singular values of Building model (left) and Orr-Sommerfeld model (right) in [33]

Remark Computation of a balancing transformation requires $\mathcal{O}(N^3)$ floating operations and $\mathcal{O}(N^2)$ storage, which is unaffordable for very large N. That is why it is only applicable for moderate-sized systems. There have been some efforts to adapt the standard method for large systems. The main focus is on the solution to the Lyapunov equations [134, 18, 15]. However, these approaches theoretically loose the error bound.

In frequency domain, it is said that balanced truncation gives good approximations at high frequencies and bad ones at low frequencies, which in some cases are of interest. Some modifications of the standard method have been made in [112, 168, 158] to improve the behavior of reduced systems at low frequencies. The improvements were also made in order to preserve some special properties of the original systems during the model reduction by balanced truncation: preservation of contraction mapping was considered in [128], passivity in [135, 180, 142], positive realness in [170, 177, 143]. A survey on balanced truncation can be found in [73].

Balanced truncation is not only an approach for first order linear ordinary dynamical systems. In [169, 143], a similar technique which is based on balanced truncation and gramians of descriptor systems was proposed. Balanced truncation method for the second-order system was aimed at in [34, 141, 80]. Generalization

of reachability and observability gramians and balanced truncation for nonlinear systems were performed in [101, 38]. □

2.2.2 Proper Orthogonal Decomposition

Proper orthogonal decomposition (POD) is a model reduction method that constructs an optimal low-dimensional projection subspace based on given data. The idea of POD may appear under different names: *Karhunen-Loève Decomposition* or *Principal Component Analysis* (PCA) and in different fields other than MOR. It is said [22] that the idea of POD originated from some publications in the early of the 1940s [96, 114, 93]. It was first used as a model reduction tool in [116] for the investigation of inhomogeneous turbulence. Since then, in addition to being applied to the study of coherent structures and turbulence [85, 162, 21], POD was also exploited to solve numerous types of problems: data compression [9], image processing [144], fluid flows [148, 147], elliptic systems [92], control and inverse problems [98, 175]. The theoretical presentation in this part of this method is based on [86, 174].

We will start with discrete data. Let $X = [x_1, \cdots, x_n] \in \mathbb{R}^{N\times n}, n \leq N$ be of rank d. In practice, X is generated by experiments or simulations of a given system. One can think of each column of X as the state of the system that has been discretized into values at nodes taken at a time instant. They are so-called *snapshots*. It is always a desire to find a smaller group of vectors, preferably orthonormal, $\{\nu_i\}_{i=1}^k, k \leq d$, such that this group is the best representative of X. The task can be expressed as an optimization problem

$$\underset{\nu_i\in\mathbb{R}^N}{\operatorname{argmax}} \sum_{i=1}^{k}\sum_{j=1}^{n} |\langle x_j, \nu_i\rangle|^2 \text{ such that } \langle \nu_i, \nu_j\rangle = \delta_{ij}, 1 \leq i, j \leq k. \tag{2.24}$$

The SVD of matrices is an ideal tool to solve this problem. Let

$$X = U\Lambda V^T \tag{2.25}$$

be the SVD of X. That is, $U \in \mathbb{R}^{N\times N}$, $V \in \mathbb{R}^{n\times n}$ are orthogonal, $\Lambda \in \mathbb{R}^{N\times n}$ is a diagonal matrix whose diagonal entries are [2]

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_d > 0 = \cdots = 0.$$

By (2.25), the columns of U and V satisfy

$$\begin{aligned} Xv_i &= \sigma u_i, i = 1, \cdots, n, \\ X^T u_i &= \sigma v_i, i = 1, \cdots, n. \end{aligned} \tag{2.26}$$

It follows that the columns of U and that of V are eigenvectors of the symmetric positive semi-definite matrices XX^T and X^TX, respectively:

$$XX^T u_i = \sigma_i^2 u_i, i = 1, \cdots, N, \tag{2.27}$$

$$X^T X v_i = \sigma_i^2 v_i, i = 1, \cdots, n. \tag{2.28}$$

[2] The last $N - n$ columns of U can be chosen freely such that they, together with the first n columns, form an orthonormal basis.

Now, we turn back to the problem (2.24). For the case $k = 1$, let us define the associated Lagrange functional

$$\underset{(\nu,\lambda)\in\mathbb{R}^N\times\mathbb{R}}{\mathcal{L}(\nu,\lambda)} = \sum_{j=1}^{n} |\langle x_j, \nu\rangle|^2 + \lambda(1 - \|\nu\|).$$

The partial derivative of $\mathcal{L}(\nu,\lambda)$ with respect to ν is

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial u}(\nu,\lambda) &= \frac{\partial}{\partial u}(\nu^T X, X^T\nu + \lambda(1 - \nu^T\nu)) \\ &= 2XX^T\nu - 2\lambda\nu.\end{aligned}$$

The first order necessary optimal condition leads to

$$XX^T\nu = \lambda\nu.$$

Taking (2.27) into account, any column vector of U satisfies the necessary condition. It remains to find one amongst them, which solves (2.24), i.e., satisfies the sufficient condition. Suppose that $\tilde{\nu} \in \mathbb{R}^N$ is any vector of length one. Since U is an orthonormal basis of $\mathbb{R}^N$, $\tilde{\nu}$ can be represented as

$$\tilde{\nu} = UU^T\tilde{\nu}.$$

As a consequence,

$$\begin{aligned}\sum_{j=1}^{n} |\langle x_j, \tilde{\nu}\rangle|^2 &= \tilde{\nu}^T XX^T\tilde{\nu} \\ &= \tilde{\nu}^T UU^T XX^T UU^T\tilde{\nu} \\ &= \tilde{\nu}^T UU^T U\Sigma V^T V\Sigma^T U^T UU^T\tilde{\nu} \\ &= \tilde{\nu}^T U\Sigma\Sigma^T U^T\tilde{\nu}. \\ &\leq \sigma_1^2\tilde{\nu}^T UIU^T\tilde{\nu} \\ &= \sigma_1^2 \\ &= u_1^T XX^T u_1 = \sum_{j=1}^{n} |\langle x_j, u_1\rangle|^2.\end{aligned}$$

In the above argument, we have made use of the fact that $\Sigma\Sigma^T = \mathrm{diag}(\sigma_1^2, \sigma_2^2, \cdots, \sigma_d^2, 0, \cdots, 0) \in \mathbb{R}^{N\times N}$. This leads to the answer that the first column of U, u_1 is a solution to the problem (2.24) for the case $k = 1$ and the maximal value is σ_1^2.

With the same argument, one can show that the solution of the problem

$$\underset{\nu\in\mathbb{R}^N}{\mathrm{argmax}} \sum_{j=1}^{n} |\langle x_j, \nu\rangle|^2 \text{ such that } \|\nu\|^2 = 1 \text{ and } \langle\nu, \nu_1\rangle = 0$$

is u_2. This fact leads to the following statement.

Theorem 2.7 *(e.g., [174], Theorem 1.1) With the above notations, for any $k = 1, \cdots, d$, the solution to the problem (2.24) is the set of first k left singular values $\{u_i, i = 1, \cdots, k\}$ and the corresponding maximal value is $\sum_{i=1}^{k} \sigma_i^2$.*

By the result of this theorem, we define

Definition 2.13 *The first $k, k \leq d$, left eigenvectors $u_i, i = 1, \cdots, k$, are called the POD basis of rank k.*

For any set of orthonormal vectors $\{\nu_j, j = 1, \cdots, k\}$, we have,

$$\begin{aligned}\sum_{i=1}^{n} \|x_i - \sum_{j=1}^{k} \langle x_i, \nu_j \rangle \nu_j\|^2 &= \sum_{i=1}^{n} \langle x_i - \sum_{j=1}^{k} \langle x_i, \nu_j \rangle \nu_j, x_i - \sum_{j=1}^{k} \langle x_i, \nu_j \rangle \nu_j \rangle \\ &= \sum_{i=1}^{n} \langle x_i, x_i \rangle - \sum_{i=1}^{n} \sum_{j=1}^{k} |\langle x_i, \nu_j \rangle|^2.\end{aligned}$$

This suggests that the maximization problem (2.24) is equivalent to the following minimization problem

$$\underset{\nu_i \in \mathbb{R}^N}{\operatorname{argmin}} \sum_{i=1}^{n} \|x_i - \sum_{j=1}^{k} \langle x_i, \nu_j \rangle \nu_j\|^2 \text{ such that } \langle \nu_j, \nu_j \rangle = \delta_{ij}, 1 \leq i, j.$$

Moreover, denote by Υ a matrix consisting of orthonormal column vectors $\nu_j, j = 1, \cdots, k$. It follows that

$$\begin{aligned}\sum_{i=1}^{n} \langle x_i, x_i \rangle - \sum_{i=1}^{n} \sum_{j=1}^{k} |\langle x_i, \nu_j \rangle|^2 &= \operatorname{trace}(X^T X - X^T \Upsilon \Upsilon^T X) \\ &= \operatorname{trace}(X^T (I - \Upsilon \Upsilon^T) X) \\ &= \operatorname{trace}(X^T (I - \Upsilon \Upsilon^T)(I - \Upsilon \Upsilon^T) X) \\ &= \operatorname{trace}(((I - \Upsilon \Upsilon^T) X)^T (I - \Upsilon \Upsilon^T) X) \\ &= \|(I - \Upsilon \Upsilon^T) X\|_F^2 \\ &= \|X - \Upsilon \Upsilon^T X\|_F^2,\end{aligned}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. A consequence of Theorem 2.7 is

Corollary 2.8 *With the aforementioned notations, we have*

$$\|X - U(1:k)U(1:k)^T X\|_F^2 \leq \|X - \Upsilon \Upsilon^T X\|_F^2, \tag{2.29}$$

where $U(1:k)$ denotes the matrix formed by the first k columns of U.

In words, inequality (2.29) says that the subspace spanned by the POD basis minimizes the Frobenius norm of the difference between X and its projection on all subspaces of the same dimension.

Remark In the POD model reduction framework, the dimension of the state space N is usually much larger than the number of snapshots n. Hence, one would not compute u_i by solving the N-dimensional eigenvalue problem (2.27). Based on (2.26), one first solves the n-dimensioanl eigenvalues problem (2.28) and then computes u_i as

$$u_i = \frac{1}{\sigma_i} X v_i, i = 1, \cdots, k.$$

To answer the question how large the size of the POD basis should be to approximate the given data X well enough, there is so far no *a priori* criterion. One clue on which the decision can be based is the ratio

$$\frac{\sum_{i=1}^{k} \sigma_i}{\sum_{i=1}^{d} \sigma_i}.$$

One can consider to choose k such that this ratio is near 1.

The inner product used in the above presentation is the usual Euclidean one. In many cases where the system is governed by a partial differential equation, it is natural to use another inner product which is derived from the spatial discretization of the underlying equation rather than the original Euclidean product

$$\langle x, y \rangle_W = x^t W y,$$

where $W \in \mathbb{R}^{N \times N}$ is a positive definite matrix. More details are provided in [174].□

Now, we turn our attention to the case of continuous data. Instead of a matrix, we are given a trajectory $\{x(t), t \in [0, T]\} \subset \mathbb{R}^N$ and asked to find a set of k orthonormal vectors $\nu_i, i = 1, \cdots, k$ which approximate the trajectory as good a possible. In other words, solve the optimization problem

$$\underset{\nu_i \in \mathbb{R}^N}{\operatorname{argmin}} \int_0^T \left\| x(t) - \sum_{i=1}^{k} \langle x(t), \nu_i \rangle \nu_i \right\|^2 dt \text{ such that } \langle \nu_i, \nu_j \rangle = \delta_{ij}, 1 \leq i, j \leq k. \quad (2.30)$$

As in the discrete data case, this problem is equivalent to

$$\underset{\nu_i \in \mathbb{R}^N}{\operatorname{argmax}} \sum_{i=1}^{k} \int_0^T \left| \langle x(t), \nu_i \rangle \right|^2 dt \text{ such that } \langle \nu_i, \nu_j \rangle = \delta_{ij}, 1 \leq i, j \leq k.$$

In order to clarify the first necessary optimality condition, we define

$$\begin{array}{rcl} \mathcal{R} : \mathbb{R}^N & \longrightarrow & \mathbb{R}^N \\ \nu & \longmapsto & \mathcal{R}\nu = \int_0^T \langle x(t), \nu \rangle x(t) dt. \end{array}$$

It is shown in [174] that $\mathcal{R}$ is linear, bounded, non-negative, and symmetric. Thus $\mathcal{R}$ has a set of non-negative eigenvalues.

$$\mathcal{R} u_i = \lambda_i u_i, \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d > 0 = \cdots = 0, \quad (2.31)$$

where d is the rank of $\mathcal{R}$. One can observe that $\mathcal{R}$ plays the same role as UU^T in the discrete data case. And as in the previous case, the eigenvectors of $\mathcal{R}$ form the POD basis as stated in the following theorem, whose proof si given in [174].

Theorem 2.9 *([174], Theorem 1.12) Suppose that $x(t) \in \mathcal{C}([0,T],\mathbb{R}^N)$ is the unique solution of the state equation with a given initial condition. Then the solution to problem (2.30) is given by the first k eigenvectors of $\mathcal{R}, \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_k$.*

We show how to avoid solving the large eigenvalue problem (2.31) by the *method of snapshots* [161]. The matrix representing operator $\mathcal{R}$ in $\mathbb{R}^N$ is

$$R = \int_0^T x(t)x^T(t)dt. \tag{2.32}$$

Now, instead of continuous data $\{x(t), t \in [0,T]\} \subset \mathbb{R}^N$, we take some snapshots of that trajectory

$$x(t_j), 0 = t_0 < t_1 < t_2 < \cdots < t_n = T.$$

Matrix (2.32) can be approximated as

$$R = \sum_{j=1}^{n} x(t_j)x^T(t_j)\Delta_j,$$

where Δ_j is the step size $t_j - t_{j-1}$. If we write

$$X = \begin{bmatrix} x_1(t_1)\sqrt{\Delta_1} & \cdots & x_1(t_n)\sqrt{\Delta_n} \\ \cdots & \cdots & \cdots \\ x_N(t_1)\sqrt{\Delta_1} & \cdots & x_N(t_n)\sqrt{\Delta_n} \end{bmatrix} \in \mathbb{R}^{N\times n},$$

then matrix R can be written as $R = XX^T$. As in the discrete data case, we solve the n-dimensional eigenvalue problem

$$X^T X v_i = \lambda_i v_i$$

and compute the first n eigenvectors of $\mathcal{R}$

$$u_i = \frac{1}{\sqrt{\lambda_i}} X v_i, i = 1, \cdots, k.$$

This argument, on the one hand, shows that discrete data and continuous data cases are treated in a unifying manner, on the other hand, is a crucial point to formulate the so-called *balanced POD* [147], which will be presented later as a remark.

Now, given a POD basis $\{u_i, i = 1, \cdots, r\}$ constructed from data which are taken from a dynamical system

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t), \end{aligned} \tag{2.33}$$

where $A \in \mathbb{R}^{N\times N}, B \in \mathbb{R}^{N\times m}, C \in \mathbb{R}^{l\times N}$, we demonstrate how to use this basis to produce a reduced system. Since the given data usually contain the most typical states [98], and moreover the POD basis is their representative, the state vector $x(t)$ of the dimension N is approximated by $U\hat{x}(t), U = [u_1, \cdots, u_r]$, where the

new state vector $\hat{x}(t)$ is of the dimension $r \ll N$. That is, $\hat{x}(t)$ is the coordinate of the projection of a vector whose coordinate is $x(t)$ on the subspace spanned by $\{u_i, i = 1, \cdots, r\}$. System (2.33) becomes

$$\begin{aligned} U\dot{\hat{x}}(t) &= AU\hat{x}(t) + Bu(t), \\ y(t) &= C\hat{x}(t). \end{aligned} \tag{2.34}$$

To avoid the overdetermination of (2.34), one forces its residual to be orthogonal with an r-dimensional subspace of $\mathbb{R}^N$. The POD method chooses a Galerkin project framework, i.e., the chosen subspace is also the subspace spanned by $\{u_i, i = 1, \cdots, r\}$. The reduced system is therefore formulated as,

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

where $\hat{A} = U^T AU, \hat{B} = U^T B, \hat{C} = CU$.

Remark Note that the application of POD to MOR is not restricted to linear systems. In fact, it is a favorite reduction method for non-linear systems. For a general model of the form (2.1), the associated reduced order model is

$$\begin{aligned} \dot{\hat{x}}(t) &= U^T f(t, U\hat{x}(t), u(t)), t \in T, \\ \hat{y}(t) &= \eta(U\hat{x}(t), u(t)). \end{aligned}$$

Recall the definition (2.9) and (2.10) of reachability and observability gramians. If we denote the columns of the input matrix B as $b_1, \cdots, b_n$, then the impulse response $e^{At}B$ can be treated as the group of the response state vectors $x^i(t) = e^{At}b_i$ to the i-th unit impulse $\delta(t)e_i$, where e_i is the i-th unit vector of $\mathbb{R}^m$. Accordingly, the reachability gramian can be written as

$$\mathcal{R} = \int_0^\infty \sum_{i=1}^m x^i(t) x^{iT}(t) dt.$$

Likewise, the observability gramian is

$$\mathcal{Q} = \int_0^\infty \sum_{i=1}^m z^i(t) z^{iT}(t) dt,$$

where $z^i(t) = e^{A^T t} c_i^T, c_i$ is the i-th row of C. In balanced truncation one has to solve Lyapunov equations (2.11) and (2.12), which is expensive. In practice, impulse response state vectors $x^i(t), z^i(t)$ are given at time instants $t_1, \cdots, t_n$. The two gramians can be approximated by

$$\begin{aligned} \mathcal{R} &= \sum_{j=1}^n \sum_{i=1}^m x^i(t_j) x^{iT}(t_j) \Delta_j, \\ \mathcal{Q} &= \sum_{j=1}^n \sum_{i=1}^m z^i(t_j) z^{iT}(t_j) \Delta_j. \end{aligned}$$

Let us set

$$X = \begin{bmatrix} x_1^1(t_1)\sqrt{\Delta_1} & \cdots & x_1^1(t_n)\sqrt{\Delta_n} & \cdots & x_1^m(t_1)\sqrt{\Delta_1} & \cdots & x_1^m(t_n)\sqrt{\Delta_n} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ x_N^1(t_1)\sqrt{\Delta_1} & \cdots & x_N^1(t_n)\sqrt{\Delta_n} & \cdots & x_N^m(t_1)\sqrt{\Delta_1} & \cdots & x_N^m(t_n)\sqrt{\Delta_n} \end{bmatrix} \in \mathbb{R}^{N\times(mn)}.$$

Accordingly,

$$\mathcal{R} = XX^T.$$

Likewise,

$$\mathcal{Q} = YY^T.$$

Let

$$Y^TX = U\Sigma V^T = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix}$$

be the SVD of Y^TX and $\Sigma_1 \in \mathbb{R}^{r\times r}, r < \text{rank}(Y^TX)$, and

$$\Phi_1 = XV_1\Sigma^{-\frac{1}{2}},\ \Psi_1 = \Sigma^{-\frac{1}{2}}U_1^TY^T.$$

Then Φ_1 is composed of the first r columns of the approximate balancing transformation, Ψ_1 is the set of the first r rows of its inverse. That is, the new system

$$\begin{aligned} \hat{x}(t) &= \Psi_1A\Phi_1\hat{x} + \Psi_1Bu(t), \\ \hat{y}(t) &= C\Phi_1\hat{x}(t) \end{aligned}$$

will be the reduced system of system (2.33) produced by the approximate balanced truncation. A proof of this can be found in [147]. One can observe that the main advantage of balanced POD is that one needs not compute the two gramians. Instead, only two matrices X, Y, which can be determined from simulations or experiments, are needed. This actually shares the same idea with the original balanced truncation method proposed in [121]. Therefore, balanced POD method is an approximation of the balanced truncation method.

There have been various improvements of POD other than its primary version. The optimality of snapshot locations was addressed in [99], while quite many researches were trying to preserve some property of the original models. The stability was preserved during the POD reduction in [137], the Lagrangian structure in [100]. In [140], POD was applied to non-linear ODE initial value problems; the error and the effect of perturbation in data was analyzed. Some others focused on dealing with PDSs [92, 111, 7, 29, 6, 45]. □

2.2.3 Krylov Subspace Methods

As mentioned in Section 2.1, an LTI system $\Sigma = \left(\begin{array}{c|c} A & B \\ \hline C & \end{array}\right)$ can be given in the frequency domain by its transfer function $H(s) = C(sI - A)^{-1}B$. Therefore, one trend in MOR is to find a smaller-order system whose transfer function approximates the original one. This can be done through matching some first terms of the Laurent expansion of $H(s)$ at some point(s). Depending on the type of points to be matched, the problem is named differently.

- *Partial realization*: matching terms of the expansion around infinity.
- *Padé approximation*: matching terms of the expansion at zero.
- *Rational interpolation*: matching terms of the expansion at arbitrary point.

By making use of Neumann series, one can write

$$H(s) = \sum_{i=0}^{\infty} CA^i Bs^{-(i+1)}.$$

If A is non-singular,

$$H(s) = \sum_{i=0}^{\infty} -CA^{-(i+1)} Bs^i.$$

If $A - s_0 I$ is non-singular,

$$H(s) = \sum_{i=0}^{\infty} -C(A - s_0 I)^{-(i+1)} B(s - s_0)^i.$$

Note that these above equalities hold locally only.

Definition 2.14 *Matrices $CA^iB, CA^{-(i+1)}B$ and $C(A - s_0 I)^{-(i+1)}B$ are called the i-th moment of $H(s)$ about infinity, zero and s_0, respectively. The matrices $CA^iB, CA^{-(i+1)}B$ are also called high frequency moment and low frequency moment.*

Note that CA^iB is also called the Markov parameter. One can check that the moments about one point of the transfer function, up to a constant, are the value and consecutive derivatives of the transfer function at that point.

The problem of matching moments now becomes: seek an LTI system $\hat{\Sigma} = \left(\begin{array}{c|c} \hat{A} & \hat{B} \\ \hline \hat{C} & \end{array}\right)$ of order r whose transfer function $\hat{H}(s)$ shares the first $q(r)$ moments with $H(s)$. That can be equivalently written as

$$H(s) = \hat{H}(s) + \mathcal{O}((s - s_0)^{q(r)}).$$

For the case of SISO systems, one can easily verify that the transfer function of the LTI system $\hat{\Sigma}$ can be represented as a rational function of frequency

$$\hat{H}(s) = \frac{P_{N-1}(s)}{Q_N(s)},$$

where $P_{N-1}(s)$ is the polynomial of degree at most $N - 1$ and $Q_N(s)$ of degree N. To find a Padé approximation, one can compute coefficients of $P_{N-1}(s), Q_N(s)$ through solving linear equations whose coefficients and right hand side are moments of $H(s)$ (see, e.g., [70]). This approach is called *explicit moment matching*. It was once widely used in the Asymptotic Waveform Evaluation (AWE) technique

[136, 35, 36]. However, it turned out that the explicit moment matching method suffers from instability and is therefore inaccurate, especially when the order of the original system becomes larger. An example in [53] shows that this method deteriorates even when the order merely exceeds 10. One reason is that the method requires to explicitly compute moments of the original system.

To avoid this shortcoming, the reduced system needs to match moments without explicitly computing them. This can be achieved with Krylov subspace methods.

Definition 2.15 *Given a square matrix $A \in \mathbb{R}^{N\times N}$ and a vector $b \in \mathbb{R}^N$, the n-th Krylov subspace $\mathcal{K}_n(A,b)$ is the subspace spanned by vectors $\{b, Ab, A^2b, \cdots, A^{n-1}b\}$, i.e.,*

$$\mathcal{K}_n(A,b) := \mathrm{colsp}([b\ Ab\ A^2b\ \cdots\ A^{n-1}b]),$$

where colsp of a matrix is the subspace spanned by the columns of this matrix. The state $x(t)$ of a SISO system with homogeneous initial condition has the form $x(t) = \int_0^t e^{A(t-\tau)}bu(\tau)d\tau$. Hence constructing a reduced system aims to approximate the term $e^{At}b$. Based on the observation that the matrix exponential operator can be approximated through projection on Krylov subspaces [62, 155], the connection between Krylov subspace and Padé approximation was exploited in [59]. Nevertheless, the original idea is most likely due to [69, 173] during solving the partial realization problem. We restate the theorem here.

Theorem 2.10 *([59], Theorem 1). Given an LTI SISO system $\Sigma = \left(\begin{array}{c|c} A & b \\ \hline c & \end{array}\right)$. Assume that there exist full-rank matrices $V, W \in \mathbb{R}^{N\times r}$ such that*

$$\begin{aligned} &\mathrm{colsp}(V) = \mathcal{K}_r(A,b), \\ &\mathrm{colsp}(W) = \mathcal{K}_r(A^T,c^T), \\ &W^TV = I. \end{aligned} \tag{2.35}$$

Let the reduced system $\hat{\Sigma} = \left(\begin{array}{c|c} \hat{A} & \hat{b} \\ \hline \hat{c} & \end{array}\right)$ be constructed by the corresponding oblique projections on Krylov subspaces $\mathcal{K}_r(A,b)$ and $\mathcal{K}_r(A^T,c^T)$, i.e., $\hat{A} = W^TAV, \hat{b} = W^Tb, \hat{c} = cV$. Then the first $2r$ Markov parameters of the reduced system and original system are identical.

For Padé approximation and rational interpolation, we have the corresponding statements. All one has to do is to replace V, W in (2.35) by $\mathrm{colsp}(V) = \mathcal{K}_r(A^{-1},b), \mathrm{colsp}(W) = \mathcal{K}_r(A^{-T},c^T)$ for Padé approximation and $\mathrm{colsp}(V) = \mathcal{K}_r((A-s_0I)^{-1},b), \mathrm{colsp}(W) = \mathcal{K}_r((A-s_0I)^{-T},c^T)$ for rational interpolation.

In practice, very often s is chosen to be $i\omega$ where $\omega \in \mathbb{R}_+$ is some frequency. Accordingly, $H(i\omega)$ becomes the *amplification factor* which amplifies the input to yield the output at that frequency. This theorem gives an efficient way to find an approximation of the transfer function of a large order system in a frequency range of interest. Quite often, one is interested in approximation at a wide range of

frequencies. Constructing a reduced system as in Theorem 2.10 cannot cover this task, since $\hat{\Sigma}$ approximates Σ only around s_0. A natural question is why one cannot include all Krylov subspaces associated with frequencies of interest in to the projection subspace spanned by V and W. This idea was most probably first explored in [173] while the authors tried to match both high and low frequency moments, i.e., partial realization and Padé approximation are both dealt with simultaneously. For rational interpolation at several points, the most general solution is due to [60, 70], which is stated for descriptor systems of the form $\Sigma = \left(\begin{array}{c|c|c} E & A & b \\ \hline & c & \end{array}\right)$.

Theorem 2.11 *([70], Theorem 3.1). Assume that*

$$\operatorname{colsp}(V) \supset \bigcup_{i=1}^{k} \mathcal{K}_{J_{b_i}}\Big((A - s_i E)^{-1}E, (A - s_i E)^{-1}b\Big), \tag{2.36}$$

$$\operatorname{colsp}(W) \supset \bigcup_{i=1}^{k} \mathcal{K}_{J_{c_i}}\Big((A - s_i E)^{-T}E^T, (A - s_i E)^{-T}c^T\Big), \tag{2.37}$$

and $W^T(A - s_i E)V$ is non-singular for all $i = 1, \cdots, k$ then

$$-c^T\Big((A - s_i E)^{-1}E\Big)^{j_i - 1}(A - s_i E)^{-1}b = -\hat{c}^T\Big((\hat{A} - s_i \hat{E})^{-1}\hat{E}\Big)^{j_i - 1}(\hat{A} - s_i \hat{E})^{-1}\hat{b} \tag{2.38}$$

for $J_i = 1, 2, \cdots, J_{b_i} + J_{c_i}$ and $i = 1, \cdots, k$ where $\hat{E} = W^T E V, \hat{A} = W^T A V, \hat{B} = W^T B, \hat{C} = CV$. In words, the first $J_{b_i} + J_{c_i}$ moments of the transfer functions of Σ and $\hat{\Sigma}$ about s_i are identical.

In addition to the capacity for interpolating at multiple points, Theorem 2.11 improves the previous results in two aspects. The first one is the flexibility in choosing projection matrices. It requires only an inclusion, instead of an equality as in Theorem 2.10. Due to the so-called *deflation* or *rank-deficiency* of Krylov matrices, the dimensions of $\mathcal{K}_r((A-s_iE)^{-1}E, (A-s_iE)^{-1}b)$ and $\mathcal{K}_r((A-s_iE)^{-T}E^T, (A-s_iE)^{-T}c^T)$ are not necessarily equal. The relaxation allows the adjustment of the number of columns of V and W to ensure that they are the same. The second advantage is that no bi-orthogonality is required. This results in the *two-sided Arnoldi* [41, 157] and the *dual rational Arnoldi* methods [127] in MOR[3].

Now, we turn our attention to computational aspects. It should be noted that, to match moments, (2.36) and (2.37) do not have to hold simultaneously. In fact, a lemma in [70] showed that if either of (2.36) and (2.37), say (2.36), holds and W is chosen such that it is full-rank, has the same dimension as V and $\det(W^T A V) \neq 0$, J_{b_i} moments are matched. Principally, there is no more constraint imposed on V. However, in order to avoid ill-conditioning of V, it is preferred to have orthonormal column vectors, i.e., $V^T V = I$. From now on, we will refer to matrices that have this property as *columnwise orthogonal matrices*. The task of constructing V can then be stated as: Compute an orthonormal basis of the Krylov subspace $\mathcal{K}_n(A, b)$. It is completed by an Arnoldi process [12].

[3] Two-sided Arnoldi was also used for eigenvalue problem, see, e.g., [154].

Algorithm 1 Arnoldi algorithm

Input: A, b

Output: V columnwise orthogonal such that $\text{colsp}\{V\} = \mathcal{K}_n(A, b)$

1: $V(:,1) := \dfrac{b}{\|b\|}$

2: **For** $i = 1$ **to** n, **do**:

3: Compute $h_{ji} = V(:,i)^T AV(:,j), j = 1, \cdots, i.$

4: Compute $v = AV(:,i) - \sum_{j=1}^{i} h_{ji} V(:,j)$.

5: $h_{i+1\ j} := \|v\|$

6: **If** $h_{i+1\ j} = 0$ **then stop**

7: $V(:,i+1); = v/h_{i+1\ j}$

8: **End do**

Note that the standard Arnoldi process uses the classical Gram-Schmidt orthogonalization. It was shown in [23] that the orthogonality can be lost in the computational process, while it is essential to the accuracy of the Arnoldi algorithm [68]. For ill-conditioned matrices, the loss of orthogonality can be severe [24]. Therefore, re-orthogonalization in Gram-Schmidt steps in the Arnoldi algorithm should be used. For more details, see [66].

Suppose that V, satisfying (2.36), has been computed using the Arnoldi algorithm. The *one-sided Arnoldi method* chooses $W = V$. The associated reduced system is then

$$\hat{\Sigma} = \left(\begin{array}{c|c|c} V^T EV & V^T AV & V^T b \\ \hline & cV & \end{array}\right).$$

Likewise, if in addition, W is constructed using the Arnoldi algorithm, the reduced system by the two-sided Arnoldi method is

$$\hat{\Sigma} = \left(\begin{array}{c|c|c} W^T EV & W^T AV & W^T b \\ \hline & cV & \end{array}\right).$$

Another way to construct V and W without having to run the Arnoldi process twice is the Lanczos algorithm [102] for unsymmetric matrices. Instead of two columnwise orthogonal matrices V and W, the Lanczos algorithm computes two bi-orthogonal matrices, i.e., $W^T V = I$, satisfying (2.36) and (2.37). Unlike the Arnoldi algorithm, whose k-th step involves k vectors, each step of the Lanczos algorithm requires only two three-term recurrences. This is clearly an advantage of storage during operating large systems. However, the computation using the Lanczos process may be unstable, see, e.g., [71].

We now turn to the case of MIMO systems. Based on results derived for the SISO case, there are two ways to establish the corresponding results for the MIMO case. The first one, due to [70], is to directly apply Theorem 2.11 for the j-th column of the input matrix B and the i-th row of the output matrix C. This approach considers each ij-th entry of the transfer function $H(s) \in \mathbb{R}^{l\times m}$ as a transfer function that connects the j-th input to the i-th output. The second approach is based on the so-called *block Krylov subspace*, which is defined in the following.

Definition 2.16 *Given $A \in \mathbb{R}^{N\times N}, B \in \mathbb{R}^{N\times m}$, where B is full-rank, the n-th Krylov subspace $\mathcal{K}_n(A, B)$ is*

$$\mathcal{K}_n(A, B) := \text{colsp}(B \; AB \; \cdots \; A^{n-1}B).$$

A similar theorem for MIMO systems like Theorem 2.11 can be proven. Since in Chapter 3 we will present an extension for parameter-dependent transfer functions, detailed discussion on this topic is skipped here. Theoretically, the two approaches lead to the same result. However, from our point of view, the second approach has a better connection with the definition of moments of MIMO transfer functions. It

Algorithm 2 Lanczos algorithm for unsymmetric matrices

Input: A, b, c

Output: V, W such that $\text{colsp}\{V\} = \mathcal{K}_n(A, b)$, $\text{colsp}\{W\} = \mathcal{K}_n(A^T, c)$ and $W^TV = I$

1: Choose $V(:, 1)$ and $W(:, 1)$ such that $W(:, 1)^TV(:, 1) = 1$.

2: Set $\beta_1 = \delta_1 = 0$, $V(:, 0) = W(:, 0) = 0$

3: **For** $i = 1$ **to** n, **do**:

4: $\quad \alpha_i := W(:, i)^TAV(:, i)$

5: $\quad v := AV(:, i) - \alpha_iV(:, i) - \beta_iV(:, i - 1)$

6: $\quad w := A^TW(:, i) - \alpha_iW(:, i) - \delta_iW(:, i - 1)$

7: $\quad \delta_{i+1} := \sqrt{w^Tv}$. **If** $\delta_{i+1} = 0$ **then stop**

8: $\quad \beta_{i+1} := \dfrac{w^Tv}{\delta_{i+1}}$

9: $\quad V(:, i + 1) := \dfrac{v}{\delta_{j+1}}$

10: $\quad W(:, i + 1) := \dfrac{w}{\beta_{j+1}}$

11: **End do**

also motivates the development of the *block Arnoldi process* [156, 90] and the *block Lanczos algorithm* [40, 153, 94, 54].

In practice, implementation of Arnoldi and Lanczos procedures may face some difficulties. The most frequently-happened situation which stops the iteration is the *breakdown*. This occurs in the Arnoldi process when $v = 0$. In the Lanczos algorithm, this happens when $\delta_{i+1} = 0$, for the reason is either v or w or both are equal to 0 (referred to as *curable breakdown*), or $v \neq 0, w \neq 0$ but v is orthogonal to w (referred to as *incurable breakdown or serious breakdown*). Another problem is the *deflation* in the block version of both Arnoldi and Lanczos processes. The so-called *inexact deflation*, which is caused by finite-precise arithmetic, has to be taken into account. In addition, a long Arnoldi iteration causes the problem of memory storage. For solutions to these problems, the reader is referred to [131, 164, 58, 165, 57].

Remark An apparent advantage of Krylov subspace methods in comparison with balanced truncation and POD is the lower computational cost. Computing an r-order reduced system of an N-order system needs $\mathcal{O}(N^2r)$ floating operations if the system is dense and $\mathcal{O}(Nr^2)$ if it is sparse [10]. However, the stability of the original system is, in general, not preserved and no *a priori* error bound is derived. There have been some improvements of Krylov subspace methods in order to preserve passivity during the reduction, see e.g., [48, 52, 181]. □

2.2.4 Final Remarks

The MOR methods presented so far in this section cannot, of course, cover all contributions to such an active field of research. Optimization based MOR has not been mentioned. These approaches try to minimize the difference between the reduced order transfer function and the original one. The difference of these approaches lies in the different norms used. The authors of [67, 88, 91, 18] chose the Hankel norm, while the authors of [72, 97] used $\mathcal{H}_2$ and $\mathcal{H}_{2,\alpha}$ norm. For MIMO systems, the approaches in [61, 179] considered a relaxation of matching moments, namely *tangential interpolation* conditions. A surprising fact is that the search for the first-order necessary condition for the optimal $\mathcal{H}_2$ model reduction results in tangential interpolation constraints [72, 97].

2.3 Some Manifolds in Linear Algebra

This section introduces some manifolds in linear algebra. We focus on Grassmann manifolds, which will be used later for the interpolation of projection subspaces. The presentations of others, $\mathcal{SPD}(n)$ - the manifold of symmetric positive definite $n \times n$ matrices and $\mathcal{GL}(n)$ - the manifold of invertible $n \times n$ matrices (also known as *general linear group*), are only for a broader view and the material for our discussions about methods proposed by others. We do not intend to recall the theory of differential geometry and abstract Riemann manifolds. The interested reader is referred to [27, 82, 167, 103] for general theory. The summary in this section is based on [56, 2, 139]. One can also consult [178, 47, 119].

2.3.1 Topological Structure of Grassmann Manifolds

Let $k \leq n$ be two positive integers and $\mathcal{ST}(k,n)$ denote the set of full-rank $n \times k$ real matrices. This set is known as the *Stiefel manifold* and has been investigated, e.g., in [89, 47, 1]. The subset $\mathcal{O}(k,n)$, composed of all columnwise orthogonal $n \times k$ real matrices, is sometimes called the *compact Stiefel manifold*. In this section, for any $n \times k$ real matrix A, we will use the Frobenius norm

$$\|A\|_F := \Big(\sum a_{ij}^2\Big)^{\frac{1}{2}}.$$

Next, we denote by $\mathcal{G}(k,n)$ the set of k-dimensional subspaces of $\mathbb{R}^n$. Each element of $\mathcal{G}(k,n)$ is represented (not uniquely) by an element of $\mathcal{ST}(k,n)$. Hence,

$$\begin{array}{rccc} \pi: & \mathcal{ST}(k,n) & \longrightarrow & \mathcal{G}(k,n) \\ & Y & \longmapsto & \mathcal{Y} := \mathrm{colsp}(Y). \end{array}$$

is a surjective mapping. For any two matrices $Y_1, Y_2 \in \mathcal{ST}(k,n), \pi(Y_1) = \pi(Y_2)$ if and only if there is $M \in \mathcal{GL}(k)$ such that $Y_1 = Y_2 M$. It is easy to verify that this is an equivalence relation. Accordingly, for $\mathcal{Y} \in \mathcal{G}(k,n), \pi^{-1}(\mathcal{Y})$ is the equivalence class $Y\mathcal{GL}(k)$, where Y spans $\mathcal{Y}$. The topology on $\mathcal{G}(k,n)$ is defined as the final topology with respect to π, i.e., the strongest topology that makes π continuous (see, e.g., [49].) We directly deduce from the definition that π is open.

2.3.2 Differential Structure of Grassmann Manifolds

For each point $\mathcal{Y}$ in $\mathcal{G}(k,n)$, one can single out one element of $\pi^{-1}(\mathcal{Y})$ by means of cross sections [2]. Let $U \in \mathcal{ST}(k,n)$, the set

$$\mathcal{S}_U := \{V \in \mathcal{ST}(k,n) : U^T(V - U) = 0\}$$

is called the *affine cross section*. Define by

$$IN_U := \{V \in \mathcal{ST}(k,n) : U^T V \text{ is invertible}\} \subset \mathcal{ST}(k,n)$$

and $\mathcal{IN}_U := \pi(IN_U) \subset \mathcal{G}(k,n)$. If $V \in IN_U$, the equivalence class $V\mathcal{GL}(k)$ intersects with $\mathcal{S}_U$ at the unique point $V(U^TV)^{-1}U^TU$. This allows us to define the mapping

$$\begin{array}{rccc} \sigma_U: & \mathcal{IN}_U & \longrightarrow & \mathcal{S}_U \\ & \pi(V) & \longmapsto & V(U^TV)^{-1}U^TU, \end{array}$$

which is called the *cross section mapping*. It is a bijection between $\mathcal{IN}_U$ and $\mathcal{S}_U$.

Let $J = (j_1, \cdots, j_k) \in \mathbb{N}^k$, where $1 \leq j_1 < \cdots < j_k \leq n$. Denote by I_n^k the set of all indeces J satifying the mentioned conditions. For any $n \times k$ matrix A, we denote by A_J the $k \times k$ submatrix consisting of the j_1-th, $\cdots$, j_k-th row of A and its complement A_J^C, the $(n-k) \times k$ submatrix that remains after removing A_J from A. Denote by $E^J := [e_{j_1} \cdots e_{j_k}], J \in I_n^k$, the matrix whose comlumns are the unit vector in $\mathbb{R}^n$. It follows directly that

$$\mathcal{S}_{E^J} = \{V \in \mathcal{ST}(k,n) : V_J = I\}$$

and

$$IN_{E^J} = \{V \in \mathcal{ST}(k,n) : \det(V_J) \neq 0\}.$$

Obviously, $\{IN_{E_J}, J \in I_n^k\}$ is an open covering of $\mathcal{ST}(k,n)$. Since π is continuous, open and surjective, $\{\mathcal{IN}_{E^J}, J \in I_n^k\}$ is an open covering of $\mathcal{G}(k,n)$. Note that, by the "freely-chosen" part V_J^C, $\mathcal{S}_{E^J}$ can be identified with $\mathbb{R}^{(n-k)\times k}$.

Theorem 2.12 *([56]) The family $\{(\mathcal{IN}_{E^J}, \sigma_{E^J}), J \in I_n^k\}$ defines a differentiable structure of dimension $(n-k)k$ on $\mathcal{G}(k,n)$.*

Definition 2.17 *The set $\mathcal{G}(k,n)$ together with the differentiable structure defined above is called the (real) Grassmann manifold of k-dimensional subspaces of the linear space $\mathbb{R}^n$.*

The proof of Theorem 2.12 can be found, e.g., in [27, 56]. A more general parameterization of elements of $\mathcal{G}(k,n)$ was given in [2]:

$$\mathbb{R}^{(n-k)\times k} \ni M \mapsto \pi(U + U_\perp M) \in \mathcal{IN}_U$$

for any $U \in \mathcal{ST}(k,n)$, where $U_\perp \in \mathcal{ST}(n-k,n)$ such that $U^T U_\perp = 0$.

2.3.3 Riemann Structure on Grassmann Manifolds

Given a point $\mathcal{W}$ of $\mathcal{G}(k,n)$, it is necessary to characterize the tangent space $T_{\mathcal{W}}\mathcal{G}(k,n)$ at $\mathcal{W}$. Let $W \in \mathcal{ST}(k,n)$ span $\mathcal{W}$. The *vertical space* V_W and the *horizontal space* H_W of W are defined as the sets of the matrices:

$$\begin{aligned} V_W &:= W\mathbb{R}^{k\times k}, \quad \text{and} \\ H_W &:= W_\perp\, \mathbb{R}^{(n-k)\times k} \subset \mathbb{R}^{n\times k}, \end{aligned} \tag{2.39}$$

respectively. While the elements of V_W do not modify the range of W, the elements of H_W do. They represent the vectors of the tangent space to $\mathcal{G}(k,n)$ at $\mathcal{W}$. Conversely, it was confirmed in [2] that for any tangent vector ξ to $\mathcal{G}(k,n)$ at $\mathcal{W}$, there exists one and only one horizontal vector $\xi_{\diamond W} \in H_W$ representing ξ. The vector $\xi_{\diamond W}$ is called the *horizontal lift* of $\xi \in T_{\mathcal{W}}\mathcal{G}(k,n)$. If we change the representation of $\mathcal{W}$ by WM instead of W, where $M \in \mathcal{GL}(k)$, the horizontal lift will change as in the following formula:

$$\xi_{\diamond WM} = \xi_{\diamond W} M, M \in \mathcal{GL}(k). \tag{2.40}$$

We are now ready to define and formulate the Riemann metric on tangent spaces. For $\xi, \zeta \in T_{\mathcal{W}}\mathcal{G}(k,n)$, the inner product of these two vectors is defined as

$$< \xi, \zeta >_{\mathcal{W}} := \text{ trace}\Big((W^T W)^{-1} \xi_{\diamond W}^T \zeta_{\diamond W}\Big).$$

By (2.40), it follows that,

$$\begin{aligned} \text{trace}\Big(\big((WM)^T WM\big)^{-1} \xi_{\diamond MW}^T \zeta_{\diamond MW}\Big) &= \text{trace}\Big(M^{-1}(W^T W)^{-1} M^{-T} M^T \xi_{\diamond W}^T \zeta_{\diamond W} M\Big) \\ &= \text{trace}\Big(M^{-1}(W^T W)^{-1} \xi_{\diamond W}^T \zeta_{\diamond W} M\Big) \\ &= \text{trace}\Big((W^T W)^{-1} \xi_{\diamond W}^T \zeta_{\diamond W}\Big) \end{aligned}$$

for any $M \in \mathcal{GL}(k)$. This means that the definition of the metric is independent of the basis of $\mathcal{W}$.

2.3.4 Geodesic Paths, the Exponential Mapping and the Logarithmic Mapping

A geodesic path connecting two points of a manifold is roughly the locally shortest curve that combines these two points. In $\mathbb{R}^n$, for example, geodesic paths are straight lines. In abstract Riemann manifolds, characterization of geodesic paths is however more complicated. It is based on the Riemann connection $\nabla_\eta \xi$, the derivative of a tangent vector field $\xi(t)$ in the direction of a tangent vector η. A smooth curve $\mathcal{Y}(t)$ on $\mathcal{G}(k,n)$ is called a *geodesic path* (see, e.g., [27]) if

$$\nabla_{\dot{\mathcal{Y}}} \dot{\mathcal{Y}} = 0. \tag{2.41}$$

In words, the tangent vector $\dot{\mathcal{Y}}$ is parallel transported along $\mathcal{Y}$. Based on (2.41), an explicit expression for geodesic paths on Grassmann manifolds is formulated.

Theorem 2.13 *([2], Theorem 3.6). Let $\mathcal{Y}(t)$ be a geodesic path on $\mathcal{G}(k,n)$ from $\mathcal{Y}_0$ with initial velocity $\dot{\mathcal{Y}}_0 \in T_{\mathcal{Y}_0}\mathcal{G}(k,n)$. Suppose that $Y_0 \in \mathcal{ST}(k,n)$ spans $\mathcal{Y}_0$, $\dot{\mathcal{Y}}_{0\diamond Y_0}$ is the horizontal lift of $\dot{\mathcal{Y}}_0$ and $\dot{\mathcal{Y}}_{0\diamond Y_0}(Y_0^T Y_0)^{-\frac{1}{2}} = U\Sigma V^T$ is the thin SVD. Then*

$$\mathcal{Y}(t) = \pi\Big(Y_0(Y_0^T Y_0)^{-\frac{1}{2}} V \cos(\Sigma t) + U \sin(\Sigma t)\Big). \tag{2.42}$$

We follow [27] in defining the exponential mapping

$$\begin{array}{rccc} Exp_{\mathcal{W}}: & T_{\mathcal{W}}\mathcal{G}(k,n) & \longrightarrow & \mathcal{G}(k,n) \\ & \xi & \longmapsto & \mathcal{Y}(1), \end{array}$$

where $\mathcal{Y}(t)$ is the only geodesic path determined by the initial condition $\mathcal{Y}(0) = \mathcal{W}$ and $\dot{\mathcal{Y}}(0) = \xi$. By Theorem 2.13, the formulation of exponential mapping is as follows: Let $\mathcal{W} \in \mathcal{G}(k,n)$ be spanned by $W \in \mathcal{ST}(k,n)$, $\xi \in T_{\mathcal{W}}\mathcal{G}(k,n)$. Assume that $\xi_{\diamond W}(W^T W)^{\frac{1}{2}} = U\Sigma V^T$. Then

$$Exp_{\mathcal{W}}(\xi) = \pi\Big(W(W^T W)^{-\frac{1}{2}} V \cos(\Sigma) + U \sin(\Sigma)\Big). \tag{2.43}$$

The logarithmic mapping $Log_{\mathcal{W}}$ maps each point in a neighborhood of $\mathcal{W} \in \mathcal{G}(k,n)$ to a vector $\xi \in T_{\mathcal{W}}\mathcal{G}(k,n)$. In this neighborhood, it is the inverse of $Exp_{\mathcal{W}}$. The formulation of logarithmic mapping is given in [2]. Assume that $\mathcal{W}, \mathcal{Z} \in \mathcal{G}(k,n)$ are spanned by two columnwise orthogonal matrices $W, Z \in \mathcal{O}(k,n)$, respectively, such that $\det(W^T Z) \neq 0$. Let

$$(I - WW^T)Z(W^T Z)^{-1} = U\Sigma V^T$$

be the thin SVD. Then the horizontal lift of $\xi = Log_{\mathcal{W}}(\mathcal{Z})$ is

$$\xi_{\diamond W} = U \arctan(\Sigma) V^T. \tag{2.44}$$

2.3.5 Examples

Consider $\mathcal{G}(1,2)$. We will use $\mathcal{O}(1,2)$, which can be visualized by the unit circle $S^1 \subset \mathbb{R}^2 \equiv \mathbb{C}$, to represent elements of $\mathcal{G}(1,2)$. Now let the point $\mathcal{Z}$ be represented by $Z = [1 \quad 0]^T$. By (2.39), the horizontal space, whose elements represent the tangent space to $\mathcal{G}(1,2)$ at $\mathcal{Z}$, is $H_Z = \{[0 \quad a]^T, a \in \mathbb{R}\}$. So this tangent space is the line tangent to S^1 at point $(1,0)$. Now take any vector $\theta \in T_{\mathcal{Z}}\mathcal{G}(1,2)$ whose horizontal lift is $[0 \quad \theta]^T$ such that $-\frac{\pi}{2} < \theta < \frac{\pi}{2}$. We have the thin SVD

$$\begin{aligned}\theta \diamondsuit_Z (Z^T Z)^{-\frac{1}{2}} &= \begin{bmatrix}0\\ \theta\end{bmatrix} \left(\begin{bmatrix}1 & 0\end{bmatrix} \begin{bmatrix}1\\ 0\end{bmatrix} \right)^{-\frac{1}{2}} \\ &= \begin{bmatrix}0\\ 1\end{bmatrix} [\theta] \, [1] \, .\end{aligned}$$

Applying formula (2.43) yields

$$\begin{aligned}Exp_{\mathcal{Z}}(\theta) &= \pi\Big(\begin{bmatrix}1\\ 0\end{bmatrix} \left(\begin{bmatrix}1 & 0\end{bmatrix} \begin{bmatrix}1\\ 0\end{bmatrix} \right)^{-\frac{1}{2}} [1] \cos(\theta) + \begin{bmatrix}0\\ 1\end{bmatrix} \sin(\theta) \Big) \\ &= \pi\Big(\begin{bmatrix}\cos(\theta)\\ \sin(\theta)\end{bmatrix} \Big) = \mathcal{W}.\end{aligned}$$

One can identify the vector θ with the complex number $i\theta$, the point $\mathcal{W}$ with the complex number $\cos(\theta) + i\sin(\theta)$. The above mapping $\mathbb{C} \ni i\theta \mapsto \cos(\theta) + i\sin(\theta) = e^{i\theta}$ is the standard exponential of a complex number.

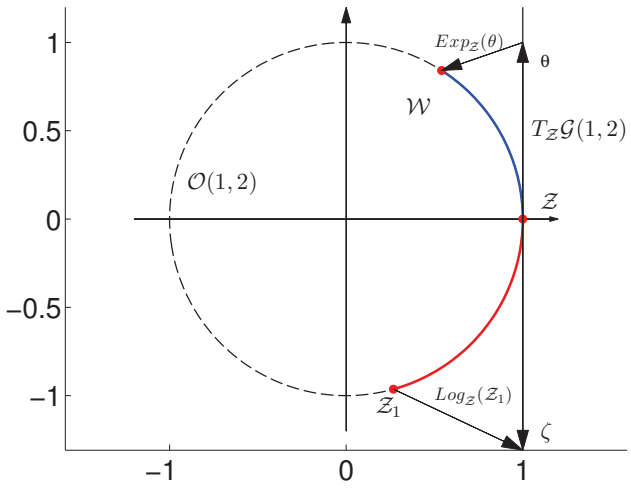

Figure 2.2: Exponential and logarithmic mappings on $\mathcal{G}(1,2)$

Now, pick another point $\mathcal{Z}_1$ spanned by $Z_1 = \begin{bmatrix}\cos(\phi)\\ \sin(\phi)\end{bmatrix}$, where $-\frac{\pi}{2} < \phi < \frac{\pi}{2}$. We

will compute $\zeta = Log_{\mathcal{Z}}(\mathcal{Z}_1)$.

$$\begin{aligned}(I - ZZ^T)Z_1(Z^TZ_1)^{-1} &= \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix} \right) \begin{bmatrix} \cos(\phi) \\ \sin(\phi) \end{bmatrix} \left(\begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \cos(\phi) \\ \sin(\phi) \end{bmatrix} \right)^{-1} \\ &= \begin{bmatrix} 0 \\ \tan(\phi) \end{bmatrix} \\ &= \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} \tan(\phi) \end{bmatrix} \begin{bmatrix} 1 \end{bmatrix}.\end{aligned}$$

Applying (2.44), the horizontal lift of ζ is

$$\begin{aligned}\zeta_{\Diamond Z} &= \begin{bmatrix} 0 \\ 1 \end{bmatrix} \arctan\left(\begin{bmatrix} \tan(\phi) \end{bmatrix} \right) \begin{bmatrix} 1 \end{bmatrix} \\ &= \begin{bmatrix} 0 \\ \phi \end{bmatrix}.\end{aligned}$$

Again, identifying $\mathcal{Z}_1$ with $\cos(\phi) + i\sin(\phi)$, vector ζ with $i\phi$, the mapping $\mathbb{C} \ni \cos(\phi) + i\sin(\phi) \mapsto i\phi = ln(\cos(\phi) + i\sin(\phi))$ is the natural logarithm.

We consider another example on $\mathcal{G}(2,3)$. Let $\mathcal{Z}$ and $\mathcal{Z}_1$ be two distinct points on $\mathcal{G}(2,3)$ represented by

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad Z_1 = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} \\ 0 & \frac{2}{\sqrt{6}} \end{bmatrix}, \text{ respectively.}$$

By (2.39), the horizontal space of Z has the form

$$H_Z = \left\{ W = \begin{bmatrix} 0 & 0 \\ w_1 & w_2 \\ 0 & 0 \end{bmatrix}, w_1, w_2 \in \mathbb{R} \right\}. \tag{2.45}$$

We will compute $\mathcal{Z}_2 = Exp_{\mathcal{Z}}(Log_{\mathcal{Z}}(\mathcal{Z}_1))$. In the following computations, as standard short form in Matlab, we display only four decimal digits. First, we have the thin SVD

$$(I - ZZ^T)Z_1(Z^TZ_1)^{-1} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 2.2361 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} -0.4472 & 0.8944 \\ 0.8944 & 0.4472 \end{bmatrix}.$$

Applying (2.44), the horizontal lift of $Log_{\mathcal{Z}}(\mathcal{Z}_1)$ is

$$H = \begin{bmatrix} 0 & 0 \\ -0.5144 & 1.0288 \\ 0 & 0 \end{bmatrix}.$$

Then, by (2.43), the matrix Z_2 spanning $\mathcal{Z}_2$ is

$$Z_2 = \begin{bmatrix} -0.1826 & 0.8944 \\ 0.9129 & 0 \\ 0.3651 & 0.4472 \end{bmatrix}.$$

With Matlab, one can easily check that Z_1 and Z_2 span the same space, i.e., $\mathcal{Z}_1 \equiv \mathcal{Z}_2$ on $\mathcal{G}(2,3)$.

2.3.6 Manifolds $\mathcal{SPD}(n)$, $\mathcal{GL}(n)$, and $\mathbb{R}^{n\times k}$

This part only summarizes the exponential mapping and logarithmic mapping based on [139]. The reader who is interested in more properties of $\mathcal{SPD}(n)$ and $\mathcal{GL}(n)$ is referred to [126, 120, 13].

First of all, we would like to recall the definitions of the exponential and the logarithm of square matrices.

Definition 2.18 *Given an $n \times n$ matrix A, the exponential of A is defined as*

$$e^A := \sum_{i=0}^{\infty} \frac{A^i}{i!}. \tag{2.46}$$

If $A \in \mathcal{GL}(n)$ and there exists an $n \times n$ matrix B such that $e^B = A$, then B is called the logarithm of A and we write $log(A) = B$

It is shown that the sequence (2.46) is always convergent and the exponential of any matrix is non-singular. The logarithm of a real non-singular matrix, however, does not always exist and, if it exists, is not necessarily unique. For any Hurwitz matrix, there is uniquely one logarithm whose eigenvalues lie in the open strip $(-\pi, \pi)$ which is called *principal logarithm*. For more properties of the exponential and the logarithm of matrices, one can consult, e.g., [42].

Now, pick $X_0, X_1 \in \mathcal{GL}(n)$ and $Z \in T_{X_0}\mathcal{GL}(n)$, the exponential and logarithm are defined as

$$\begin{array}{rccc} Exp_{X_0}: & T_{X_0}\mathcal{GL}(n) & \longrightarrow & \mathcal{GL}(n) \\ & Z & \longmapsto & e^Z X_0 \end{array} \tag{2.47}$$

and

$$\begin{array}{rccc} Log_{X_0}: & \mathcal{GL}(n) & \longrightarrow & T_{X_0}\mathcal{GL}(n) \\ & X_1 & \longmapsto & log(X_1 X_0^{-1}). \end{array} \tag{2.48}$$

For $X_0, X_1 \in \mathcal{SPD}(n)$ and $Z \in T_{X_0}\mathcal{SPD}(n)$, we define

$$\begin{array}{rccc} Exp_{X_0}: & T_{X_0}\mathcal{SPD}(n) & \longrightarrow & \mathcal{SPD}(n) \\ & Z & \longmapsto & X_0^{\frac{1}{2}} e^Z X_0^{\frac{1}{2}} \end{array} \tag{2.49}$$

and

$$\begin{array}{rccc} Log_{X_0}: & \mathcal{SPD}(n) & \longrightarrow & T_{X_0}\mathcal{SPD}(n) \\ & X_1 & \longmapsto & log(X_0^{-\frac{1}{2}} X_1 X_0^{-\frac{1}{2}}). \end{array} \tag{2.50}$$

On $\mathbb{R}^{n\times k}$, its tangent space at any point is itself. Therefore, the exponential and logarithmic mappings are defined quite simply. They are nothing else but the addition and the subtraction. That is, given $X_0, X_1 \in \mathbb{R}^{n\times k}$ and $Z \in T_{X_0}\mathbb{R}^{n\times k}$,

$$\begin{array}{rccc} Exp_{X_0}: & T_{X_0}\mathbb{R}^{n\times k} & \longrightarrow & \mathbb{R}^{n\times k} \\ & Z & \longmapsto & X_0 + Z \end{array} \tag{2.51}$$

and

$$\begin{array}{rccc} Log_{X_0}: & \mathbb{R}^{n\times k} & \longrightarrow & T_{X_0}\mathbb{R}^{n\times k} \\ & X_1 & \longmapsto & X_1 - X_0. \end{array} \tag{2.52}$$

Chapter 3

Approaches to MOR of PDSs

Contents

This chapter presents the state-of-the-art approaches to MOR of PDSs. As we will see, these approaches must be somewhat based on existing MOR methods. That may be a generalization of one method, or a relaxation of one condition. It may also be a combination of one MOR method with some other technique that helps to deal with the dependence on parameters such as interpolation, sensitivity analysis or convex optimization.

3.1 Krylov Subspace Based Methods

The use of Krylov subspace based methods for MOR of PDSs started, to our knowledge, by the article [176], and was then generalized in [43] to the case of more than two parameters. The result was then applied to the computation of the coupling capacitances of a three-conductor model where the distances between conductors may change during the operation.

An effective and reliable algorithm to construct matrices for the projections is a crucial point in this approach. An approximate moment matching method based on projecting on Krylov subspaces was proposed in [55], while a so-called two-directional Arnoldi process was used in [110]. In addition, there are many other suggestions concerning the development of this direction as well as its applications. They can be found, for instance, in [76, 75, 37, 51, 107, 108, 122]. Most part of this section is, however, based only on [176, 43, 55, 110].

3.1.1 Multi-parameter Moment Matching Methods

Consider a PDS of the form

$$\begin{aligned} E\dot{x}(t) &= \Big(A - \sum_{i=1}^{k} p_i A_i\Big)x(t) + Bu(t), x(0) = 0, \\ y(t) &= Cx(t), \end{aligned} \tag{3.1}$$

where $E, A, A_i \in \mathbb{R}^{N\times N}, B \in \mathbb{R}^{N\times m}, C \in \mathbb{R}^{l\times N}$. $p_i \in \Omega_i$ are (time-independent) parameters. Systems of the form (3.1) quite often occur as the spatially discretized versions of parabolic PDEs [152]. The transfer function of (3.1), which depends on both, frequency and parameters, is formulated as

$$H(p_1, \cdots, p_k, s) = C\Big(sE - (A - \sum_{i=1}^{k} p_i A_i)\Big)^{-1} B. \tag{3.2}$$

From (3.2), one can observe that sE and $p_i A_i$ symbolically play the same role. Hence one can write $p_0 A_0$ instead of sE in order to simplify the notation. Applying Neumann series, as long as all the matrix inverses below exist, one can write

$$\begin{aligned} H(p_0, \cdots, p_k) &= C\Big(- A + \sum_{i=0}^{k} p_i A_i\Big)^{-1} B \\ &= C\Big(- A(I - \sum_{i=0}^{k} p_i A^{-1} A_i)\Big)^{-1} B \\ &= -C\Big(I - \sum_{i=0}^{k} p_i A^{-1} A_i\Big)^{-1} A^{-1} B \\ &= -C \sum_{n=0}^{\infty} \Big(\sum_{i=0}^{k} p_i A^{-1} A_i\Big)^{n} A^{-1} B \\ &= -C\Big(I + \sum_{i=0}^{k} p_i A^{-1} A_i + \sum_{i,j=0}^{k} p_i (A^{-1} A_i)(A^{-1} A_j) + \cdots\Big) A^{-1} B \\ &= -CA^{-1}B - \sum_{i=0}^{k} p_i C(A^{-1} A_i) A^{-1} B \\ &\quad - \sum_{i,j=0}^{k} p_i p_j C(A^{-1} A_i)(A^{-1} A_j) A^{-1} B + \cdots \end{aligned} \tag{3.3}$$

The set of parameters $(p_0, \cdots, p_k)$ is treated similarly to the frequency s in parameter-independent cases. As such, (3.3) is nothing else but the expansion of $H(p_0, \cdots, p_k)$ about $(0, \cdots, 0)$. Its coefficients are so-called *mixed moments* or *generalized moments*.

MOR by moment matching applied to the system (3.1) is conducted in a manner corresponding to the conventional method presented in Chapter 2. It involves the

construction of two r-dimensional subspaces, which are represented by two $N \times r$ matrices V and Z, on which the original system is projected. Accordingly, the reduced system will take the form

$$\begin{aligned}\hat{E}\dot{\hat{x}}(t) &= \Big(\hat{A} - \sum_{i=1}^{k} p_i \hat{A}_i\Big)\hat{x}(t) + \hat{B}u(t), \hat{x}(0) = 0,\\ y(t) &= \hat{C}\hat{x}(t),\end{aligned}$$

where $\hat{E} = Z^T EV, \hat{A} = Z^T AV, \hat{A}_i = Z^T A_i V, \hat{B} = Z^T B, \hat{C} = CV$. Matrices V and Z are built in such a way that the reduced transfer function

$$\hat{H}(p_0, \cdots, p_k) = \hat{C}\Big(\hat{E} - \hat{A} + \sum_{i=0}^{k} p_i \hat{A}_i)\Big)^{-1}\hat{B}$$

matches the chosen moments about zero of the transfer function (3.2).

In what follows, for ease of presentation and understanding, we consider the case $k = 1$, i.e., there is one frequency parameter which is still denoted by p_0 and one actual parameter p_1. Denote by f_i^n the $(i+1)^{th}$ coefficient of an n^{th} binomal, i.e., $f_i^n(\xi, \eta) = \xi^{n-i}\eta^i, 0 \le i \le n$, we rearrange terms in (3.3) as follows.

$$\begin{aligned}H(p_0, p_1) &= -C\Big(I - (p_0 A^{-1}A_0 + p_1 A^{-1}A_1)\Big)^{-1} A^{-1}B\\ &= -C\sum_{n=0}^{\infty}\Big(p_0 A^{-1}A_0 + p_1 A^{-1}A_1\Big)^n A^{-1}B\\ &= -C\sum_{n=0}^{\infty}\Big(\sum_{i=0}^{n} f_i^n(A^{-1}A_0, A^{-1}A_1)p_0^i p_1^{n-i}\Big)A^{-1}B\\ &= \sum_{n=0}^{\infty}\sum_{i=0}^{n}\Big(-Cf_i^n(A^{-1}A_0, A^{-1}A_1)A^{-1}B\Big)p_0^{n-i}p_1^i.\end{aligned}$$

The moment corresponding to $p_0^{n-i}p_1^i$ is denoted by M_i^n, i.e.,

$$M_i^n := -Cf_i^n(A^{-1}A_0, A^{-1}A_1)A^{-1}B.$$

If there is one parameter, the function f_i^n is merely the n-th power. In our case, the formulation is rather more complicated. It was proven that f_i^n satisfies the following recursion expression.

Lemma 3.1 *([176], Theorem 1)*

$$\begin{aligned}f_i^n(\Phi_1, \Phi_2) &= \Phi_2 f_{i-1}^{n-1}(\Phi_1, \Phi_2) + \Phi_1 f_i^{n-1}(\Phi_1, \Phi_2)\\ &= f_{i-1}^{n-1}(\Phi_1, \Phi_2)\Phi_2 + f_i^{n-1}(\Phi_1, \Phi_2)\Phi_1, i \le n = 0, 1, 2, \cdots\end{aligned} \tag{3.4}$$

Note that in this lemma and what follow,

$$\begin{aligned}f_0^0(\Phi_1, \Phi_2) &= I,\\ f_0^1(\Phi_1, \Phi_2) &= \Phi_1,\\ f_1^1(\Phi_1, \Phi_2) &= \Phi_2,\\ f_i^n(\Phi_1, \Phi_2) &= 0, \text{ for } i > n \text{ or } i < 0.\end{aligned}$$

More properties are provided in the following statements.

Lemma 3.2 *([176], Theorem 2)*

$$f_i^n(\Psi\Phi_1, \Psi\Phi_2)\Psi = \Psi f_i^n(\Phi_1\Psi, \Phi_2\Psi), i \leq n = 0, 1, 2, \cdots . \tag{3.5}$$

Lemma 3.3 *([176], Theorem 3)*

$$f_i^n(\Phi_1, \Phi_2) = \sum_{\alpha=0}^{\beta} f_{\beta-\alpha}^{\beta}(\Phi_1, \Phi_2) f_{i-\beta+\alpha}^{n-\beta}(\Phi_1, \Phi_2), n = 0, 1, 2, \cdots . \tag{3.6}$$

Based on these lemmas, a theorem on moment matching can be proven. However, we first need to generalize the definition of conventional block Krylov subspaces. For systems depending on two parameters, given two matrices Φ_1, Φ_2 and a full-rank matrix Λ, the so-called *generalized block Krylov subspace* $\mathcal{K}_J(\Phi_1, \Phi_2, \Lambda)$ is defined as

Definition 3.1

$$\begin{aligned}\mathcal{K}_J(\Phi_1, \Phi_2, \Lambda) &:= \mathrm{colsp}\Big\{ \bigcup_{n=0}^{J} \bigcup_{i=0}^{n} f_i^n(\Phi_1, \Phi_2)\Lambda \Big\} \\ &:= \mathrm{colsp}\Big\{ \Lambda, \Phi_1\Lambda, \Phi_2\Lambda, \Phi_1^2\Lambda, \Phi_1\Phi_2\Lambda, \Phi_2\Phi_1\Lambda, \Phi_2^2\Lambda, \cdots \Big\}.\end{aligned}$$

We are now ready to state the first result on a multi-parameter moment matching method.

Lemma 3.4 *If $V \in \mathbb{R}^{N\times r}$ is constructed such that $\mathcal{K}_J(A^{-1}A_0, A^{-1}A_1, A^{-1}B) \subseteq \mathrm{colsp}\{V\}$ and Z is a matrix such that Z^TAV is non-singular, then*

$$f_i^n(A^{-1}A_0, A^{-1}A_1)A^{-1}B = V f_i^n(\hat{A}^{-1}\hat{A}_0, \hat{A}^{-1}\hat{A}_1)\hat{A}^{-1}\hat{B} \tag{3.7}$$

for $0 \leq i \leq n \leq J$.

Proof We follow the strategy in [176], which is given only for SISO systems, by using an induction process. First, let us define $W^T := (Z^TAV)^{-1}Z^TA$. It is evident that $W^TV = I$. When $n = 0$, (3.7) is simply

$$A^{-1}B = V\hat{A}^{-1}\hat{B}. \tag{3.8}$$

Indeed, transforming the right hand side of the above equality, we get

$$\begin{aligned}V\hat{A}^{-1}\hat{B} &= V(Z^TAV)^{-1}Z^TB \\ &= V(Z^TAV)^{-1}Z^TAA^{-1}B \\ &= VW^TA^{-1}B.\end{aligned}$$

By hypothesis, $\mathrm{colsp}\{A^{-1}B\} \subseteq \mathrm{colsp}\{V\}$ and $W^TV = I$, which implies (3.8).

Suppose that (3.7) is fulfilled for all $0 \le i \le n-1$, applying (3.4) we have

$$
\begin{aligned}
&V f_i^n(\hat{A}^{-1}\hat{A}_0, \hat{A}^{-1}\hat{A}_1)\hat{A}^{-1}\hat{B} \\
&= V\Big(\hat{A}^{-1}\hat{A}_1 f_{i-1}^{n-1}(\hat{A}^{-1}\hat{A}_0, \hat{A}^{-1}\hat{A}_1) + \hat{A}^{-1}\hat{A}_0 f_i^{n-1}(\hat{A}^{-1}\hat{A}_0, \hat{A}^{-1}\hat{A}_1)\Big)\hat{A}^{-1}\hat{B} \\
&= V\Big(\hat{A}^{-1}Z^T A_1 V f_{i-1}^{n-1}(\hat{A}^{-1}\hat{A}_0, \hat{A}^{-1}\hat{A}_1) + \hat{A}^{-1}Z^T A_0 V f_i^{n-1}(\hat{A}^{-1}\hat{A}_0, \hat{A}^{-1}\hat{A}_1)\Big)\hat{A}^{-1}\hat{B} \\
&= V\Bigg(\hat{A}^{-1}Z^T A_1\Big(V f_{i-1}^{n-1}(\hat{A}^{-1}\hat{A}_0, \hat{A}^{-1}\hat{A}_1)\hat{A}^{-1}\hat{B}\Big) \\
&\quad + \hat{A}^{-1}Z^T A_0\Big(V f_i^{n-1}(\hat{A}^{-1}\hat{A}_0, \hat{A}^{-1}\hat{A}_1)\hat{A}^{-1}\hat{B}\Big)\Bigg) \\
&\text{(induction hypothesis)} \\
&= V\Bigg(\hat{A}^{-1}Z^T A_1\Big(f_{i-1}^{n-1}(A^{-1}A_0, A^{-1}A_1)A^{-1}B\Big) \\
&\quad + \hat{A}^{-1}Z^T A_0\Big(f_i^{n-1}(A^{-1}A_0, A^{-1}A_1)A^{-1}B\Big)\Bigg) \\
&= V\Bigg((Z^T AV)^{-1}Z^T A A^{-1} A_1\Big(f_{i-1}^{n-1}(A^{-1}A_0, A^{-1}A_1)A^{-1}B\Big) \\
&\quad + (Z^T AV)^{-1}Z^T A A^{-1} A_0\Big(f_i^{n-1}(A^{-1}A_0, A^{-1}A_1)A^{-1}B\Big)\Bigg) \\
&= V(Z^T AV)^{-1}Z^T A\Big(A^{-1}A_1 f_{i-1}^{n-1}(A^{-1}A_0, A^{-1}A_1) \\
&\quad + A^{-1}A_0 f_i^{n-1}(A^{-1}A_0, A^{-1}A_1)\Big)A^{-1}B \\
&\overset{(3.4)}{=} VW^T f_i^n(A^{-1}A_0, A^{-1}A_1)A^{-1}B \\
&(\operatorname{colsp}\{f_i^n(A^{-1}A_0, A^{-1}A_1)A^{-1}B\} \subseteq \operatorname{colsp}\{V\}) \\
&= f_i^n(A^{-1}A_0, A^{-1}A_1)A^{-1}B.
\end{aligned}
$$

∎

Lemma 3.4 provides a theoretical base for the one-sided multi-parameter projection method. Indeed, multiplying C to the left of (3.7) implies the equality of moments of the reduced system and the corresponding ones of the full order system. So far, the information to build up the projection matrices is only taken from the input matrix B. Output matrix C has not been exploited yet. In fact, C makes the same contribution. It is shown in the following lemma. We skip the proof due to the similarity to the proof of Lemma 3.4.

Lemma 3.5 *If $Z \in \mathbb{R}^{N\times r}$ satisfies $\mathcal{K}_J(A^{-T}A_0^T, A^{-T}A_1^T, A^{-T}C^T) \subseteq \operatorname{colsp}\{Z\}$ and V is a matrix such that $Z^T AV$ is non-singular, then*

$$CA^{-1}f_i^n(A_0A^{-1}, A_1A^{-1}) = \hat{C}\hat{A}^{-1}f_i^n(\hat{A}_0\hat{A}^{-1}, \hat{A}_1\hat{A}^{-1})Z^T \tag{3.9}$$

for $0 \le i \le n \le J$.

By these results, we come to the main theorem, which was proven in [176] but only for SISO systems.

Theorem 3.6 *If* $V, Z \in \mathbb{R}^{N\times r}$ *satisfy* $\mathcal{K}_{J_B}(A^{-1}A_0, A^{-1}A_1, A^{-1}B) \subseteq \operatorname{colsp}\{V\}$, $\mathcal{K}_{J_C}(A^{-T}A_0^T, A^{-T}A_1^T, A^{-T}C^T) \subseteq \operatorname{colsp}\{Z\}$ *and* Z^TAV *is non-singular, then*

$$Cf_i^n(A^{-1}A_0, A^{-1}A_1)A^{-1}B = \hat{C}f_i^n(\hat{A}^{-1}\hat{A}_0, \hat{A}^{-1}\hat{A}_1)\hat{A}^{-1}\hat{B} \tag{3.10}$$

for $0 \le i \le n \le J_B + J_C + 1$.

Proof The case $n = 0$ is trivial, as it is a consequence of Lemma 3.4. Now, for any $n \le J_B + J_C + 1$, one can always find $0 \le i_B \le J_B$ and $0 \le i_C \le J_C$ such that $n = j_B + j_C + 1$. By (3.6) we have

$$\begin{aligned}
&Cf_i^n(A^{-1}A_0, A^{-1}A_1)A^{-1}B\\
&\overset{(3.3)}{=} C\sum_{\alpha=0}^{j_C+1} f_{j_C+1-\alpha}^{j_C+1}(A^{-1}A_0, A^{-1}A_1)f_{i-j_C-1+\alpha}^{j_B}(A^{-1}A_0, A^{-1}A_1)A^{-1}B\\
&\overset{(3.4)}{=} C\sum_{\alpha=0}^{j_C+1}\Big(f_{j_C-\alpha}^{j_C}(A^{-1}A_0, A^{-1}A_1)A^{-1}A_1 + f_{j_C+1-\alpha}^{j_C}(A^{-1}A_0, A^{-1}A_1)A^{-1}A_0\Big)\\
&\quad f_{i-j_C-1+\alpha}^{j_B}(A^{-1}A_0, A^{-1}A_1)A^{-1}B\\
&\overset{(3.5)}{=} \sum_{\alpha=0}^{j_C+1}\Big(CA^{-1}f_{j_C-\alpha}^{j_C}(A^{-1}A_0, A^{-1}A_1)A_1 + CA^{-1}f_{j_C+1-\alpha}^{j_C}(A^{-1}A_0, A^{-1}A_1)A_0\Big)\\
&\quad f_{i-j_C-1+\alpha}^{j_B}(A^{-1}A_0, A^{-1}A_1)A^{-1}B.
\end{aligned}$$

Applying Lemma 3.5 for the first two terms, Lemma (3.4) for the third one of the above expression yields

$$\begin{aligned}
&Cf_i^n(A^{-1}A_0, A^{-1}A_1)A^{-1}B\\
&= \sum_{\alpha=0}^{j_C+1}\Big(\hat{C}\hat{A}^{-1}f_{j_C-\alpha}^{j_C}(\hat{A}^{-1}\hat{A}_0, \hat{A}^{-1}\hat{A}_1)Z^TA_1 + \hat{C}\hat{A}^{-1}f_{j_C+1-\alpha}^{j_C}(\hat{A}^{-1}\hat{A}_0, \hat{A}^{-1}\hat{A}_1)Z^TA_0\Big)\\
&\quad Vf_{i-j_C-1+\alpha}^{j_B}(\hat{A}^{-1}\hat{A}_0, \hat{A}^{-1}\hat{A}_1)\hat{A}^{-1}\hat{B}\\
&= \hat{C}\sum_{\alpha=0}^{j_C+1}\Big(\hat{A}^{-1}f_{j_C-\alpha}^{j_C}(\hat{A}^{-1}\hat{A}_0, \hat{A}^{-1}\hat{A}_1)\hat{A}_1 + \hat{A}^{-1}f_{j_C+1-\alpha}^{j_C}(\hat{A}^{-1}\hat{A}_0, \hat{A}^{-1}\hat{A}_1)\hat{A}_0\Big)\\
&\quad f_{i-j_C-1+\alpha}^{j_B}(\hat{A}^{-1}\hat{A}_0, \hat{A}^{-1}\hat{A}_1)\hat{A}^{-1}\hat{B}\\
&\overset{(3.5)}{=} \hat{C}\sum_{\alpha=0}^{j_C+1}\Big(f_{j_C-\alpha}^{j_C}(\hat{A}^{-1}\hat{A}_0, \hat{A}^{-1}\hat{A}_1)\hat{A}^{-1}\hat{A}_1 + f_{j_C+1-\alpha}^{j_C}(\hat{A}^{-1}\hat{A}_0, \hat{A}^{-1}\hat{A}_1)\hat{A}^{-1}\hat{A}_0\Big)\\
&\quad f_{i-j_C-1+\alpha}^{j_B}(\hat{A}^{-1}\hat{A}_0, \hat{A}^{-1}\hat{A}_1)\hat{A}^{-1}\hat{B}\\
&\overset{(3.4)}{=} \hat{C}\sum_{\alpha=0}^{j_C+1} f_{j_C+1-\alpha}^{j_C+1}(\hat{A}^{-1}\hat{A}_0, \hat{A}^{-1}\hat{A}_1)f_{i-j_C-1+\alpha}^{j_B}(\hat{A}^{-1}\hat{A}_0, \hat{A}^{-1}\hat{A}_1)A^{-1}B\\
&\overset{(3.6)}{=} \hat{C}f_i^n(\hat{A}^{-1}\hat{A}_0, \hat{A}^{-1}\hat{A}_1)\hat{A}^{-1}\hat{B}.
\end{aligned}$$

∎

Remark If one prefers expanding the transfer function at $(\rho_0, \cdots, \rho_k)$ rather than at $(0, \cdots, 0)$, $H(p_0, \cdots, p_k)$ in (3.3) can be written as

$$\begin{aligned} H(p_0, \cdots, p_k) &= C\Big(-\big(A - \sum_{j=0}^{k} \rho_j A_j\big) + \sum_{i=0}^{k} (p_i - \rho_i) A_i \Big)^{-1} B \\ &= -C\Big(I - \sum_{i=0}^{k} (p_i - \rho_i)\big(A - \sum_{j=0}^{k} \rho_j A_j\big)^{-1} A_i\Big)\Big(A - \sum_{j=0}^{k} \rho_j A_j\Big)^{-1} B \\ &= -C\Big(A - \sum_{j=0}^{k} \rho_j A_j\Big)^{-1} B \\ &\quad - C \sum_{i=0}^{k} \Big(A - \sum_{j=0}^{k} \rho_j A_j\Big)^{-1} A_i \Big(A - \sum_{j=0}^{k} \rho_j A_j\Big)^{-1} B (p_i - \rho_i) + \cdots \end{aligned}$$

and replace A by $A - \sum_{i=0}^{k} \rho_j A_j$ and p_i by $p_i - \rho_i$ in the rest of this sum.

As mentioned, the generalization of the two-parameter case to a multi-parameter case was performed in [43]. In our framework, if we work for instance with three parameters, the generalized block Krylov subspace in Definition 3.1 is defined correspondingly via $f(\Phi_1, \Phi_2, \Phi_3, \Lambda)$. One can easily recognize the rule for the products $\Phi_1^{\alpha}\Phi_2^{\beta}\Phi_3^{\gamma}$. They are nothing else but the terms of powers of the trinomial $\Phi_1+\Phi_2+\Phi_3$ over a non-commutative ring. We skip the presentation of this straightforward generalization here.

We would like to emphasize that Lemma 3.4, Lemma 3.5 and Theorem 3.6 were proven in [176, 43], but only for SISO cases. The authors of [43] mentioned the extension of the result to the multi-input case, but rather entrywise. That is, the construction of V, which is well-established for the single-input case, is repeated with all columns of the input matrix. The union of the resulting matrices is then the sought-after projection matrix. It is worthwhile to note that only one-sided projection was considered in this approach. From our point of view, Theorem 3.6, which is using the two-sided projection and applicable for MIMO systems, cannot be directly proven using that approach. Therefore, a systematic presentation of this subject in this subsection might be regarded as our contribution to this direction.

Projection matrices V, Z in Theorem 3.6 are constructed in order to match the generalized moments at one point. To match moments at more points, following the rational interpolation approach [70], one has to compute matrices corresponding to those points and the required matrices are the union of computed matrices. □

3.1.2 Some Other Developments

Although the Arnoldi process is well-established for the construction of Krylov subspaces in MOR, it works only with standard Krylov subspaces. Therefore, invoking the Arnoldi process in the present problem means one has to modify the construction of the subspace or even redefine the moments. This subsection will present

some of these developments and some other related ideas.

As in the previous parts, we consider a PDS of the form (3.1) depending on only one parameter p,

$$\begin{aligned} E\dot{x}(t) &= (A - pA_p)x(t) + Bu(t), x(0) = 0, \\ y(t) &= Cx(t), \end{aligned} \tag{3.11}$$

and will discuss the case of multi parameters at the end. The transfer function therefore reads

$$H(s,p) = C\Big(sE - (A - pA_p)\Big)^{-1}B.$$

Unlike the multi-parameter moment matching method, the frequency parameter s and the parameter p are treated differently. This is because in this approach, we do not consider $H(s,p)$ as a function of two independent variables when expanding it. Alternatively, we first expand $H(s,p)$ as a function of one variable, and put the other in the coefficients. Clearly, the parameter that is chosen first, in our case the frequency s, will play the dominant role [1]. Accordingly, we have

$$\begin{aligned} H(s,p) &= C\Big(sE - (A - pA_p)\Big)^{-1}B \\ &= -C\Big(I - s(A - pA_p)^{-1}E\Big)^{-1}(A - pA_p)^{-1}B \\ &= -C\sum_{i=0}^{\infty}\Big((A - pA_p)^{-1}E\Big)^{i}(A - pA_p)^{-1}Bs^i. \end{aligned}$$

Since moments of $H(s,p)$ depend on p, one can by no means match them with Krylov subspace techniques. If we, however, accept to match these moments in an approximation sense, we can carry out the task by matching the so-called *submoments*. Let us write

$$M_i = \Big((A - pA_p)^{-1}E\Big)^{i}(A - pA_p)^{-1}B, i = 0, 1, 2, \cdots$$

Consider first

$$\begin{aligned} M_0 &= (A - pA_p)^{-1}B \\ &= \Big(A(I - pA^{-1}A_p)\Big)^{-1}B \\ &= \sum_{j=0}^{\infty}(A^{-1}A_p)^jA^{-1}Bp^j, \end{aligned}$$

in which the matrix $(A^{-1}A_p)^jA^{-1}B$ are called submoments. One can observe that the expansion of M_0 has the same structure as expansion of the transfer function of

[1] Separating parameters and frequency in matching moments of transfer functions is also used in [108]. However, they suggested explicitly matching moments for parameters first and then implicitly matching moments for the frequency.

an LTI system. Therefore, one can employ the Krylov subspace method to match the submoments $(A^{-1}A_p)^j A^{-1}B$. To this end, we define

$$\text{colsp}\{V_0\} = \mathcal{K}_{i_0}(A^{-1}A_p, A^{-1}B)$$

with the intention of approximating M_0 by matching its first i_0 submoments of M_1, where

$$\begin{aligned} M_1 &= (A - pA_p)^{-1}E(A - pA_p)^{-1}B \\ &= \sum_{j_1=0}^{\infty}(A^{-1}A_p)^{j_1}A^{-1}E\sum_{j_0=0}^{\infty}(A^{-1}A_p)^{j_0}A^{-1}Bp^{j_1}p^{j_0}. \end{aligned}$$

By its definition, M_1 contains information on M_0. This must be taken into account during the construction of V_1. Note that we did not record full data of M_0, but only an approximation of it through V_0. For this reason, we define

$$B_1 := EV_0$$

and define the subspace

$$\text{colsp}\{V_1\} := \mathcal{K}_{i_1}(A^{-1}A_p, A^{-1}B_1),$$

And analogously,

$$\text{colsp}\{V_2\} = \mathcal{K}_{i_2}(A^{-1}A_p, A^{-1}B_2).$$

Suppose we proceed in this way k times, the projection matrix needed for the reduction is V, such that

$$\text{colsp}\{V\} = \text{colsp}\{V_0, V_1, \cdots, V_k\}.$$

The reduced system is then determined by a one-sided projection. That is,

$$\begin{aligned} &\hat{E}\dot{\hat{x}} = (\hat{A} - p\hat{A}_p)\hat{x}(t) + \hat{B}u(t), \hat{x}(0) = 0, \\ &\hat{y}(t) = \hat{C}\hat{x}(t), \end{aligned} \tag{3.12}$$

where $\hat{E} = V^TEV, \hat{A} = V^TAV, \hat{A}_p = V^TA_pV, \hat{B} = V^TB, \hat{C} = CV$. The following theorem details how well system (3.12) approximates its original system.

Theorem 3.7 *([55], Theorem 2) The first $i_0, i_1, \cdots, i_k$ submoments of $M_0, M_1, \cdots, M_k$, respectively of the original system (3.11) are matched by the corresponding quantities of the transfer function of the reduced system (3.12).*

There are some points we would like to stress here. First, like many other Krylov subspace methods, there is no strategy for choosing the size of the subspaces $V_j, j = 0, \cdots, k$. It is empirically decided, for instance, based on how well one wants to approximate the moments $M_j, j = 0, \cdots, k$.

Second, since V_j determines the size of the starting block B_{j+1} in constructing V_{j+1} and this fact repeats constantly, the size of the projection matrix V, which is

likely the sum of the sizes of all V_j, increases rapidly together with j. For example, assume that a large MIMO system with 3 inputs needs to be treated and suppose that no deflation occurs during the reduction. In order to approximate moments, we match, e.g., 10 submoments for all M_j. Then V_j will have 3×10^j columns and therefore we will get a reduced system of the order $3 \times 10 \times (10^{k+1} - 1)/(10 - 1)$. In practice, however, one can *hope*, as the numerical example in [55] showed, that matching only a few submoments is good enough for the reduced system to capture the main behavior of the full-order one.

Third, an extension of this approach to multi-parameter cases is, although formally straightforward, actually cumbersome. After expanding the transfer function with respect to frequency s, one can approximate the moment M_j in two ways: defining submoments as in the multi-parameter cases presented in Subsection 3.1.1 or expanding and matching the submoments with respect to one parameter after another. Whichever way to be chosen, the procedure for approximating higher order (second or third) soon becomes unfeasibly complicated.

The left hand side matrix of system (3.1) may depend on parameters as well, for instance

$$E + \sum_{i=1}^{k} p_i E_i. \tag{3.13}$$

This kind of system arises during the simulation of parametrized interconnect networks or MEMS [113, 3, 151, 110]. In the simplest case when only one parameter is concerned, the transfer function takes the form

$$\begin{aligned} H(s,p) &= C\Big(s(E+pE_p) - (A - pA_p)\Big)^{-1} B \\ &= C\Big(-(A-pA_p)\Big(I - s(A-pA_p)^{-1}(E+pE_p)\Big)\Big)^{-1} B \\ &= -C\Big(I - s(A-pA_p)^{-1}(E+pE_p)\Big)^{-1}(A-pA_p)^{-1}B \\ &= -C\sum_{i=0}^{\infty}\Big((A-pA_p)^{-1}(E+pE_p)\Big)^{i}(A-pA_p)^{-1}Bs^i. \end{aligned} \tag{3.14}$$

The presence of E_p, if we still follow the approach above, accelerates the increase of the dimension of V_j. Turning back to the example we just mentioned in the previous part, the projection matrix V will have $3\times(2\times10)\times\big((2\times10)^{k+1}-1\big)/\big((2\times10)-1\big)$ columns.

From (3.14), one way to avoid this situation is to define an extra parameter $q = sp$ and end up with a three-parameter-dependent transfer function

$$H(s,p,q) = C(-A + sE + qE_p + pA_p)^{-1}B.$$

This leads to a complicated situation related to the extension to multi-parameter cases mentioned in the previous point. Of course, the multi-parameter method can be invoked, but we want to carry on with Arnoldi-typed approach.

We rewrite the transfer function (3.14) as

$$\begin{aligned} H(s,p) &= C(-(A-A_p)+s(E+pE_p))^{-1}B \\ &= \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} M_j^i s^j p^i, \end{aligned}$$

where moments M_j^i are of the form $M_j^i = -C\mu_j^i$. It was shown in [110] that the vector μ_j^i satisfies the so-called *two-directional* recurrence

$$A\mu_j^i = E\mu_{j-1}^i + A_p\mu_j^{i-1} + E_p\mu_{j-1}^{i-1}, \tag{3.15}$$

in which $\mu_0^0 = A^{-1}B$ and $\mu_j^i = 0$ for all negative indices i, j.

Following the moment matching method, if we would like to match $(n+1)\times(k+1)$ moments $M_j^i, 0 \le i \le n, 0 \le j \le k$, by a one-sided projection, we have to build a matrix V such that

$$\text{colsp}\{V\} = \text{span}\{M_j^i, 0 \le i \le n, 0 \le j \le k\}. \tag{3.16}$$

It remains to stably and effectively construct V. For SISO systems, the authors of [110] proposed what they called PIMTAP (Parametrized Interconnect Macromodeling via Two-directional Arnoldi Process) as follows.

Rearrange the set of vectors μ_j^i as

$$\begin{bmatrix} \mu_0^0 & \mu_1^0 & \cdots & \mu_k^0 \\ \mu_0^1 & \mu_1^1 & \cdots & \mu_k^1 \\ \cdots & \cdots & \cdots & \cdots \\ \mu_0^n & \mu_1^n & \cdots & \mu_k^n \end{bmatrix}. \tag{3.17}$$

Denote by R_i the full-rank matrix spanning the same subspace as the one spanned by the i-th row of (3.17), i.e.,

$$\text{colsp}\{R_i\} = \text{span}\{\mu_0^{i-1}, \mu_1^{i-1}, \cdots, \mu_k^{i-1}\}.$$

Matrix V is then constructed by adding R_i step by step

$$\begin{aligned} &\text{colsp}\{V_i\} = \text{colsp}\{V_{i-1} \quad R_i\},\ i = 1, \cdots, n+1, \\ &V_0 = [\,]\text{(empty matrix)}. \end{aligned} \tag{3.18}$$

One can observe that R_1 can be directly built by using a standard Arnoldi process, since it is the standard Krylov subspace $\mathcal{K}_{k+1}(A^{-1}E, A^{-1}B)$. For $i = 2, \cdots, n+1$, let us write

$$\mu_{[j]}^{[i]} = \left[\begin{array}{c} \mu_{j-1}^0 \\ \mu_{j-1}^1 \\ \vdots \\ \hline \\ \mu_{j-1}^{i-1} \end{array}\right] = \left[\begin{array}{c} \mu_{[j]}^{[i-1]} \\ \hline \\ \mu_{j-1}^{i-1} \end{array}\right] \in \mathbb{R}^{iN}, \tag{3.19}$$

$$A_{[i]} = \left[\begin{array}{cc|c} A_{[i-1]} & & \\ [0]^* & A_p & A \end{array}\right],\ E_{[i]} = \left[\begin{array}{cc|c} E_{[i-1]} & & \\ [0]^* & E_p & E \end{array}\right].$$

The notation $[0]^*$ means that the size of this zero matrix is flexible in order to fit the size of the whole matrix. In this notation, $A_{[1]} = A, E_{[1]} = E$. Relation (3.15) can be rewritten in the form of a linear expression

$$A_{[i]}\mu_{[j]}^{[i]} = E_{[i-1]}\mu_{[j-1]}^{[i]}, j = 2, \cdots, k+1.$$

This suggests that

$$\text{colsp}\{W_{[i]}\} := \text{span}\left\{\mu_{[1]}^{[i]}, \cdots, \mu_{[k+1]}^{[i]}\right\} = \mathcal{K}_{k+1}\left(A_{[i]}^{-1}E_{[i]}, \mu_{[1]}^{[i]}\right).$$

We however do not need the entire $W_{[i]}$ but only the data that relate to $\left\{\mu_0^{i-1}, \cdots, \mu_k^{i-1}\right\}$. If we decompose $W_{[i]}$ as

$$W_{[i]} = \begin{bmatrix} W_{[i]}^1 \\ W_{[i]}^2 \end{bmatrix},$$

where $W_{[i]}^1$ consists of the first $(i-1)N$ rows of $W_{[i]}$ and $W_{[i]}^2$ consists of the last N rows, it follows by (3.19) that

$$\text{colsp}\{W_{[i]}^2\} = \text{span}\{R_i\}.$$

It remains to update the columnwise orthogonal matrix V_i by (3.18). Another method for computing R_i is provided by the same authors in [109].

Since the PIMTAP algorithm includes only the standard Arnoldi procedure and the Gram-Schmidt re/orthogonalization, the computation is well-behaved. It was also shown that the reduced system constructed by PIMTAP matches all moments $M_j^i, 0 \le i \le n, 0 \le j \le k$. A generalization to the multi-parameter cases was also presented. The interested reader is referred to the aforementioned articles.

Another noteworthy approach is based on direct computation of moments in the frequency domain [76, 75, 32]. The approaches were proposed to deal with linear subnetworks, which in the most general case can be expressed as

$$Y(s,p)X(s,p) = b, \tag{3.20}$$

where $Y(s,p) = -A(p) + sE(p) \in \mathbb{R}^{N\times N}, b \in \mathbb{R}^N$ stands for the network excitation. Denote by M_i^s and M_i^p the moments of X with respect to s and p at $(s,p) = (0,p_0)$, respectively. They can be computed by

$$\begin{aligned} A(p_0)M_i^s &= E(p_0)M_{i-1}^s, \\ -A(p_0)M_0^s &= b, \\ A(p_0)M_i^p &= -\sum_{j=1}^{i} \frac{\frac{\partial^j}{\partial p^j}A(p)\,|_{p=p_0}\, M_{i-j}^p}{j!}, \\ -A(p_0)M_0^p &= b. \end{aligned} \tag{3.21}$$

The idea of the moment matching method suggests that the projection matrix V should be constructed such that

$$\text{colsp}\{V\} = \text{span}\{M_0^s, \cdots, M_n^s, M_0^p, \cdots, M_k^p\}. \tag{3.22}$$

Then system (3.20), after the projection, becomes

$$\left(s\hat{E}(p) - \hat{A}(p)\right)\hat{X}(p,s) = \hat{b}, \tag{3.23}$$

where $\hat{E}(p) = V^T E(p)V, \hat{A}(p) = V^T A(p)V, \hat{b} = V^T b$. It was proven in [76] that the moments $M_0^s, \cdots, M_n^s, M_0^p, \cdots, M_k^p$ of the original system (3.20) are matched by the corresponding ones of the reduced system (3.23) as long as (3.22) is fulfilled.

This approach was then rediscovered in [122]. In his thesis, the author considered to match only the so-called *pure moments*, which are exactly the moments in (3.21), and ignore the other moments.

3.2 Interpolation of Transfer Functions

Interpolation is quite an old subfield of mathematics. It is used to approximate the values of functions on a set within the range of a given discrete set of points at which the value of the function is known. Applying interpolation on transfer functions for PMOR was first proposed in [16] and then improved in [17]. It combines the balanced truncation method and a kind of polynomial interpolation to construct the transfer function for the reduced system. The difficulty lies in conducting it in such a way that it inherits from the standard balanced truncation method all useful properties: stability preservation and a constructible error bound. In the following, some selected facts will be presented. For a complete account, the reader is referred to [17]. Consider a parameter-dependent system

$$\begin{aligned} \dot{x}(t) &= A(p)x(t) + B(p)u(t), x(p,0) = 0, \\ y(t) &= C(p)x(t), \end{aligned} \tag{3.24}$$

where $A(p) \in \mathbb{R}^{N\times N}, B(p) \in \mathbb{R}^{N\times m}, C(p) \in \mathbb{R}^{l\times N}, p$ belongs to a closed, bounded set $\Omega \in \mathbb{R}^d$. Let $\{p_1, \cdots, p_k\} \subset \Omega$ be a chosen discrete set of parameter values. At first, like other interpolation based methods, reduced systems have to be constructed at each p_i:

$$\begin{aligned} \dot{\hat{x}}_i(t) &= \hat{A}_i\hat{x}_i(t) + \hat{B}_i u(t), \\ \hat{y}_i(t) &= \hat{C}_i\hat{x}_i(t), i = 1, \cdots, k. \end{aligned} \tag{3.25}$$

In this approach, balanced truncation was used. Let

$$\hat{H}_i(s) = \hat{C}_i(sI - \hat{A}_i)^{-1}\hat{B}_i$$

denote the reduced transfer function at p_i. We emphasize that in this approach, not the system matrices $\hat{C}_i, \hat{A}_i, \hat{B}_i$ but the reduced transfer function $\hat{H}_i(s)$ is the object to be interpolated. If we think of the parameter-dependent reduced transfer function $\hat{H}(p,s)$ which is computed at each given parameter value p by the same method and the same reduced order, its "values" (actually a function of frequency s) at p_i, $\hat{H}(p_i,s)$ is nothing else but $\hat{H}_i(s)$. Therefore, $\hat{H}_i(s), i = 1, \cdots, k$ are used as

data to reconstruct $\hat{H}(p, s)$ on the whole parameter space. This process is illustrated by Figure 3.1.

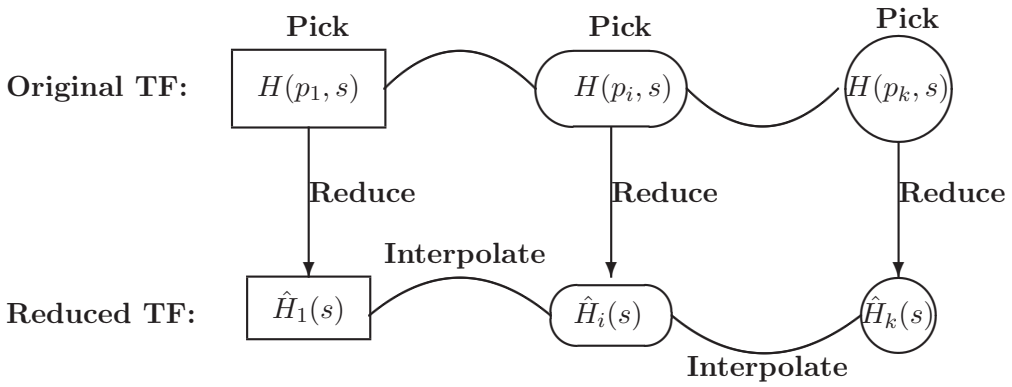

Figure 3.1: Interpolation of transfer function

Depending on the choice of interpolation methods, we receive different resulting transfer functions. In [16, 17], Lagrange, Hermite and sinc were utilized. A common feature of these interpolation types is that the resulting interpolant is directly constructed

$$\sum_{i=1}^{k} \ell_i(p)\hat{H}_i(s). \tag{3.26}$$

This interpolant is used as an approximation of the parameter-dependent reduced transfer function $\hat{H}(p, s)$. Henceforth, we will denote it by the same notation $\hat{H}(p, s)$. Spline interpolation has not been used so far. In the next chapter, we will show our new result on using spline interpolation in the case of single parameter for this purpose.

Stability preservation can be derived quite directly. If (3.24) is stable for all parameter values, each reduced system in (3.25) is stable thanks to the fact that the standard balanced truncation applied to the original system at each p_i preserves the stability. The reduced transfer function (3.26) is a sum of stable functions, and therefore stable.

Formulating a bound for the error between the original transfer function and the reduced one, on the other hand, needs more arguments. First, one has to define a norm for parameter-dependent systems. Since we use balanced truncation, which yields an $\mathcal{H}_\infty$-norm error bound, we define

$$\|H(p, s)\|_\infty := \sup_{p\in\Omega} \|H(p, s)\|_\infty. \tag{3.27}$$

Accordingly, the error bound is principally derived by combining the errors caused by balanced truncation and by interpolation. If Lagrange polynomials on $\Omega = [a, b]$

are used, it was proven in [17] that

$$\|H(p,s)-\hat{H}(p,s)\|_\infty \le \sup_{p\in\Omega}\|R_k(H,p,s)\|_\infty + \varepsilon \sup|\sum_{i=1}^{k}\ell_i(p)|,$$

where $R_k(H,p,s)=\frac{1}{(k+1)!}\left(\frac{\partial^{k+1}}{\partial p^{k+1}}H(\xi(p),s)\right)\prod_{i=1}^{k}(p-p_i), \xi(p)\in[\min p_i, \max p_i]$.

Function $\hat{H}(p,s)$ is a representation of the reduced system in the frequency domain. If the original system is given in state space representation, one may want to have the reduced system in state space representation as well. Classical techniques deriving a state space representation from the transfer function (see [10] and reference therein) are not applicable, since $\hat{H}(p,s)$ depends on the parameter p. A simple self-suggesting approach which makes use of the state space representations of the reduced systems at $p_1,\cdots,p_k$, however, is the following. One can write

$$\begin{aligned}\sum_{i=1}^{k}\ell_i(p)\hat{H}_i(s) &= \sum_{i=1}^{k}(\ell_i(p)\hat{C}_i)(sI_{r_i}-\hat{A}_i)^{-1}\hat{B}_i\\ &= \begin{bmatrix}\hat{C}_1(p) & \hat{C}_2(p) & \cdots & \hat{C}_k(p)\end{bmatrix}\begin{bmatrix}sI_{r_1}-\hat{A}_1 & & & \\ & \ddots & & \\ & & & sI_{r_k}-\hat{A}_k\end{bmatrix}^{-1}\begin{bmatrix}\hat{B}_1\\ \vdots\\ \hat{B}_k\end{bmatrix}\\ &= \hat{C}(p)(sI_{r_1+\cdots+r_k}-\hat{A})^{-1}\hat{B},\end{aligned} \tag{3.28}$$

where $\hat{C}_i(p)=\ell_i(p)\hat{C}_i, i=1,...,k, \hat{A}=\mathrm{diag}(\hat{A}_1,\cdots,\hat{A}_k)$, $\hat{B}=[\hat{B}_1^T\cdots\hat{B}_k^T]^T$. The required representation reads

$$\begin{aligned}\dot{\hat{x}}(t) &= \hat{A}\hat{x}(t)+\hat{B}u(t),\ \hat{x}(0)=0,\\ \hat{y}(t) &= \hat{C}(p)\hat{x}(t).\end{aligned} \tag{3.29}$$

Remark As mentioned in Chapter 2, balanced truncation is a favorable method, since it provides an error bound, but it is rather expensive due to the solution of Lyapunov equations and the computation of the SVD. Interpolation of transfer functions in this style is therefore computationally expensive, especially for systems with high dimensional parameter space. In such cases, the method soon becomes unaffordable. To deal with the so-called curse of dimensionality of the problem, sparse grids [30] have been used in [17].

3.3 Direct Interpolation of System Matrices

This method, proposed in [130], is applicable to any PDS of the form

$$\begin{aligned}E(p)\dot{x}(t) &= A(p)x(t)+B(p)u(t),\\ y(t) &= C(p)x(t),\end{aligned} \tag{3.30}$$

where $E(p), A(p), \in \mathbb{R}^{N\times N}, B(p) \in \mathbb{R}^{N\times m}, C(p) \in \mathbb{R}^{l\times N}$, p are in $\Omega \subset \mathbb{R}^d$. We only assume that the system matrices can be derived for each given p.

Let $p_i \in \Omega, i = 1, \cdots, k$ be a chosen set of parameter values. At each of these points, one uses a standard MOR method as presented in Chapter 2. Note that all MOR methods can be described in the projection framework by two columnwise orthogonal or bi-orthogonal matrices W, V. Denote by $W_i, V_i \in \mathbb{R}^{N\times r}$ the projection matrices used to reduce system (3.30) at p_i. Accordingly, the reduced system at point p_i reads

$$\begin{aligned} \hat{E}_i\dot{\hat{x}}_i(t) &= \hat{A}_i\hat{x}_i(t) + \hat{B}_iu(t), \\ \hat{y}(t) &= \hat{C}\hat{x}_i(t), \end{aligned} \tag{3.31}$$

where $\hat{E}_i = W_i^T E(p_i)V_i, \hat{A}_i = W_i^T A(p_i)V_i, \hat{B}_i = W_i^T B(p_i), \hat{C}_i = C(p_i)V_i$. Now, the resulting reduced system is constructed via interpolation of its system matrices

$$\begin{aligned} \hat{E} = \sum_{i=1}^{k} f_i(p)\hat{E}_i, \quad \hat{A} = \sum_{i=1}^{k} f_i(p)\hat{A}_i, \\ \hat{B} = \sum_{i=1}^{k} f_i(p)\hat{B}_i, \quad \hat{C} = \sum_{i=1}^{k} f_i(p)\hat{C}_i, \end{aligned} \tag{3.32}$$

where $f_i(p)$ are some weight functions which form a partition of unity $\sum_{i=1}^{k} f_i(p) = 1, \forall p \in \Omega$.

Forming a reduced system like (3.32) from local reduced systems (3.31) only makes sense if all local coordinates $\hat{x}_i$ are equal. More precisely, they have to be the coordinates of vectors with respect to one basis. But the projection matrices $V_i, i = 1, \cdots, k$, used to construct (3.31) are in general pairwise different, which means that the original state vector $x(t)$ is approximated by different vectors $V_i\hat{x}_i$ in different subspaces depending on p_i. Therefore, some additional procedure has to "equalize" the coordinates $\hat{x}_i$ and has to do it in such a way that the input-output behavior of the local reduced systems is not affected.

Clearly, the input-output behavior of (3.31) remains unchanged if we multiply from both sides with two non-singular matrices, say,

$$\begin{aligned} M_i\hat{E}_iT_i^{-1}\dot{\hat{x}}_i^*(t) &= M_i\hat{A}_iT_i^{-1}\hat{x}_i^*(t) + M_i\hat{B}_iu(t), \\ \hat{y}(t) &= \hat{C}_iT_i^{-1}\hat{x}_i^*(t). \end{aligned} \tag{3.33}$$

The matrices T_i are the transformations that change the coordinate from $\hat{x}_i^*$ to $\hat{x}_i$,

$$\hat{x}_i^* = T_i\hat{x}_i.$$

With a choice of T_i, we can change the local coordinate $\hat{x}_i$ freely. This is the crucial tool to turn all $\hat{x}_i, i = 1, \cdots, k$, to a common coordinate $\hat{x}^*$ referring to the same subspace.

We will now analyze the situation in order to find T_i. In the original space, the state vector is, after transforming with T_i, approximated by $z_i = V_iT_i^{-1}\hat{x}_i^*$. These

are, however, vectors that lie on the subspace spanned by the columns of V_i. One wants these vectors to belong to a common subspace, say, spanned by columnwise orthogonal matrix $R \in \mathbb{R}^{N\times r}$. Matrix R is usually chosen in such a way that the resulting reduced system captures the most important dynamics of the original system. The coordinate of z_i with respect to the basis formed by columns of R is now

$$\hat{x}^* = R^T z_i = R^T V_i T_i^{-1} \hat{x}_i^*. \tag{3.34}$$

If we choose $T_i = R^T V_i$, all $\hat{x}_i^*$ will become identical to $\hat{x}^*$ and that is what we would like to have.

Remark The choice of R, in addition to containing important dynamics of the original system, has to guarantee the non-singularity of $R^T V_i, i = 1, \cdots, k$. □

A direct way to collect important dynamics and store in R is as follows. Let

$$V = [V_1\ V_2\ \cdots V_k].$$

and

$$\Phi\Lambda\Psi = V$$

be the SVD of V. Matrix R can be chosen as the first r singular vectors. The cost for computing the SVD of an $N \times kr$ matrix V is quite high; fortunately, this has to be done only once for the whole reduction process. The authors of [130] also mentioned another way to construct R, which gives a better approximation and is flexible with the variation of parameters. However, the SVD must be performed for all new parameter values p. It is, from our opinion, not advisable for real time simulation. We will come back to this issue in Chapter 4.

For the choice of M_i in (3.33), it was advised in [130] to choose $M_i = (W_i^T R)^{-1}$. Accordingly, (3.33) becomes

$$\begin{aligned}
&(W_i^T R)^{-1} W_i^T E(p_i) V_i (R^T V_i)^{-1} \dot{\hat{x}}_i^*(t) \\
&\quad = (W_i^T R)^{-1} W_i^T A(p_i) V_i (R^T V_i)^{-1} \hat{x}_i^*(t) + (W_i^T R)^{-1} W_i^T B(p_i) u(t), \\
&\hat{y}(t) = C(p_i) V_i (R^T V_i)^{-1} \hat{x}_i^*(t).
\end{aligned} \tag{3.35}$$

3.4 Indirect Interpolation of System Matrices

As we learnt in the previous section, interpolation of system matrices must be handled with care due to the incompatibility of local projection bases. Another issue which should be considered when interpolating system matrices is the structure of these matrices. A specific structure represents specific properties. For example, when we are dealing with an ordinary differential equation of the form

$$\dot{x}(t) = Ax(t) + f(t),$$

a common hypothesis is the non-singularity of A. Another example is the symmetric positive definiteness of matrices. It frequently arises in structural analysis and

simulations of mechanical systems. One can easily verify that such properties are, in general, not preserved during the interpolation. Therefore, direct interpolation on sets of structural matrices is not advisable.

The idea to deal with interpolation of structural matrices was first proposed in [6, 5] and then followed by [45]. These works share the same idea with [7] which involves interpolation on Grassmann manifolds. The core of these methods is to interpolate on the tangent spaces to the underlying manifolds, which is composed of these system matrices, rather than on manifolds themselves.

According to the analysis in Section 2.3, such structured matrices constitute differential manifolds, e.g., $\mathcal{SPD}(n), \mathcal{GL}(n), \mathcal{G}(k,n)$. Consequently, tangent spaces to these manifolds exist everywhere. More important, the logarithmic mapping and exponential mappings are well-defined. These are important tools to transfer data between manifolds and their tangent spaces and therefore help to carry out the task. The formulation of these mappings highly depends on their associated manifolds. The manner, the outcome and the application of the methods are therefore quite distinct. In the remainder of this section, for the sake of simplicity, the result derived in [45] will be presented. Some other aspects of the approach will be given as remarks.

Consider the system

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

where $A(p) \in \mathbb{R}^{N\times N}, B(p) \in \mathbb{R}^{N\times m}, C(p) \in \mathbb{R}^{l\times N}, p \in \Omega \subset \mathbb{R}^d$. Matrix $A(p)$ is assumed to be invertible for all p. The dependence of A, B, C on p may be non-linear or even implicit, i.e., as in Section 3.3, no analytic expressions are required.

The reduction method used in this approach is based on POD. At the beginning, it constructs a POD basis $V \in \mathbb{R}^{N\times r}$. Unlike the standard POD method, in which V is built based on the SVD of the set of snapshots taken at different time instants, the variation of parameters must be taken into consideration; the snapshots here are taken corresponding to different parameter values: $p_1, \cdots, p_k$ as well. That is, the total numbers of snapshots is the product of that of time instants and of parameter values. The next step is to approximate the original state $x(t)$ by $V\hat{x}(t)$ and to project the resulting system on the subspace spanned by the left projection matrix $W \in \mathbb{R}^{N\times r}, W^TV = I$. The bi-orthogonality is added in order to keep the reduced system in ordinary form.

Since A, B, C may not be given with explicit dependence on p, or the dependence is too complicated to be handled, one cannot (or can hardly) formally multiply, e.g., $W^TA(p)V$. Even though the evaluations of $A(p), B(p), C(p)$ for each value p are well-performed, the reduced system formed in this way is not suitable for online simulations. Because at each new value of p, to get the reduced system, one has to perform computations whose complexity depends on the full-order N.

To avoid this, it is proposed to reduce the system at the points of a given sample $p_1, \cdots, p_k$ and then use the so-called local pre-computed reduced order models as

data to construct a reduced order model for each new parameter value p. Let us denote by

$$\begin{aligned} \dot{\hat{x}}(t) &= \hat{A}_i\hat{x}(t) + \hat{B}_i u(t), \\ \hat{y}(t) &= \hat{C}_i\hat{x}(t), i = 1, \cdots, k, \end{aligned} \tag{3.36}$$

where $\hat{A}_i = W^T A(p_i)V, \hat{B}_i = W^T A(p_i)$ and $\hat{C}_i = C(p_i)V, i = 1, \cdots, k$, the k pre-computed reduced order models. Now it remains to compute the reduced system at new values p, i.e., the triplet $\hat{A}(p), \hat{B}(p), \hat{C}(p)$, using only these k reduced models (3.36). That is, we follow the trend II for MOR of PDSs.

Interpolation is the first idea to be thought of in such a situation. By hypothesis, $\hat{A}_i$ is non-singular for all i and therefore belongs to manifold $\mathcal{GL}(r)$. Hereinafter, elements of $\mathcal{GL}(r)$ may be referred to as points. Direct interpolation on $\mathcal{GL}(r)$ is doomed to failure due to the easily verified fact that $\mathcal{GL}(r)$ is not closed under addition.

In the following, the interpolation procedure which was first proposed in [7] and reused in [45] on $\mathcal{GL}(r)$ is carried out indirectly on its tangent space.

Step 1 Choose one matrix, say $\hat{A}_1$, as a reference point at which the tangent space $\mathcal{T}_{\hat{A}_1}\mathcal{GL}(r)$ is used. $\hat{A}_1$ is chosen such that the distances from the other points in the sample to $\hat{A}_1$ are not too large.

Step 2 Map all points $\hat{A}_2, \cdots, \hat{A}_k$ from $\mathcal{GL}(r)$ to $\mathcal{T}_{\hat{A}_1}\mathcal{GL}(r)$ by the logarithmic mapping $Log_{\hat{A}_1}$. Here we use the formula (2.48) in Chapter 2

$$Log_{\hat{A}_1}(\hat{A}_i) = \log\left(\hat{A}_i\hat{A}_1^{-1}\right). \tag{3.37}$$

Step 3 With given p, the "vector" associated with $p, \hat{\mathcal{A}}(p) \in \mathcal{T}_{\hat{A}_1}\mathcal{GL}(r)$ is approximated via interpolation of $Log_{\hat{A}_1}(\hat{A}_i)$. If we use, for example, Lagrange interpolation, then

$$\hat{\mathcal{A}}(p) = \sum_{i=1}^{k} \ell_i(p) Log_{\hat{A}_1}(\hat{A}_i),$$

where $\ell_i(p)$ are Lagrangian polynomials associated with the given grid points p_i.

Step 4 Map $\hat{\mathcal{A}}(p)$ back to $\mathcal{GL}(r)$ by the exponential mapping $Exp_{\hat{A}_1}$ defined as in (2.47). This image is considered as an approximation of $\hat{A}(p)$ corresponding to p:

$$\hat{A}(p) = Exp_{\hat{A}_1}(\hat{\mathcal{A}}(p)) := \exp(\hat{\mathcal{A}}(p))\hat{A}_1. \tag{3.38}$$

So far, the interpolation procedure is only proposed for $\hat{A}(p)$. It remains to compute $\hat{B}(p)$ and $\hat{C}(p)$ from $\hat{B}_i$, $\hat{C}_i$. Should the same process be applied? It is worth to recall that there are no constraints imposed on $B(p)$ and $C(p)$. Therefore, as p varies on Ω, $B(p)$ and $C(p)$ do not belong to specified manifolds but actually lie

on subspaces $\mathbb{R}^{N\times m}$ and $\mathbb{R}^{l\times N}$, respectively. As a consequence, we do not need to use the interpolation method that is applied to $A(p)$. Instead, a standard interpolation technique, for instance, spline interpolation or Lagrange interpolation is invoked to compute $\hat{B}(p)$ and $\hat{C}(p)$. It turns out that the standard interpolation is a special case of the framework of interpolation on manifolds. Indeed, applying the four-step procedure mentioned above to $\mathbb{R}^{N\times m}$ and $\mathbb{R}^{l\times N}$ using the logarithmic mapping and the exponential mapping defined as (2.52) and (2.51) will result in the standard interpolation.

The use of the approach mentioned above for simulations is divided into two stages. In the first stage, we compute and store $\hat{A}_i, \hat{B}_i, \hat{C}_i, i = 1, \cdots, k$. This step is rather time consuming, since the computation has complexity of original order N, which is very large. However, this is merely the preparatory step and the requirement of computational speed is therefore not pressing. In the second so-called online stage, given a parameter value $p \notin \{p_1, \cdots, p_k\}$ but in the range of this set, the reduced system corresponding to p need to be computed as fast as possible. This task is done by invoking the 4-step procedure given above. Note that in this stage, we only work with small matrices $\hat{A}_i, \hat{B}_i, \hat{C}_i$, and hence the computational cost is low. The idea of separating the reduction process into two stages, widely-called *offline-online decomposition* derives from solving affinely parameter-dependent problems [172, 117]. This idea is also exploited in Chapter 4 to improve an existing algorithm and therefore discussed more thoroughly in Section 4.2.

Remark The authors of [45] suggested employing the algorithm in [44] for the logarithm of non-singular matrices (3.37) and the technique in [83] for the exponent (3.38).

In [6], the approach was applied to the construction of ROMs for a parameter-dependent mechanical system, whose system matrices are symmetric positive definite. Since it is a second-order differential equation, the presentation is skipped here. The interested reader is referred to the mentioned reference.

It was pointed out in [5, 8] that interpolation of precomputed ROMs by their system matrices, which were computed by projecting on different subspaces at different parameter values, was ill-advised. It was therefore suggested that, before interpolating, the difference between projection subspace bases used for computing ROMs and that for the ROM whose system matrices were chosen as reference point should be minimized. In our opinion, however, putting this step (Step A in [8]) in online stage as mentioned in both these publications is unnecessary. One can observe that, the search for the matrix Q (see [8], (4.4)) is independent of the new value of the parameter set $\mu_{N_\mu+1}$, and Algorithm 1 in [8] for Step A is completely based on the available precomputed data. This step can therefore be carried out in offline stage. □

3.5 Some More References

The aforementioned methods in this chapter cannot cover all approaches to such an active research issue. Here are some developments which were left out. In [25], piecewise-linear moment matching was used to deal with highly nonlinear parameterized systems. In another direction, a *greedy algorithm* [132] was invoked in [29, 77] to effectively select parameter samples and time instants in producing projection spaces. The authors of [77] went even further to give an *a posteriori* error estimation between the reduced and the original output. Also based on POD, the method developed in [81] analyzed the shape sensitivity of the models whose geometries depend on parameters and then used this information to build bases for reduced order models. In [106] a framework for PMOR based on generalized Loewner matrices was proposed. Most recently, parameter-dependent systems were considered as bilinear systems and the $\mathcal{H}_2$-norm MOR method for bilinear systems was invoked to treat the parameter-dependent case in [19]. Also based on $\mathcal{H}_2$-norm MOR methods, the authors of [14] extended the known result to the case of parameter-dependent systems and succeeded to characterize the first-order necessary optimality condition for a local minimizer.

Chapter 4

Main Results

Contents

This chapter presents new results for two different PMOR approaches. In the first section, we consider the construction of a reduced transfer funtion based on balanced truncation and spline interpolation. The use of linear and cubic spline interpolation is investigated and error bounds are derived. The second section presents an improvement for the interpolation of projection subspaces to construct reduced order models of PDSs. This improvement helps to speed up an existing algorithm and enable it to be used in real time. Examples are provided in both sections to illustrate our methods.

4.1 Interpolation of Transfer Function

Consider a PDS $\Sigma(p)$ given by

$$\begin{aligned} \dot{x}(t) &= A(p)x(t) + B(p)u(t),\ x(0) = 0, \\ y(t) &= C(p)x(t), \end{aligned} \tag{4.1}$$

where $A \in \mathbb{R}^{N\times N}, B \in \mathbb{R}^{N\times m}, C \in \mathbb{R}^{l\times N}$ depend on a single parameter $p \in [a, b]$. Denote by $H(p, s)$ its transfer function. System (4.1) is assumed to be stable, reachable and observable for all $p \in [a, b]$. This hypothesis ensures that the balanced truncation works for any values of p in the parameter domain.

The effort to directly find a parameter-dependent balancing transformation is soon realized to be unfeasible. The solution to the generally parametric Lyapunov equations

$$A(p)\mathcal{P} + \mathcal{P}A^T(p) + B(p)B^T(p) = 0,$$
$$A^T(p)\mathcal{Q} + \mathcal{Q}A(p) + C^T(p)C(p) = 0,$$

to the best of our knowledge, is still unknown, not to mention the matrix inverse in formulating ϕ in (2.5). Nevertheless, it is well-formulated when the parameter is fixed. This fact leads to an idea that one can reduce the original system at a given parameter set, say $a = p_0, \cdots, p_k = b$ whose reduced transfer functions are $\hat{H}_i(s)$. After that, $\{(p_i, \hat{H}_i(s)), i = 0, \cdots, k\}$ are available and, as presented in Section 3.2, then used as data in interpolation to construct the reduced transfer function $\hat{H}(p, s)$ over $[a, b]$. In this section, linear and cubic splines will be utilized.

4.1.1 Using Linear Spline Interpolation

We will proceed similarly to Section 3.2. First, a discrete set of parameter values $a = p_0, \cdots, p_k = b$ is chosen. The choice of this set will influence the quality of the reduction. To avoid digression, we discuss this issue at the end.

At each p_i, we reduce system $\Sigma(p_i)$ to the order of r_i by balanced truncation,

$$\dot{\hat{x}}(t) = \hat{A}_i x(t) + \hat{B}_i u(t),$$
$$\hat{y}(t) = \hat{C}_i \hat{x}(t), \ i = 0, \cdots, k$$

and denote by $\hat{H}_i(s)$ the corresponding reduced transfer function.

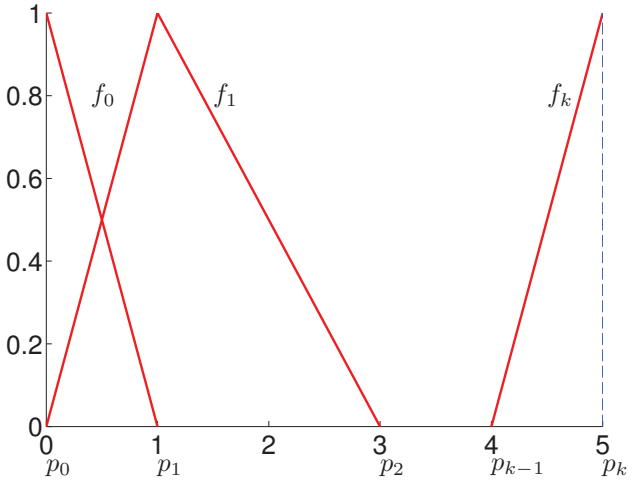

Figure 4.1: Linear splines

For the given grid on $[a, b]$, the basis linear splines

$$f_i(p) = \begin{cases} f_i^- = \dfrac{p - p_{i-1}}{p_i - p_{i-1}} & \text{if } p \in [p_{i-1}, p_i], \\ f_i^+ = \dfrac{p - p_{i+1}}{p_i - p_{i+1}} & \text{if } p \in [p_i, p_{i+1}], \\ 0, & \text{otherwise} \end{cases}$$

for $i = 0, \cdots, k$, in which $f_0(p)$ and $f_k(p)$ are defined to be identical zero outside $[p_0, p_k]$, will be used. By imposing the interpolation conditions, which require that at all interpolation points p_i, the resulting parameter-dependent reduced transfer function is equal to the pre-computed reduced functions $\hat{H}_i(s)$, we derive the interpolant as

$$\hat{H}(p, s) = \sum_{i=0}^{k} f_i(p)\hat{H}_i(s). \tag{4.2}$$

The interpolant $\hat{H}(p, s)$, which is considered as an approximation of the parameter-dependent reduced transfer function, is the external description in the frequency domain of the reduced order system. As described in the Section 3.2, we derive a state space representation by (3.28) and (3.29). Note that we can also attach the dependence on the parameter to the reduced load matrix $\hat{B}$ instead of $\hat{C}(p)$

To measure the quality of PMOR, we use the infinity norm (3.27). In the SISO case $H(p, s)$ is a scalar function, thus we have

$$\|\Sigma(p)\|_\infty = \sup_{p\in[a,b]} \|H(p, s)\|_{\mathcal{H}_\infty} = \sup_{p\in[a,b]} \sup_{s\in\mathbb{C}^+} |H(p, s)|. \tag{4.3}$$

For the sake of convenience, we recall the definition of the Lipschitz condition, which will be used to obtain an error bound.

Definition 4.1 *A parameter-dependent transfer function $H(p, s)$ is said to satisfy the Lipschitz condition with respect to p if there exists a constant L such that*

$$\forall p_1, p_2 \in [a, b], \|H(p_1, s) - H(p_2, s)\|_{\mathcal{H}_\infty} \leq L|p_1 - p_2|. \tag{4.4}$$

The following theorem gives an error bound for the linear spline interpolation based PMOR given above.

Theorem 4.1 *Assume that the reduced system $\hat{\Sigma}(p)$ is constructed from system $\Sigma(p)$ in (4.1) as above and assume moreover that the transfer function $H(p, s)$ of (4.1) satisfies the Lipschitz condition (4.4). Then*

$$\|\Sigma(p) - \hat{\Sigma}(p)\|_\infty \leq Lh + \mathcal{E}, \tag{4.5}$$

where L is the Lipschitz constant defined in (4.4), $h = \max\{p_{i+1} - p_i, i = 0, \cdots, k - 1\}$ and $\mathcal{E} = \max\left\{\|H(p_i, s) - \hat{H}_i(s)\|_{\mathcal{H}_\infty}, i = 0, \ldots, k\right\}$ is the maximal value of errors caused by balanced truncation of the original system at grid points $p_0, \ldots, p_k$.

Proof By the triangle inequality, we have

$$\begin{aligned}
&\Big\|H(p,s)-\hat{H}(p,s)\Big\|_\infty = \Big\|H(p,s)-\sum_{i=0}^{k} f_i(p)\hat{H}_i(s)\Big\|_\infty \\
&\leq \Big\|H(p,s)-\sum_{i=0}^{k} f_i(p)H(p_i,s)\Big\|_\infty + \Big\|\sum_{i=0}^{k} f_i(p)H(p_i,s)-\sum_{i=0}^{k} f_i(p)\hat{H}_i(s)\Big\|_\infty . \quad (4.6)
\end{aligned}$$

Suppose that $p \in [p_i, p_{i+1}]$ and consider the first term in (4.6),

$$\begin{aligned}
&\Big\|H(p,s)-\sum_{i=0}^{k} f_i(p)H(p_i,s)\Big\|_\infty = \Big\|H(p,s)-f_i^+(p)H(p_i,s)-f_{i+1}^-H(p_{i+1},s)\Big\|_\infty \\
&= \Big\|\Big(f_i^+(p)+f_{i+1}^-(p)\Big)H(p,s)-f_i^+(p)H(p_i,s)-f_{i+1}^-(p)H(p_{i+1},s)\Big\|_\infty \\
&= \Big\|f_i^+(p)\Big(H(p,s)-H(p_i,s)\Big)+f_{i+1}^-(p)\Big(H(p,s)-H(p_{i+1},s)\Big)\Big\|_\infty \\
&\leq \Big(f_i^+(p)+f_{i+1}^-(p)\Big)\max_{p\in[p_i,p_{i+1}]}\Big\{\Big\|H(p,s)-H(p_i,s)\Big\|_\infty, \Big\|H(p,s)-H(p_{i+1},s)\Big\|_\infty\Big\} \\
&\leq Lh. \quad (4.7)
\end{aligned}$$

In the above argument, we have made use of the fact that $f_i^+(p)+f_{i+1}^-(p)\equiv 1$. It remains to bound the second term of (4.6) by $\mathcal{E}$.

$$\begin{aligned}
&\Big\|\sum_{i=0}^{k} f_i(p)H(p_i,s)-\sum_{i=0}^{k} f_i(p)\hat{H}_i(s)\Big\|_\infty \\
&= \Big\|f_i^+(p)H(p_i,s)+f_{i+1}^-(p)H(p_{i+1},s)-f_i^+(p)\hat{H}_i(s)-f_{i+1}^-(p)\hat{H}_{i+1}(s)\Big\|_\infty \\
&= \Big\|f_i^+(p)\Big(H(p_i,s)-\hat{H}_i(s)\Big)+f_{i+1}^-(p)\Big(H(p_{i+1},s)-\hat{H}_{i+1}(s)\Big)\Big\|_\infty \\
&\leq \sup_{p\in[p_i,p_{i+1}]}\Big(f_i^+(p)+f_{i+1}^-(p)\Big)\times \\
&\quad \max\Big\{\Big\|H(p_i,s)-\hat{H}_i(s)\Big\|_{\mathcal{H}_\infty}, \Big\|H(p_{i+1},s)-\hat{H}_{i+1}(s)\Big\|_{\mathcal{H}_\infty}\Big\} \\
&\leq \mathcal{E}. \quad (4.8)
\end{aligned}$$

The statement of the theorem is directly implied from (4.6) - (4.8). ∎

Since balanced truncation preserves the stability of the system, all $\hat{H}_i(s), i = 0, \cdots, k$, are stable. Fortunately, this already guarantees the stability of $\hat{H}(p,s)$, because the construction of $\hat{H}(p,s)$ results from $\hat{H}_i(s)$ by addition and multiplication with scalars $f_i(p)$ only.

In order to receive the bound (4.5), the Lipschitz condition plays a key role. In addition, it ensures the finiteness of the infinity norm of a parameter-dependent system $\Sigma(p)$. One question naturally arising is, whether this condition is too strict. We will seek an answer for this question in the following part, first for SISO systems and then for MIMO ones.

The following holds for any complex-valued function of one real variable.

Lemma 4.2 *Let $[a,b]$ denote a closed interval in $\mathbb{R}$ and $f:[a,b]\to\mathbb{C}$. Assume that the first derivative of f is continuous and bounded by the constant M over $[a,b]$, then for all $t_1,t_2\in[a,b]$*

$$|f(t_1)-f(t_2)|\le\sqrt{2}M|t_1-t_2|. \tag{4.9}$$

Proof One can always write

$$f(t)=\alpha(t)+i\beta(t),$$

where $\alpha(t),\beta(t)\in\mathbb{R},t\in[a,b]$. For any $t_1<t_2\in[a,b]$, applying the Mean Value Theorem for $\alpha(t)$ and $\beta(t)$, we have

$$\begin{aligned}
|f(t_1)-f(t_2)| &= |\alpha(t_1)-\alpha(t_2)+i(\beta(t_1)-\beta(t_2))|\\
&= \sqrt{(\alpha(t_1)-\alpha(t_2))^2+(\beta(t_1)-\beta(t_2))^2}\\
&= \sqrt{\alpha'^2(t_0^\alpha)+\beta'^2(t_0^\beta)}|t_1-t_2|,\ \text{for some } t_0^\alpha,t_0^\beta\in(t_1,t_2)\\
&\le \sqrt{(\alpha'^2(t_0^\alpha)+\beta'^2(t_0^\alpha))+(\alpha'^2(t_0^\beta)+\beta'^2(t_0^\beta))}|t_1-t_2|\\
&= \sqrt{|f'(t_0^\alpha)|^2+|f'(t_0^\beta)|^2}|t_1-t_2|\\
&\le \sqrt{2}M|t_1-t_2|.
\end{aligned}$$

∎

Seemingly, one only needs to show that the transfer function has a bounded derivative and then apply Lemma 4.2 to derive the Lipschitz condition. However, the considered situation is more complicated since $H(p,s)$ depends on, in addition to p, the complex variable s, and the upper bound M therefore depends on s. One has to show that there is a common upper bound for all $s\in\mathbb{C}^+$. We have the following assertion.

Lemma 4.3 *In addition to the hypotheses on stability, reachability and observability, we assume moreover that all entries of $A(p),B(p),C(p)$ of the parameter-dependent system (4.1) with $l=m=1$ are continuously differentiable over $[a,b]$, then there is a constant M such that*

$$\left|\frac{\partial H}{\partial p}(p,s)\right|\le M,\forall p\in[a,b],s\in\mathbb{C}^+. \tag{4.10}$$

Proof By assumption, the transfer function can be written in the form of

$$H(p,s)=\frac{s^{N-1}z_{N-1}(p)+s^{N-2}z_{N-2}(p)+\cdots+z_0(p)}{s^N+s^{N-1}w_{N-1}(p)+s^{N-2}w_{N-2}(p)+\cdots+w_0(p)}.$$

Then, the partial derivative of $H(p,s)$ with respect to p has the form

$$\frac{\partial H}{\partial p}(p,s)=\frac{Q_{2N-1}(p,s)}{\underline{Q}_{2N}(p,s)},$$

where $Q_{2N-1}(p,s)$ is a polynomial of variable s, with coefficients depending on p and a degree not greater than $2n-1$, and $\underline{Q}_{2N}(p,s) = \Big(\det(H(p,s))\Big)^2$ is one with the exact degree $2N$. Since all entries of A, B, C are continuously differentiable, the coefficients of $Q_{2N-1}(p,s)$ and $\underline{Q}_{2N}(p,s)$ are bounded on $[a,b]$. On the other hand, the degree of $\underline{Q}_{2N}$ is higher than that of $Q_{2N-1}(p,s)$; this ensures that $\frac{\partial H}{\partial p}(p,s)$ is bounded in a neighborhood of infinity, i.e., for some large real number R, there exists a constant M_1 such that

$$\Big|\frac{\partial H}{\partial p}(p,s)\Big| \leq M_1, \forall p \in [a,b], s \in \mathbb{C}^+, |s| > R. \tag{4.11}$$

In the closed domain $\{(p,s) : p \in [a,b], s \in \mathbb{C}^+, |s| \leq R\}$, $\frac{\partial H}{\partial p}(p,s)$ is continuous. Therefore,

$$\Big|\frac{\partial H}{\partial p}(p,s)\Big| \leq M_2, \forall p \in [a,b], s \in \mathbb{C}^+, |s| \leq R. \tag{4.12}$$

By choosing $M = \max\{M_1, M_2\}$, inequalities (4.11) and (4.12) directly complete the proof. ∎

Combining the above two lemmas implies the following theorem.

Theorem 4.4 *Under the hypotheses of Lemma 4.3, the transfer function of the system (4.1) with $l = m = 1$ satisfies the Lipschitz condition (4.4).*

Proof Recall that for any SISO transfer function,

$$\|H(s)\|_{\mathcal{H}_\infty} = \sup_{s\in\mathbb{C}^+} |H(s)|.$$

Applying Lemma 4.3 implies that the inequality (4.10) holds for the transfer function of (4.1), $H(p,s)$. Note that this is true for any $s \in \mathbb{C}^+$. Then, using Lemma 4.2, we deduce

$$\begin{aligned}\|H(p_1,s) - H(p_2,s)\|_{\mathcal{H}_\infty} &= \sup_{s\in\mathbb{C}^+} |H(p_1,s) - H(p_2,s)| \\ &\leq \sup_{s\in\mathbb{C}^+} \sqrt{2}M|p_1 - p_2| \\ &= \sqrt{2}M|p_1 - p_2|.\end{aligned}$$ ∎

In the SISO case, by making use of (4.3), one can directly apply Lemma 4.2 and Lemma 4.3 to derive Theorem 4.4. However, in the MIMO case these two lemmas are not enough, as the inequalities (4.9) and (4.10) only hold for the entries of the (matrix-valued) transfer function. Meanwhile, the norm appearing in the Lipschitz condition (4.4) dependes on the largest singular value of the matrix $H(p,s)$. To cope with this difficulty, one has to investigate the relationship between the singular values of a matrix and its entries. That inspires us to use a Gerschgorin-type theorem [138]. For the clarity of our discussion it is recalled here.

Theorem 4.5 *([138], Theorem 2). Suppose* $A = (a_{ij}) \in \mathbb{C}^{m\times n}$. *Let*

$$r_i = \sum_{j=1, j\neq i}^{n} |a_{ij}|,\ c_i = \sum_{j=1, j\neq i}^{m} |a_{ji}| \text{ and } s_i = \max(r_i, c_i),\ a_i = |a_{ii}|$$

for $i = 1, 2, \ldots, \min(m, n)$. *For* $m \neq n$, *define*

$$s = \begin{cases} \max\limits_{n+1\leq i\leq m} \left\{ \sum\limits_{j=1}^{n} |a_{ij}| \right\}, \text{for } m > n \\ \max\limits_{m+1\leq i\leq n} \left\{ \sum\limits_{j=1}^{m} |a_{ji}| \right\}, \text{for } m < n. \end{cases}$$

In the case $m \geq n$, *each singular value of* A *lies in one of the real intervals*

$$B_i = [(a_i - s_i)_+, a_i + s_i], i = 1, \ldots, n,\ B_{n+1} = [0, s], \tag{4.13}$$

where $(a_i - s_i)_+ = \max(0, a_i - s_i)$. *If* $m = n$ *or if* $m > n$ *and* $a_i \geq s_i + s, i = 1, \ldots, n$, *then* B_{n+1} *is not needed in the above statement. Furthermore, every component interval of the union of* $B_i, 1, \ldots, n+1$ *(*n *for* $m = n$*), contains exactly* k *singular values if it contains* k *intervals of* $B_1, \ldots, B_n$.
In the case of $m \leq n$, n *is replaced by* m *in (4.13).*

Theorem 4.6 *Under the hypotheses of Lemma 4.3, the transfer function of the system (4.1) with arbitrary numbers of inputs and outputs satisfies the Lipschitz condition (4.4).*

Proof Let $H(p, s) = (h_{ij}(p, s)) \in \mathbb{C}^{l\times m}$. Clearly, inequalities (4.9) and (4.10) hold for all $h_{ij}(p, s), i = 1, \ldots, l, j = 1, \ldots, m$. We deduce that each entry satisfies the Lipschitz condition, with different Lipschitz constants. However, one can take a common bound for all of them, say

$$\sup_{s\in\mathbb{C}^+} |h_{ij}(p_1, s) - h_{ij}(p_2, s)| \leq K|p_1 - p_2|\ \forall p_1, p_2 \in [a, b]. \tag{4.14}$$

Now, we apply Theorem 4.5 to the matrix $H(p_1, s) - H(p_2, s) \in \mathbb{C}^{m\times l}$. Taking (4.14) into account, the quantities $a_i + s_i$ and s in Theorem 4.5 are bounded by $\min(l, m)K|p_1 - p_2|$. It is noteworthy that this bound holds for all values $p_1, p_2 \in [a, b]$ and $s \in \mathbb{C}^+$. Finally, we get the desired result

$$\forall p_1, p_2 \in [a, b], \|H(p_1, s) - H(p_2, s)\|_{\mathcal{H}_\infty} \leq \min(l, m)K|p_1 - p_2|. \qquad \blacksquare$$

If the original system depends smoothly on parameters, the reduced system is preferred to be so. The linear spline interpolation cannot give such property. Higher-order spline interpolation is therefore needed. Nevertheless, the high-order spline interpolation usually results in unnecessarily complicated procedure. Hence, cubic spline interpolation will be our choice. In the quite well-known paper [78] published in 1968, it was proven that the sequence of cubic spline interpolants associated with

a given fourth order continuously differentiable function $g(x)$ uniformly converges to $g(x)$ at a fourth order rate of the maximal distance between nodes. Therefore, one expects that a better error estimate than (4.5) will be derived if cubic spline interpolation is used. However, unlike linear spline interpolation where the interpolant is directly formulated as (4.2), one has to determine coefficients c_i^r (see (4.16) below) through solving a system of linear equations. This results in a difficulty in establishing the state space representation of the reduced system as well as getting a bound for the error. The next subsection will describe in detail how to interpolate the reduced transfer function by cubic splines, how much better the error bound is, how to reconstruct the state space representation as well as with which condition such an error bound can be achieved.

4.1.2 Using Cubic Spline Interpolation

In this subsection, we consider a SISO PDS

$$\begin{aligned} \dot{x}(t) &= A(p)x(t) + b(p)u(t), x(0) = 0, \\ y(t) &= c(p)x(t), \end{aligned} \tag{4.15}$$

where $A \in \mathbb{R}^{N\times N}, b, c$ are a column vector and a row vector in $\mathbb{R}^N$, respectively, depending on a single parameter $p \in [a, b]$. As in the preceding subsection, this system is assumed to be stable, reachable and observable for all $p \in [a, b]$. Its transfer function is denoted by $H(p, s)$.

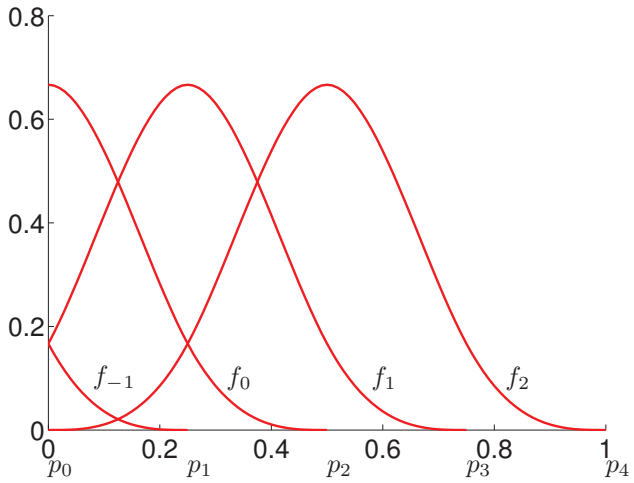

Figure 4.2: Cubic B-splines on a uniform grid

The interval $[a, b]$ is partitioned into k parts with a uniform grid $p_j = jh, j = 0, \ldots, k$, where $h = (b - a)/(k)$. In addition, we define two extra nodes, $p_{-1} = a - h, p_{k+1} = b + h$. The r_j-th order reduced transfer function of (4.15) created by balanced truncation at node p_j is again denoted by $\hat{H}_j(s)$.

The cubic basis splines (B-splines) with the given uniform grid are

$$f_i(p) = \frac{1}{6h^3}\begin{cases} (p-p_{i-2})^3, \text{ if } p \in [p_{i-2}, p_{i-1}), \\ -3(p-p_{i-1})^3 + 3h(p-p_{i-1})^2 + 3h^2(p-p_{i-1}) + h^3, \text{ if } p \in [p_{i-1}, p_i), \\ 3(p-p_{i+1})^3 + 3h(p-p_{i+1})^2 - 3h^2(p-p_{i+1}) + h^3, \text{ if } p \in [p_i, p_{i+1}), \\ -(p-p_{i+2})^3, \text{ if } p \in [p_{i+1}, p_{i+2}), \\ 0, \text{ otherwise,} \end{cases}$$

for $i = -1, \ldots, k+1$. Note that all of these functions are defined identical zero outsite the parameter interval $[a, b]$.

We construct the interpolant, which will be considered as the transfer function of the reduced system $\hat{\Sigma}(p)$, of the form

$$\hat{H}(p, s) = \sum_{i=-1}^{k+1} c_i^r(s) f_i(p), \tag{4.16}$$

where $c_i^r(s)$ are coefficients depending on s which need to be determined. Following Section 3.2, for interpolation conditions, we do not use the original transfer function but the pre-computed reduced $\hat{H}_j(s) = \hat{c}_j(sI_{r_j} - \hat{A}_j)^{-1}\hat{b}_j$ at node p_j

$$\hat{H}(p_j, s) = \hat{H}_j(s) \text{ or } \sum_{i=j-1}^{j+1} c_i^r(s) f_i(p_j) = \hat{H}_j(s), j = 0, \ldots, k. \tag{4.17}$$

In order to ensure the uniqueness of the cubic spline interpolation, end conditions must be added. There are three common choices [26, 79], and the choice of the end conditions, surprisingly, heavily influences the result. In our case we choose natural end conditions in order to avoid evaluating the derivatives of the original transfer function:

$$\frac{\partial^2 \hat{H}(p_0, s)}{\partial p^2} = \frac{\partial^2 \hat{H}(p_k, s)}{\partial p^2} = 0. \tag{4.18}$$

We will show later why this is also useful to get a state space representation. The interpolant is forced to satisfy (4.17) and (4.18) and this allows us to determine coefficients $c_i^r(s)$ through the linear system

$$FC^r(s) = H^r(s), \tag{4.19}$$

where

$$C^r(s) = [c_{-1}^r(s)\ c_0^r(s)\ \cdots\ c_{k+1}^r(s)]^T,$$
$$H^r(s) = [0, \hat{H}_0(s), \cdots, \hat{H}_k(s), 0]^T,$$

$$F = \frac{1}{6}\begin{bmatrix} \frac{6}{h^2} & -\frac{12}{h^2} & \frac{6}{h^2} & 0 & \cdots & 0 & 0 & 0 \\ 1 & 4 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & 4 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 & 4 & 1 \\ 0 & 0 & 0 & 0 & \cdots & \frac{6}{h^2} & -\frac{12}{h^2} & \frac{6}{h^2} \end{bmatrix} \in \mathbb{R}^{(k+3)\times(k+3)}.$$

Remark We do emphasize that the procedure proposed above is not the standard cubic spline interpolation technique that is well-known among the numerical analysis community. That is due to the fact that instead of using the values of the original transfer at interpolation points, we use their *reduced data*.

Since the right hand side of the equation (4.19) is composed of stable transfer functions of s, its solution C^r is also stable. This implies the stability of the reduced transfer function (4.16). □

The derivation of bound (4.5) for linear spline interpolation was quite straightforward, since the interpolant (4.2) was directly formulated from the pre-computed data and the linear basis spline. To derive a corresponding error bound for the cubic interpolation, a similar proof to that of Theorem 4.1 will be used. By studying the proof of Theorem 4.1, it is realized that an upper bound for the infinity norm - the maximum absolute row sum - of F^{-1}, must be constructed.

Forming bounds for the inverse of a matrix is quite an attractive but challenging task. The derived results can be used to estimate the error in approximation [4, 78], the lower bound for the smallest eigenvalue [171, 124] and therefore the bound for the matrix's condition number. In the general case, one may want to invoke a classical well-known statement taken from theory of functional analysis: "If a square matrix $A \in \mathbb{R}^{n\times n}$ satisfies $\|Ax\| \geq c\|x\|, \forall x \in \mathbb{R}^n$ for some fix constant c, then A is non-singular and $\|A^{-1}\| \leq c^{-1}$." Verifying the hypothesis of the above assertion for the infinity norm is, however, not tractable at all. Therefore, it is advisable to restrict the consideration to a smaller class of matrices. The authors of [123, 124, 171] get results for strictly diagonally dominant (SDD) and $\mathcal{S}$-strictly diagonally dominant ($\mathcal{S}$-SDD) [63] matrices. Regretfully, these results cannot be directly applied to our situation as F is neither an SDD nor an $\mathcal{S}$-SDD matrix. In [95], bounds for the norm of the inverses of a subclass of M-matrices [129] and H-matrices, so-called partitioned M-matrices and partitioned H-matrices (PM- and PH-matrices) respectively, were discussed. A direct application of these results is, however, not available, for F is neither a PM nor a PH-matrix. One idea is that, to deal with our matrix F, we utilize the scaling technique mentioned in [123] before applying the main result in [95]. For the elucidation of our later arguments, definitions of such matrices are recalled.

Definition 4.2 *Given a square matrix* $A = (a_{ij}) \in \mathbb{R}^{n\times n}$.

SDD matrix: *Matrix A is called an SDD matrix if $|a_{jj}| > \sum_{i\neq j} |a_{ij}|$ $\forall 1 \leq j \leq n$.*

$\mathcal{S}$-SDD matrix *[63, 64]: Let $\mathcal{S}$ be a non-empty subset of the index set $\{1, \cdots, n\}$. Denote by $\bar{\mathcal{S}}$ the complement of $\mathcal{S}$ in $\{1, \cdots, n\}$. Define $r_i^{\mathcal{S}}(A) := \sum_{j\in\mathcal{S}\setminus\{i\}} |a_{ij}|$. A is called an $\mathcal{S}$-SDD matrix if*

$$\begin{cases} |a_{ii}| > r_i^{\mathcal{S}}(A), & \forall i \in \mathcal{S}, \\ \Big(|a_{ii}| - r_i^{\mathcal{S}}(A)\Big)\Big(|a_{jj}| - r_j^{\bar{\mathcal{S}}}(A)\Big) > r_i^{\bar{\mathcal{S}}}(A) r_j^{\mathcal{S}}(A), & \forall i \in \mathcal{S}, j \in \bar{\mathcal{S}}. \end{cases}$$

M-matrix *[129]: A is called an M-matrix if A can be decomposed as*

$$A = sI - B,$$

in which $B = (b_{ij}), b_{ij} \geq 0$ $\forall i, j$ (matrices that have this property are called Z-matrices), s is a positive real number greater than the spectral radius of B.

H-matrix: *The comparison matrix of A, $\mathcal{M}(A) = (\tilde{a}_{ij})$ is defined as*

$$\tilde{a}_{ij} := \begin{cases} |a_{ij}|, i = j, \\ -|a_{ij}|, i \neq j. \end{cases}$$

Matrix A is called an H-matrix if $\mathcal{M}(A)$ is an M-matrix.

PM-matrix and PH-matrix *[95]: Let*

$$\bigcup_{i=1}^{k} M_i = \{1, \cdots, n\} \tag{4.20}$$

be a partitioning of the index set into k disjoint non-empty subsets which have n_i elements. We write

$$A_{ij} = A[M_i, M_j], i, j = 1, \cdots, k$$

for a representation of A in the block form, i.e.,

$$A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1k} \\ A_{21} & A_{22} & \cdots & A_{2k} \\ \cdots & \cdots & \cdots & \cdots \\ A_{k1} & A_{k2} & \cdots & A_{kk} \end{bmatrix}.$$

Now, we define $\prod_{i=1}^{k} n_i$ aggregated matrices by

$$A^{(i_1, i_2, \cdots, i_k)} := \begin{bmatrix} r_{i_1}(A_{11}) & r_{i_1}(A_{12}) & \cdots & r_{i_1}(A_{1k}) \\ r_{i_2}(A_{21}) & r_{i_2}(A_{22}) & \cdots & r_{i_2}(A_{2k}) \\ \cdots & \cdots & \cdots & \cdots \\ r_{i_k}(A_{k1}) & r_{i_k}(A_{k2}) & \cdots & r_{i_k}(A_{kk}) \end{bmatrix}, i_j \in M_j, j = 1, \cdots, k.$$

We call A a PM-matrix with respect to partitioning (4.20) if A is a Z-matrix and all aggregated matrices $A^{(i_1, i_2, \cdots, i_k)}$ are non-singular M-matrices.

Matrix A is called a PH-matrix if $\mathcal{M}(A)$ is an PM-matrix.

Remark Most of the aforementioned definitions and results also hold for complex matrices. For the sake of ease we restrict ourselves to real matrices.

Besides these definitions, there are numerous characterizations of such matrices. They can be found for instance in [20, 166]. □

The following theorem will be used in the proof of the lemma later.

Theorem 4.7 *([95], Theorem 3.1) If $A \in \mathbb{C}^{m\times m}, m \geq 1$ is a PH-matrix with respect to the partitioning (4.20), then A is an H-matrix, and its inverse satisfies the upper bound*

$$\|A^{-1}\|_\infty \leq \max_{i_1,\cdots,i_n} \|(\mathcal{M}(A)^{(i_1,\cdots,i_n)})^{-1}\|_\infty \tag{4.21}$$

Lemma 4.8 *Matrix F in (4.19) is non-singular and the following inequality holds*

$$\|F^{-1}\|_\infty \leq 6 \max\left\{\frac{64}{17}, \frac{288+48h^2}{77}\right\}, \tag{4.22}$$

where the infinity norm of an $n \times n$ constant matrix M is recalled as $\|M\|_\infty := \max_i \sum_{j=1}^n |m_{ij}|$.

Proof We would like to start the proof with a comment, that scaling a matrix M with a diagonal matrix whose diagonal entries have absolute values smaller than or equal to one will increase the infinity norm of the inverse. More precisely, if $0 < d_i \leq 1$ then $\|M^{-1}\|_\infty \leq \|(M\mathrm{diag}(d_i))^{-1}\|_\infty$.

First, the proof of the main theorem in [123] inspires us to scale matrix F to get better properties. Two quantities $B_1^{\mathcal{S}}, B_2^{\mathcal{S}}$ are defined as

$$0 \leq B_1^{\mathcal{S}} := \max_{i\in\mathcal{S}} \frac{r_i^{\bar{\mathcal{S}}}(A)}{|a_{ii}| - r_i^{\mathcal{S}}(A)} < B_2^{\mathcal{S}} := \min_{j\in\bar{\mathcal{S}}} \frac{|a_{jj}| - r_j^{\bar{\mathcal{S}}}(A)}{r_j^{\mathcal{S}}(A)} \leq 1. \tag{4.23}$$

It was shown in [63] that if $B_1^{\mathcal{S}} < \gamma < B_2^{\mathcal{S}}$, then F can be scaled by the diagonal matrix

$$\mathrm{diag}(\underbrace{\gamma,\cdots,\gamma}_{\mathcal{S}},\underbrace{1,\cdots,1}_{\bar{\mathcal{S}}})$$

to an SDD matrix. In our case, since only the first and the last rows of F contain h, we choose $\mathcal{S} = \{2, 3, \cdots, k+2\}$. The corresponding quantities are $B_1^{\mathcal{S}} = B_2^{\mathcal{S}} = 1/3$ which violates (4.23). Nevertheless, we still define $D_1 = \mathrm{diag}(1, 1/3, \cdots, 1/3, 1)$ and use this matrix for scaling F. Accordingly, we have

$$F_1 = FD_1 = \frac{1}{6}\begin{bmatrix} \frac{6}{h^2} & -\frac{4}{h^2} & \frac{2}{h^2} & 0 & \cdots & 0 & 0 & 0 \\ 1 & \frac{4}{3} & \frac{1}{3} & 0 & \cdots & 0 & 0 & 0 \\ 0 & \frac{1}{3} & \frac{4}{3} & \frac{1}{3} & \cdots & 0 & 0 & 0 \\ \cdots & \cdots & \cdots & \ddots & \ddots & \ddots & \cdots & \cdots \\ 0 & 0 & 0 & 0 & \cdots & \frac{1}{3} & \frac{4}{3} & 1 \\ 0 & 0 & 0 & 0 & \cdots & \frac{2}{h^2} & -\frac{4}{h^2} & \frac{6}{h^2} \end{bmatrix}.$$

It can be checked that the matrix F_1 is now an $\mathcal{S}$-SDD (not an SDD!) matrix with $\mathcal{S} = \{3, 4, \cdots, k+1\}$ as well as a PH-matrix with the partition $(1|2\cdots k+2|k+3)$. By Theorem 1.2 in [95], the non-singularity of F is, again, ensured.

A quite remarkable fact is, that applying the main theorems in [123] and Theorem 4.7 yields the same bound: $\|F_1^{-1}\| \leq 36$. Meanwhile, continuing to scale this matrix surprisingly gives a better bound. Since the two quantities $B_1^S = 1/3, B_2^S = 1$, one can multiply F_1 with matrix $D_2 = \text{diag}(1, 1, \gamma, \cdots, \gamma, 1, 1)$ where $\gamma \in (1/3, 1)$. The next step is to apply Theorem 4.7 to the PH-matrix $F_2 = F_1 D_2$. Obviously, the resulting bound for $\|F_2^{-1}\|_\infty$ depends on $\gamma \in (1/3, 1)$. The smallest upper bound among the bounds produced by applying Theorem 4.7 when γ varies in $(1/3, 1)$ is expected. However, the continuous min-max problem

$$\min_{\gamma\in(1/3,1)} \max_{i_1,i_2,i_3} \|(\mathcal{M}(F_2)^{(i_1,i_2,i_3)})^{-1}\|_\infty$$

can hardly be solved. Therefore, we will choose the best value for γ in the discrete sense by the following steps:

- First, discretize the interval $(1/3, 1)$ into a uniform mesh with step size $1/12$ and let γ take values $5/12, 6/12, \cdots, 11/12$ one by one. In each case, we apply Theorem 3.1 in [95] for $F_2 = F_1 D_2$ with the aforementioned partitioning.

- Then, pick two adjacent nodes whose corresponding bounds are the smallest. These two bounds are 24 and $6(3h^2/4 + 9/2)$ corresponding to $\gamma = 7/12$ and $\gamma = 8/12$. We can hope that the value of γ whose resulting bound is smallest lies in this subinterval.

- Repeat the above steps with the new interval $(7/12, 8/12)$ with step size $1/48$, we will get the interval $(28/48, 29/48)$. And do it again in this interval with step size $1/192$. At $\gamma = 115/192$, we get the required inequality.

That is, setting $F_2 = F_1 \text{diag}(1, 1, 115/192, \cdots, 115/192, 1, 1)$, applying Theorem 4.7 to F_2 results in the bound (4.22). ∎

Remark The bound (4.22) cannot, of course, be considered as the best bound. It is the smallest among only the mentioned discrete points.

In order to check the quality of the bound, we consider as examples two cases where F has the dimension 7 and 10. The infinity norms of the inverses of the comparison matrices are $6(31/10 + 13h^2/30)$ and $6(61/20 + 17h^2/40)$, respectively. It reveals that the bound (4.22) is quite close to $\|\mathcal{M}(F)^{-1}\|_\infty$.

If we use the Hermite end condition, the "collocation matrix" F reads,

$$F = \frac{1}{6}\begin{bmatrix} -\frac{3}{h} & 0 & \frac{3}{h} & 0 & \cdots & 0 & 0 & 0 \\ 1 & 4 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & 4 & 1 & \cdots & 0 & 0 & 0 \\ & & & & & & & \\ \cdots & \cdots & \cdots & \ddots & \ddots & \ddots & \cdots & \cdots \\ & & & & & & & \\ 0 & 0 & 0 & 0 & \cdots & 1 & 4 & 1 \\ 0 & 0 & 0 & 0 & \cdots & -\frac{3}{h} & 0 & \frac{3}{h} \end{bmatrix}.$$

By applying Theorem 3.1 of [95], one can show that

$$\|F^{-1}\| \leq 6\frac{h+1}{2}.$$

However, there is no feasible way to reconstruct the state space representation for the reduced system. □

Based on Lemma 4.8, we get the following error bound for the PMOR method using cubic spline interpolation.

Theorem 4.9 *Assume that the reduced system $\hat{\Sigma}(p)$ is constructed from the original system (4.15) by cubic spline interpolation as described above. Assume moreover that $\forall s \in \mathbb{C}^+$ the transfer function $H(p,s)$ is fourth order continuously differentiable with respect to p on $[a,b]$. Then*

$$\left\|\Sigma(p) - \hat{\Sigma}(p)\right\|_\infty \leq \frac{5}{384}\left\|\frac{\partial^4 H(p,s)}{\partial p^4}\right\|_\infty h^4 + 6\max\left\{\frac{64}{17}, \frac{288}{77} + \frac{48}{77}h^2\right\}\mathcal{E} \tag{4.24}$$

where $\mathcal{E} = \max\left\{\|H(p_i,s) - \hat{H}_i(s)\|_{\mathcal{H}_\infty}, i = 0,\ldots,k\right\}$.

Proof Let us denote by $C^f = [c^f_{-1}(s), c^f_0(s), \cdots, c^f_{k+1}(s)]^T$ the solution of the linear equation (4.19) but with the right hand side $H^f = [0, H(p_0,s), \cdots, H(p_k,s), 0]^T$, where $H(p_i,s)$ is the transfer function of the original full-order system at the parameter value p_i. Then the function $\sum_{i=-1}^{k+1} c^f_i f_i(p)$ is the cubic spline conventionally constructed from the original function $H(p,s)$. Next, we proceed analogously to the proof of Theorem 4.1

$$\begin{aligned}\left\|H(p,s) - \hat{H}(p,s)\right\|_\infty &= \left\|H(p,s) - \sum_{i=-1}^{k+1} c^r_i(s) f_i(p)\right\|_\infty \\ &\leq \left\|H(p,s) - \sum_{i=-1}^{k+1} c^f_i(s) f_i(p)\right\|_\infty + \left\|\sum_{i=-1}^{k+1} c^f_i(s) f_i(p) - \sum_{i=-1}^{k+1} c^r_i(s) f_i(p)\right\|_\infty. \end{aligned} \tag{4.25}$$

According to Theorem 1 in [78], the first term of (4.25) is dominated by $\frac{5}{384}\left\|\frac{\partial^4 H(p,s)}{\partial p^4}\right\|_\infty h^4$. It is noteworthy that $\frac{\partial^4 H(p,s)}{\partial p^4}$ is stable provided that $H(p,s)$

is so. This guarantees that $\left\| \frac{\partial^4 H(p,s)}{\partial p^4} \right\|_\infty$ is finite. Suppose that $p \in [p_j, p_{j+1}]$ for some $j = 0, \cdots, k$, the second term of (4.25) satisfies

$$\Big\| \sum_{i=-1}^{k+1} c_i^f(s) f_i(p) - \sum_{i=-1}^{k+1} c_i^r(s) f_i(p) \Big\|_\infty = \Big\| \sum_{i=j-1}^{j+2} \Big(c_i^f(s) - c_i^r(s) \Big) f_i(p) \Big\|_\infty$$

$$\leq \max \Big\{ \sup_{s \in \mathbb{C}^+} |c_i^f(s) - c_i^r(s)|, i = j-1, \cdots, j+2 \Big\} \sup_{p \in [p_j, p_{j+1}]} \sum_{i=j-1}^{j+2} f_i(p).$$

Note that the B-splines form a partition of unity, therefore the sum $\sum_{i=j-1}^{j+2} f_i(p)$ is identically equal to 1. By hypothesis, $F(C^f - C^r) = H^f - H^r$ which implies that $C^f - C^r = F^{-1}(H^f - H^r)$ and

$$\|C^f - C^r\|_\infty \leq \|F^{-1}\| \|H^f - H^r\|_\infty \tag{4.26}$$

where $\|.\|_\infty$ of a vector of functions is understood as the maximum of supremums of its elements. Thus $\|H^f - H^r\|_\infty$ is nothing else but the maximal value of errors $\mathcal{E}$ caused by balanced truncation of the original system at grid points $p_0, \ldots, p_k$. The theorem is directly deduced from Lemma 4.8 and (4.25) and (4.26). ∎

Remark A state space representation of the reduced system can be constructed as follows. Denote by $f(p) = [f_{-1}(p)\ f_0(p)\ \cdots\ f_{k+1}(p)]^T$, the column of cubic basis splines and $\mathcal{F}(p)^T = [\mathcal{F}_i(p)]_{i=-1,\cdots,k+1}^T = f(p)^T F^{-1}$. We get

$$\begin{aligned}
\hat{H}(p,s) &= \sum_{i=-1}^{k+1} f_i(p) c_i^r(s) = f(p)^T C^r = f(p)^T F^{-1} H^r \\
&= \mathcal{F}(p)^T H^r = \sum_{i=0}^{k} \mathcal{F}_i(p) \hat{c}_i (s I_{r_i} - \hat{A}_i)^{-1} \hat{b}_i \\
&= \begin{bmatrix} \mathcal{F}_0(p)\hat{c}_0 & \mathcal{F}_1(p)\hat{c}_1 & \cdots & \mathcal{F}_k(p)\hat{c}_k \end{bmatrix} \begin{bmatrix} sI_{r_0} - \hat{A}_0 & & & \\ & \ddots & & \\ & & & sI_{r_k} - \hat{A}_k \end{bmatrix}^{-1} \begin{bmatrix} \hat{b}_0 \\ \vdots \\ \hat{b}_k \end{bmatrix}.
\end{aligned}$$

Then the reduced system is $\hat{\Sigma}(p) = \left(\begin{array}{c|c} \hat{A} & \hat{b} \\ \hline \hat{c}(p) & 0 \end{array} \right)$, where $\hat{A} = \mathrm{diag}(\hat{A}_0, \cdots, \hat{A}_k), \hat{b} = [\hat{b}_0\ \cdots\ \hat{b}_k]^T$ and $\hat{c}(p) = [\mathcal{F}_0(p)\hat{c}_0\ \cdots\ \mathcal{F}_k(p)\hat{c}_k]$, of the order $\sum_{i=0}^k r_i$.

As in the previous subsection, it is evident that if all entries of A, b, c of the stable system (4.15) are fourth order continuously differentiable, so is its transfer function.

In the case of MIMO systems, by using Theorem 1 in [78], the bound for the first term of (4.25) only holds elementwise. The deduction from the boundedness of the entries of a matrix to the boundedness of its infinity norm is likely possible

by using Theorem 4.5. However, the bound for the second term of (4.25) is still not known. Therefore, the extension of Theorem 4.9 to the MIMO case still remains unsolved. Regardless of this fact, the procedure using cubic spline interpolation presented above still works for MIMO systems. □

4.1.3 Numerical Example

To show the effectiveness of our method, we consider in this subsection a model derived from the discretization of a convection-diffusion equation

$$\frac{\partial \Phi}{\partial t} = \triangle \Phi + p \cdot \nabla \Phi + S(\zeta, \xi) u(t) \tag{4.27}$$

on $Q_T = \Omega \times (0, T), \Omega = (0, 1)^2$ with homogeneous initial and boundary conditions

$$\begin{aligned} &\Phi(\zeta, \xi, 0) = 0 \text{ on } \Omega, \\ &\Phi|_{\partial\Omega \times (0,T)} = 0. \end{aligned} \tag{4.28}$$

In (4.27) and (4.28), Φ represents the concentration of some material in a fluid medium, $p = (p_1, p_2)$ is the velocity field and $S(\zeta, \xi)u(t)$ stands for the source of a contaminant. These equations are usually used as a mathematical model for a pollution process.

Equation (4.27) is semi-discretized by the finite difference method over the domain Ω such that it results in a dynamical system of size 576,

$$\begin{aligned} \dot{x}(t) &= (A + p_1 A_1 + p_2 A_2) x(t) + b u(t), \\ y(t) &= c x(t), \end{aligned}$$

where $x(t)$ denotes the spatially discretized state $\Phi(\zeta, \xi, t)$. The partial derivatives are approximated as follows

$$\begin{aligned} \triangle \Phi_{i,j} &= \frac{1}{h^2}(\Phi_{i-1,j} + \Phi_{i+1,j} + \Phi_{i,j-1} + \Phi_{i,j+1} - 4\Phi_{i,j}), \\ \Big(\frac{\partial \Phi}{\partial x}\Big)_{i,j} &= \frac{1}{h}(\Phi_{i+1,j} - \Phi_{i,j}), \\ \Big(\frac{\partial \Phi}{\partial y}\Big)_{i,j} &= \frac{1}{h}(\Phi_{i,j+1} - \Phi_{i,j}). \end{aligned}$$

Matrices A, A_1, A_2 result from discretizations of the diffusion and convection terms, respectively. The function S is specified as

$$S(\zeta, \xi) = \frac{1}{\exp\Big(100\big((\zeta - \frac{1}{2})^2 + (\xi - \frac{1}{2})^2\big)\Big)},$$

which stands for the location of the source at $(1/2, 1/2)$ and yields the input vector b.

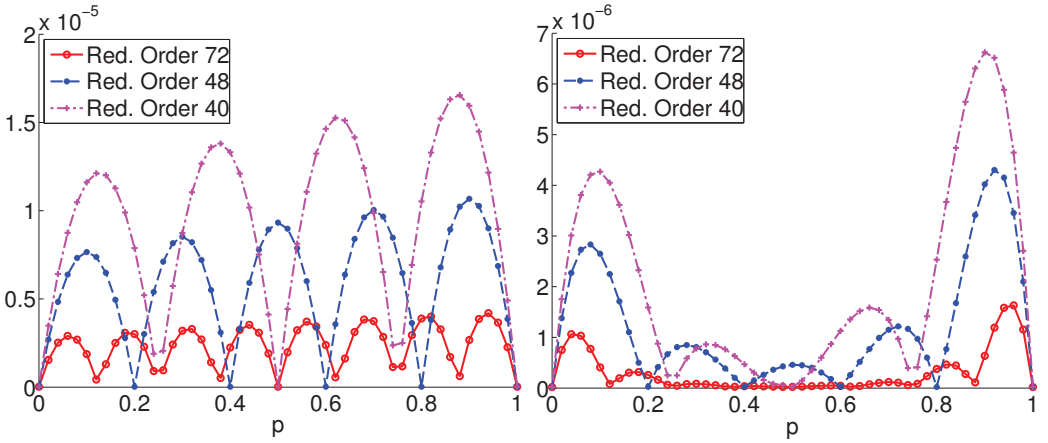

Figure 4.3: Absolute error of linear spline (left) and cubic spline (right) interpolation based method

The output vector c represents a linear function of the states whose expression is

$$l = x_1 + 3x_2 - 3x_3 + 90x_4 - 4x_{50} + 5x_{100} + 3x_{150} + 27x_{200} + 11x_{250} \\ -34x_{300} + 12x_{350} + 6x_{400} - 5x_{450} - 4x_{496} + x_{490} + 3x_{520} + x_{570}.$$

The velocity field is treated as parameters. As we would like to study the single parameter case, p_1 is assumed to change within $[0, 1]$ while p_2 is a fixed constant and quite small in comparison with p_1. For the sake of simplicity, we use a uniform grid for the parameter space and equally reduced orders at grid points. In our example, the reduced order at each grid point is always 8. Hence, the resulting

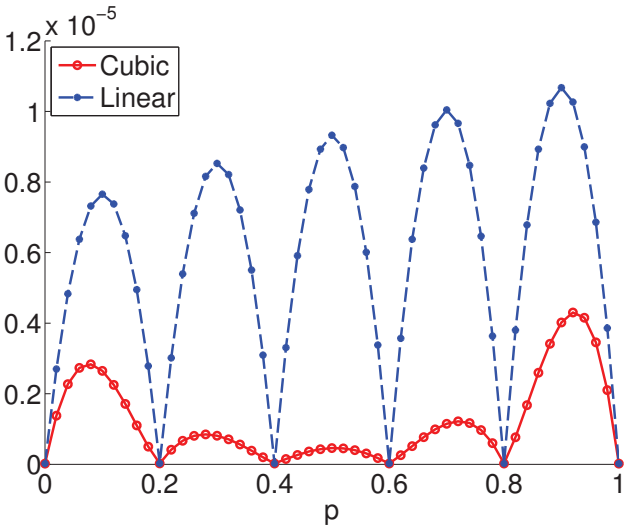

Figure 4.4: A comparison of the errors of two methods; reduced order 48

reduced order will be a multiple of 8. For example, if the reduced order is 48, 6 grid points $0, 0.2, 0.4, 0.6, 0.8, 1$ have been chosen. Both linear splines and cubic splines are used to interpolate the reduced transfer function and then construct the state space representation of the overall reduced system. Figure 4.3 depicts the derived absolute errors caused by the reduction. Figure 4.4 compares the error caused by cubic spline interpolation to that caused by linear spline interpolation. These errors are computed with infinity norm by the Matlab command `norm(sys,inf)` at all points of the grid $0 : 0.02 : 1$.

As we can see in the two figures, the error decreases when more interpolation points are added. And as expected, the cubic spline interpolation gives a smaller error than that given by linear spline interpolation. Another thing may be concluded from the shape of the error plots is that the error caused by interpolation is rather large. It can also be observed that the error caused by cubic spline interpolation tends to be larger at the two ends of the parameter interval. This may be due to the effect of the end conditions.

4.1.4 Discussion

The first term of the derived error bound using spline interpolation is, in general, difficult to be computed. If, however, by some mean, one can compute it or have an estimation of it, the derived error bound can be used as a hint to get a reduced systems that satisfies a given error tolerance as follows. First, the step size h should be chosen such that the first term of the error bound is less than a half of the tolerance. Then, the local reduced orders at grid points are decided such that the maximum of the errors, $\mathcal{E}$, is small. Clearly, the smaller the (local) reduced order is, the larger the (local) error is. Meanwhile, the reduced order is always expected as small as possible. Therefore, $\mathcal{E}$ is chosen such that the reduced order is "large" enough for the second term of the error bound is, again, less than a half of the tolerance.

So far, our method is presented to work with ordinary systems. For algebraic dynamical system, we can use the balanced truncation method specially designed for descriptor systems. For more details, the reader is referred to [169].

A challenging issue is the choice of interpolation points, including both the number of points and their locations. Too many grid points will enlarge the reduced order, produce more unnecessary computations, and therefore reduce the efficiency of the method. Nevertheless, too few points cannot capture the variation of the system. In [17], it is proposed to use a sparse grid [30] to reduce the number of grid points.

Since the variation of the system may be quite different from part to part in the parameter domain, the locations of grid points, as many other interpolation methods, needs to be optimized. In our problem, both aspects, number of interpolation points and their locations, can be addressed based on the study of the effect of perturbation on the reduced order models of dynamical systems, which is the main purpose of two research papers [87, 160]. In these papers, the difference between

the reduced order state, produced by POD, and the original state was estimated via the approximation error and the magnitude of the parameter perturbation. Based on this, the so-called *regions of validity* is determined, which helps to choose the size of a grid point's neighborhood on which the error caused by reduction is still acceptable. We may address this subject in a future project.

By analyzing the data from the above numerical example and the derived error bound (4.24), it appears that the norm of the fourth derivative of the transfer function dominates the bound. Usually, systems which change a lot when the parameter changes, have very large derivatives. We henceforth call them *highly varying systems*. This raises doubts about the effectiveness of the proposed method when applied to such systems. We consider the following theoretical example.

$$A = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & -3 & 0 & 0 & 0 \\ 0 & 0 & 0 & -4 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1.1 & -p/2 - 0.55 \\ 0 & 0 & 0 & 0 & p/2 + 0.5505 & p \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ -1 \\ 2 \end{bmatrix} \tag{4.29}$$

$$c = \begin{bmatrix} 3 & 2 & 1 & 4 & 2 & -3 \end{bmatrix}, \qquad d = 0.$$

When parameter p is in the closed interval $[0, 1]$, system (4.29) is stable, reachable and observable. We apply the proposed method using cubic splines with 5 interpolation points. The system is reduced to order 1 at the first 4 points and 2 at the fifth point. This results in an overall reduced system of the order 6. Figure 4.5 shows the absolute error and the norm of original system.

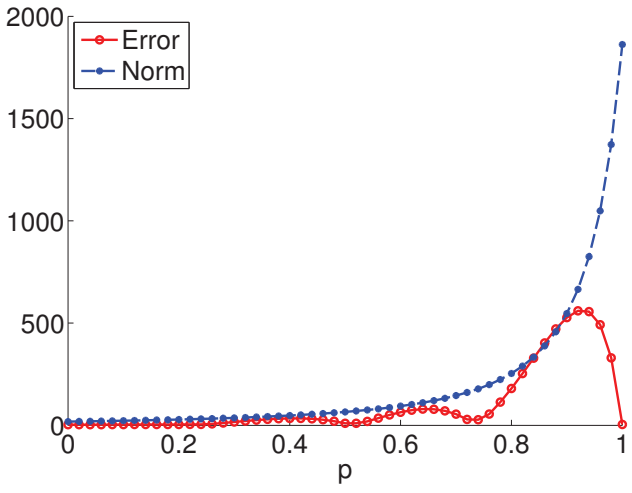

Figure 4.5: Absolute errors of cubic spline method vs. norm of the original system

One can observe that in the domain $(0.81, 0.94)$, the errors are extraordinarily large and even exceed the norm of the original system. One can easily check the

Hankel singular values of (4.29) in that domain and therefore be sure that it is not caused by the truncation of states. What makes the method deteriorate? It turns out that the bound for the absolute error, as shown in (4.24), can also depend on the fourth order derivative of the transfer function. The values of $\|\frac{\partial^4 H(p,s)}{\partial p^4}\|_\infty$ at nodes $0.02i,\ i = 1, \cdots, 50$ are rowwise given in Table 4.1.

Table 4.1: Infinity-norm of the fourth-order derivative of the transfer function of system (4.29)

1.7416e+03	1.9454e+03	2.1776e+03	2.4429e+03	2.7467e+03
3.0956e+03	3.4975e+03	3.9616e+03	4.4993e+03	5.1243e+03
5.8531e+03	6.7059e+03	7.7073e+03	8.8879e+03	1.0285e+04
1.1946e+04	1.3928e+04	1.6305e+04	1.9169e+04	2.2638e+04
2.6862e+04	3.2035e+04	3.8407e+04	4.6308e+04	5.6171e+04
6.8570e+04	8.4279e+04	1.0435e+05	1.3020e+05	1.6383e+05
2.0802e+05	2.6670e+05	3.4556e+05	4.5287e+05	6.0091e+05
8.0818e+05	1.1030e+06	1.5295e+06	2.1576e+06	3.0984e+06
4.5266e+06	6.6973e+06	1.0059e+07	1.6912e+07	3.8180e+07
2.5958e+08*	2.8061e+07	2.7601e+07	8.6382e+07	7.3125e+07

One can see that the norm of the fourth derivative of the transfer function in the mentioned domain (*) tends to be the largest. This explains, taking the error bound (4.24) into account, why the absolute errors in this area are large.

From these facts, we would like to emphasize that using interpolation based (spline interpolation and perhaps all kinds of polynomials interpolation based) methods for PMOR should be conducted with care. These methods may deteriorate when being applied to highly varying parameter-dependent systems.

The spline interpolation based method proposed in this section is, mathematically, to approximate a parameter-dependent transfer function over a parameter domain. However, adding two transfer functions may lead to a transfer function with no meaningful physical interpretation. For example, we consider two transfer functions [1]

$$H_1(s) = \frac{\omega_1}{s^2 + \omega_1^2}, \quad H_2(s) = \frac{\omega_2}{s^2 + \omega_2^2},$$

whose impulse responses are $h_1(t) = \sin \omega_1 t$ and $h_2(t) = \sin \omega_2 t$, respectively. They can be considered as simple models of two tuning forks with different lengths, which produce two different tones when excited. The function

$$H_3(s) = \frac{1}{2}\Big(H_1(s) + H_2(s)\Big)$$

is the average of the two given transfer functions whose impulse response is $h_3(t) = 1/2(\sin \omega_1 t + \sin \omega_2 t)$. It certainly gives out an accord, instead of a tone, which is far different from the original sound. This is also illustrated through their sigma plots,

[1] This example is due to a private discussion with Boris Lohmann

see Figure 4.6. As we can observe, the interpolation adds an extra pole to $H_3(s)$. From this fact, one would like an approach that can work directly with system

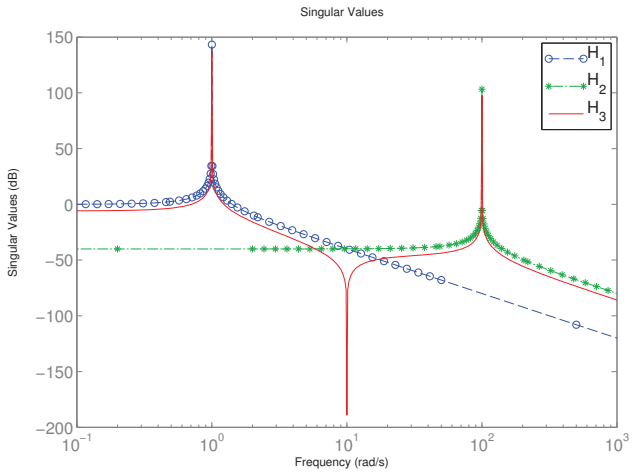

Figure 4.6: Sigma plot of transfer functions $H_1(s)$ and $H_2(s)$ and its interpolant

matrices but still can avoid the problem mentioned in Section 3.3. Interpolation of projection subspaces might be a good choice. In the next section, the details of this method will be presented.

4.1.5 Final Remarks

The terminology "interpolation of a transfer function" has so far been used in several different contexts, we would like to discriminate them in order to avoid confusion. It was first used to name the multi-point moment matching method in Section 2.2.3, in which the reduced transfer function is constructed in such a way that it matches the value and several consecutive derivatives of the original transfer function at some frequencies. This is the normal interpolation of one-variable functions whose values are in $\mathbb{C}$ or $\mathbb{C}^{l\times m}$ (in the case of a MIMO system). In Chapter 3, it is recalled during the Krylov subspace based methods; this is merely the interpolation of functions of several variables. Also in Section 3.2 of Chapter 3 and this section, this term is used to call the two approaches of interpolation of parameter-dependent transfer functions. They may be considered as the interpolation of functions whose values lie in the Banach spaces $\mathcal{H}_\infty$.

The approach of this work shares the same idea with [16, 17], in which the authors used Lagrange, Hermite, rational and sinc interpolations. We would like to emphasize that no error bound estimate was given in [16]. There was one error bound given in [17] for Lagrange interpolation.

4.2 Interpolation of Projection Subspace

Model order reduction is, thoroughly speaking, to find an appropriate subspace for projection to gain some purpose, e.g., matching specific moments, removing marginal states. The cooperation between projection and interpolation is the backbone of the interpolation based methods. In the approaches presented so far, the data for interpolation are the reduced models given a set of parameter samples; the projections on subspaces corresponding to a chosen set of parameter values are performed before the interpolation. We may also try to reverse this order, i.e., we interpolate the projection subspace first, to get a parameter-dependent subspace, and then straightforwardly project the original system on to the derived subspace. This method, on the one hand, results in no extra poles, which happens when interpolating the transfer function, and on the other hand, avoids the problem explained in Section 3.3 when reduced system matrices are interpolated. It is because the interpolation is performed in the set of r-dimensional subspaces in $\mathbb{R}^N$, which constitute a Grassmann manifold. As a consequence, we are led to the problem of interpolation of data on a manifold, whose structure has to be taken into account. As mentioned in Section 3.4, interpolation on Grassmann manifold and its application to constructing ROMs based on a set of precomputed ROMs was first proposed in [7]. An interpolation algorithm was also proposed. It will be explained in the first part of this section. In the second part, we will give a strategy to reduce the computational complexity of the proposed algorithm.

4.2.1 Application to MOR for PDSs

Consider a PDS

$$\begin{aligned} E(p)\dot{x}(t) &= A(p)x(t) + B(p)u(t), \\ y(t) &= C(p)x(t), \end{aligned} \tag{4.30}$$

in which $E(p), A(p) \in \mathbb{R}^{N\times N}, B(p) \in \mathbb{R}^{N\times m}, C(p) \in \mathbb{R}^{l\times N}$ depend on parameters $p \in \Omega \subset \mathbb{R}^d$, where Ω is closed and bounded. We will use the Krylov subspace method with one-sided projection to reduce this system. To this end, we have to build a matrix W satisfying (2.36). However, the projection matrix W in this case is no longer a constant matrix, since the system matrices depend on p. For each $p \in \Omega$, we denote the mentioned projection matrix by $W(p)$ to emphasize its dependence on p. The direct computation of $W(p)$ for all $p \in \Omega$, i.e., an explicit formula of $W(p)$, is impossible. Thus, interpolation is invoked as a tool to construct an approximation of $W(p)$. Let us state the problem of interpolation as follow.

Given $p_0, \cdots, p_k \in \Omega$, denote by $W_0, \cdots, W_k$ columnwise orthogonal matrices whose columns span projection subspaces $S_0, \cdots, S_k$, respectively. Construct a parameter-dependent basis $W(p)$ for $S(p)$ by interpolating the data $(p_0, W_0), \cdots, (p_k, W_k)$.

The simplest idea is to form $W(p)$ as a weighted sum of $W_0, \cdots, W_k$

$$W(p) := \sum_{i=1}^{k} \omega_i(p) W_i,$$

where $\omega_i(p)$ are some weight functions. This solution may lead to the situation in which the resulting matrix $W(p)$ is no longer a basis for a subspace. In other words, direct interpolation on a Grassmann manifold may result in a point not being included in it. This can be illustrated in the following example. Consider $S_1, S_2 \in \mathcal{G}(2,3)$ represented by

$$W_1 = \begin{bmatrix} 0 & 0 \\ 9 & 3 \\ 18 & 0 \end{bmatrix} \text{ and } \mathrm{W}_2 = \begin{bmatrix} 2 & \frac{1}{3} \\ 1 & 0 \\ 0 & \frac{1}{3} \end{bmatrix}.$$

Then $S = (1/10)S_1 + (9/10)S_2$ is represented by

$$\begin{aligned} W &= \frac{1}{10} W_1 + \frac{9}{10} W_2 \\ &= \begin{bmatrix} 0 & 0 \\ \frac{9}{10} & \frac{3}{10} \\ \frac{18}{10} & 0 \end{bmatrix} + \begin{bmatrix} \frac{18}{10} & \frac{3}{10} \\ \frac{9}{10} & 0 \\ 0 & \frac{3}{10} \end{bmatrix} \\ &= \begin{bmatrix} \frac{9}{5} & \frac{3}{10} \\ \frac{9}{5} & \frac{3}{10} \\ \frac{9}{5} & \frac{3}{10} \end{bmatrix}, \end{aligned}$$

and therefore $S \notin \mathcal{G}(2,3)$. The reason is that Grassmann manifolds are not a space; they are not flat. Hence, interpolation must be modified or performed in approximation sense.

One cannot directly interpolate a function whose values lie on a Grassmann manifold, but one can do that on its tangent space. A reliable connection between the given data and their corresponding data on the tangent space must be established, because the data is initially given on the manifold, they have to be mapped to the tangent space, where the interpolation is performed, and then the interpolated data is mapped back to the manifold. This idea was first proposed in [7], and was then followed by [6, 45, 8] which have been mentioned in Section 3.4. For the clarity of our discussion, we restate the procedure here, which is for Grassmann manifolds only.

Given $S_0, S_1, \cdots, S_k \in \mathcal{G}(r,N)$ represented by columnwise orthogonal matrices $W_0, W_1, \cdots, W_k$.

Step 1 Choose the contact point for the tangent space, e.g., S_0.

Step 2 Map point $S_1, \cdots, S_k$ to $\mathcal{T}_{S_0}\mathcal{G}(r,N)$ by Log_{S_0}. By (2.44), $Log_{S_0}(S_i) = Y_i$ is a vector represented by

$$Z_i = U_i \arctan(\Lambda_i) V_i^T, \tag{4.31}$$

where

$$(I - W_0 W_0^T) W_i (W_0^T W_i)^{-1} = U_i \Lambda_i V_i^T, i = 1, \cdots, k$$

is the thin SVD.

Step 3 Interpolate on $\mathcal{T}_{S_0}\mathcal{G}(r, N)$ using some standard interpolation technique. Note that $Log_{S_0}(S_0) = Y_0 = 0$. Given a parameter value $p \in \Omega$, we denote by $Y(p)$ the vector on $\mathcal{T}_{S_0}(r, N)$ corresponding to the parameter value p, which will be computed by interpolation. By any common interpolation technique, $Y(p)$ is represented by the matrix

$$Z(p) = \sum_{i=1}^{k} f_i(p) Z_i. \tag{4.32}$$

Step 4 Map the interpolated result $Y(p)$ back to the Grassmann manifold. Using the exponential mapping, (2.43), one has to compute first the thin SVD

$$Z(p) = U(p)\Lambda(p)V(p)^T, \tag{4.33}$$

and then the matrix representation of the subspace is

$$W(p) = W_0 V(p) \cos(\Lambda(p)) + U(p) \sin(\Lambda(p)). \tag{4.34}$$

Finally, the system matrices of the sought-after reduced system are constructed as

$$\begin{aligned} \hat{E}(p) &= W^T(p) E(p) W(p), \\ \hat{A}(p) &= W^T(p) A(p) W(p), \\ \hat{B}(p) &= W^T(p) B(p), \\ \hat{C}(p) &= C(p) W(p). \end{aligned} \tag{4.35}$$

Remark This method can be combined with all MOR methods that can be formulated as a one-sided projection such as POD and one-sided Krylov subspace method.

One condition that has to be satisfied when using this method is, that $S_1, \cdots, S_k$ are in a neighborhood of S_0. This is because the connection between the Grassmann manifold and its tangent space is based on geodesic paths, which are determined by a second order differential equation [2]. The closeness of S_i to S_0 is a requirement for the existence of the solution of the underlying equation. If the distance, which is defined in [2], between S_i and S_0 is rather large, one should partition the parameter domain into some subdomains and choose one contact point for each subdomain. □

4.2.2 Reduction of Computational Complexity

Formula (4.35), mathematically, provides a nice expression for constructing the reduced system. In practice especially in online simulation, the computational costs for each new parameter value p are essential: Given $p \in \Omega$, one needs to evaluate the reduced system at p as fast as possible. Applying the aforementioned four-step procedure, one has to compute

- interpolation on the tangent spaces at Step 3 requiring $\mathcal{O}(Nr)$ operations;
- thin SVD (4.33) requiring $\mathcal{O}(Nr^2)$ operations;
- matrix multiplication (4.34) requiring $\mathcal{O}(Nr^2)$ operations;
- reduced system matrices (4.35) requiring $\mathcal{O}(N^2r)$ operations.

In the model reduction framework, N is typically very large. With computational complexity of $\mathcal{O}(N^2r)$, the online computation is in general rather slow. This is the reason why the result in [7] failed to be usable in real time. To enable this, the only way is to exclude the dependence of the computational complexity on N. The presentation of our solution will start with a simple case.

4.2.2.1 Linear Interpolation and Single Parameter

Let $\Omega = [a, b]$ and $a = p_0 < \cdots < p_k = b$. Since the procedure can be applied to each subinterval $[p_{i-1}, p_i]$, we can restrict ourselves, without loss of generality, to the case $k = 1$. The process of interpolating on Grassmann manifolds for the case of linear interpolation of a single parameter is illustrated in Figure 4.7.

At Step 3, using linear interpolation with two vectors Y_1 and $Y_0(\equiv 0)$ leads to

$$Y(p) = \frac{p - p_0}{p_1 - p_0} Y_1.$$

Therefore

$$Z(p) = U_1 \frac{p - p_0}{p_1 - p_0} \arctan(\Lambda_1) V_1^T. \tag{4.36}$$

Note that (4.36) is still the thin SVD of $Z(p)$, we do not have to compute the SVD (4.33) in Step 4; the basis for the projection subspace at p is straightforwardly written down as

$$W(p) = W_0 V_1 \cos\Big(\frac{p - p_0}{p_1 - p_0} \arctan(\Lambda_1)\Big) + U_1 \sin\Big(\frac{p - p_0}{p_1 - p_0} \arctan(\Lambda_1)\Big). \tag{4.37}$$

Inspired by the reduced basis method [132] which was initially proposed to deal with parameterized elliptic equations, we now assume that the system matrices of (4.30)

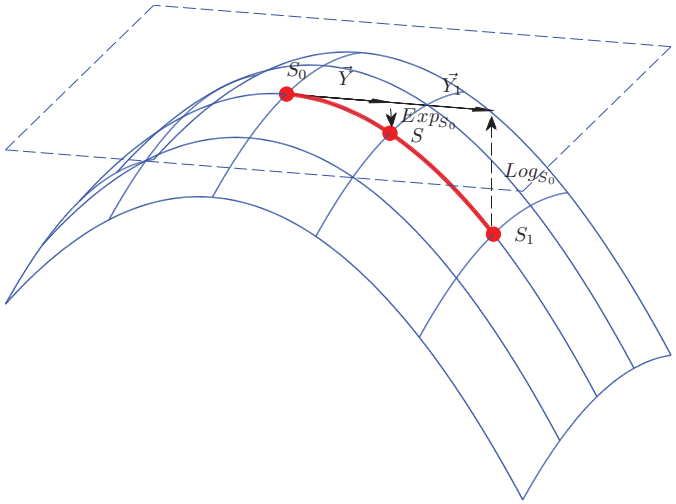

Figure 4.7: Interpretation of the interpolation on Grassmann manifolds

are affinely dependent on p, i.e.,

$$\begin{aligned} E(p) &= \sum_{i=1}^{\Phi_E} f_i^E(p) E_i, \\ A(p) &= \sum_{i=1}^{\Phi_A} f_i^A(p) A_i, \\ B(p) &= \sum_{i=1}^{\Phi_B} f_i^B(p) B_i, \\ C(p) &= \sum_{i=1}^{\Phi_C} f_i^C(p) C_i, \end{aligned} \tag{4.38}$$

where E_i, A_i, B_i, C_i are independent of p. For the effectiveness of the method presented later, we assume moreover, that Φ_E, Φ_A, Φ_B, Φ_C are very small compared to N, and the evaluations of $f_i^E, f_i^A, f_i^B, f_i^C$ for all p are cheap. Indeed many mathematical models satisfy these conditions, e.g., the Helmholtz problem [118], heat conduction problems [159], and thermal flow [149]. Moreover, one can always linearize the nonlinear dependence or interpolate the implicit dependence on parameters to derive the affine structure (4.38).

For the sake of brevity, we denote by $\Xi(p)$ the diagonal matrix $\Big((p-p_0)/(p_1-$

$p_0)\Big)\arctan(\Lambda_1)$. Accordingly, the reduced matrices (4.35) are written as

$$\begin{aligned}
\hat{E}(p) &= W^T(p)E(p)W(p) = \sum_{i=1}^{\Phi_E} f_i^E(p)W^T(p)E_iW(p) \\
&= \sum_{i=1}^{\Phi_E} f_i^E(p)\Big(\cos(\Xi(p))V_1^TW_0^T + \sin(\Xi(p))U_1^T\Big)E_i\Big(W_0V_1\cos(\Xi(p)) + U_1\sin(\Xi(p))\Big) \\
&= \sum_{i=1}^{\Phi_E} f_i^E(p)\cos(\Xi(p))\mathbf{V_1^TW_0^TE_iW_0V_1}\cos(\Xi(p)) \\
&\quad + \sum_{i=1}^{\Phi_E} f_i^E(p)\cos(\Xi(p))\mathbf{V_1^TW_0^TE_iU_1}\sin(\Xi(p)) \\
&\quad + \sum_{i=1}^{\Phi_E} f_i^E(p)\sin(\Xi(p))\mathbf{U_1^TE_iW_0V_1}\cos(\Xi(p)) \\
&\quad + \sum_{i=1}^{\Phi_E} f_i^E(p)\sin(\Xi(p))\mathbf{U_1^TE_iU_1}\sin(\Xi(p)).
\end{aligned} \tag{4.39}$$

The matrix $\hat{A}(p)$ is computed analogously:

$$\begin{aligned}
\hat{A}(p) &= \sum_{i=1}^{\Phi_A} f_i^A(p)\cos(\Xi(p))\mathbf{V_1^TW_0^TA_iW_0V_1}\cos(\Xi(p)) \\
&\quad + \sum_{i=1}^{\Phi_A} f_i^A(p)\cos(\Xi(p))\mathbf{V_1^TW_0^TA_iU_1}\sin(\Xi(p)) \\
&\quad + \sum_{i=1}^{\Phi_A} f_i^A(p)\sin(\Xi(p))\mathbf{U_1^TA_iW_0V_1}\cos(\Xi(p)) \\
&\quad + \sum_{i=1}^{\Phi_A} f_i^A(p)\sin(\Xi(p))\mathbf{U_1^TA_iU_1}\sin(\Xi(p)).
\end{aligned} \tag{4.40}$$

Likewise,

$$\begin{aligned}
\hat{B}(p) &= W^T(p)B(p) = \sum_{i=1}^{\Phi_B} f_i^B(p)W^T(p)B_i \\
&= \sum_{i=1}^{\Phi_B} f_i^B(p)\cos(\Xi(p))\mathbf{V_1^TW_0^TB_i} + \sum_{i=1}^{\Phi_B} f_i^B(p)\sin(\Xi(p))\mathbf{U_1^TB_i}.
\end{aligned} \tag{4.41}$$

$$\begin{aligned}
\hat{C}(p) &= C(p)W(p) = \sum_{i=1}^{\Phi_C} f_i^C(p)C_iW(p) \\
&= \sum_{i=1}^{\Phi_C} f_i^C(p)\mathbf{C_iW_0V_1}\cos(\Xi(p)) + \sum_{i=1}^{\Phi_C} f_i^C(p)\mathbf{C_iU_1}\sin(\Xi(p)).
\end{aligned} \tag{4.42}$$

All matrices in (4.39)-(4.42) emphasized with bold letters are independent of p; they can be computed and stored before starting the online stage. In the following, we summarize the whole PMOR process in terms of offline-online decomposition.

Offline We compute and store

- Two columnwise orthogonal projection matrices W_0, W_1 at p_0 and p_1 by the Krylov subspace method.
- The thin SVD
$$(I - W_0 W_0^T) W_1 (W_0^T W_1)^{-1} = U \Lambda V^T.$$
- The parameter-independent terms which are emphasized with bold letters in (4.39)–(4.42): $\mathbf{V^T W_0^T E_i W_0 V}, \mathbf{V^T W_0^T E_i U}, \cdots, \mathbf{C_i U_1}$.

The most expensive computations here are the SVD and matrix multiplications in the second and the third step of the offline stage, which need $\mathcal{O}(N^2 r)$ operations.

Online Given a parameter value p, compute the reduced system matrices via (4.39)-(4.42).

The computational cost of the online stage is $\mathcal{O}(r^2)$, totally independent of N. This will accelerate the computation and therefore enable the method to be used in real time.

Remark The case of linear interpolation and single parameter was also analyzed in [5]. The formula (4.37) was derived in this work as well. However, it was mentioned only in order to show the relation between interpolation on Grassmann manifolds and interpolation of subspace angles derived in [111]. The improvement of computational speed was not considered there. □

4.2.2.2 General Case

The key point of the solution to linear interpolation of a single parameter is the formula (4.36). Thanks to the simplicity of the linear interpolation, we have an explicit expression for $W(p)$ without the computation of SVD (4.33) as in the general case. To extend the result to the general case, we have to deal with the SVD of the sum (4.32). More precisely, a suitable strategy to compute the SVD of this sum has to be set up and in the development which will be seen later, a careful combination of the SVD with the offline-online decomposition must be performed.

One can observe that no matter what the dimension of the parameter domain is and/or no matter how high the order of interpolation is, one derives the interpolant of the form (4.31)-(4.32) with weight coefficients $\alpha_i(p) \geq 0$ for all $p \in \Omega$. It can be rewritten as

$$Z(p) = \sum_i U_i \alpha_i(p) \arctan(\Lambda_i) V_i^T. \tag{4.43}$$

Obviously, each term in (4.43) is still a thin SVD. Hence, this very structure of $Z(p)$ should be exploited in formulating its SVD. Based on the modification technique for thin SVDs proposed in [28], where the SVD of the sum of an SVD and a low rank updating matrix was considered, we succeeded in solving the problem in the general case as described below. However, for the sake of brevity of the presentation, we consider the situation that the sum is composed of three terms. We can think of the case when there are two parameters $(p, q) \in [p_0, p_1] \times [q_0, q_1]$ and we use bilinear interpolation. We can then write

$$\begin{aligned} Z(p,q) &= \sum_{i=1}^{3} U_i \alpha_i(p,q) S_i V_i^T \\ &= \begin{bmatrix} U_1 & U_2 & U_3 \end{bmatrix} \begin{bmatrix} \alpha_1(p,q)S_1 & & \\ & \alpha_2(p,q)S_2 & \\ & & \alpha_3(p,q)S_3 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \\ V_3^T \end{bmatrix}. \end{aligned} \tag{4.44}$$

Denote by P the columnwise orthogonal matrix whose columns span the intersection of the orthogonal complement of the subspace spanned by the columns of U_1 and the subspace spanned by both, U_2 and U_3. The matrix P, supposed to have the size $N \times n, (n \leq 2r)$ can be computed as the left singular vectors of $(I - U_1 U_1^T)[U_2 \quad U_3]$. Note that P depends on the ordering of the triplet U_1, U_2, U_3. Thanks to the projection, we have

$$\begin{bmatrix} U_1 & U_2 & U_3 \end{bmatrix} = \begin{bmatrix} U_1 & P \end{bmatrix} \begin{bmatrix} I & U_1^T U_2 & U_1^T U_3 \\ 0 & P^T(I - U_1 U_1^T) U_2 & P^T(I - U_1 U_1^T) U_3 \end{bmatrix}.$$

For the same reason, we can write

$$[V_1 \quad V_2 \quad V_3] = V_1 [I \quad V_1^T V_2 \quad V_1^T V_3].$$

Now replace the first and the last factors in (4.44) by the corresponding quantities, we get

$$\begin{aligned} Z(p,q) &= \begin{bmatrix} U_1 & P \end{bmatrix} \begin{bmatrix} I & U_1^T U_2 & U_1^T U_3 \\ 0 & P^T(I - U_1 U_1^T) U_2 & P^T(I - U_1 U_1^T) U_3 \end{bmatrix} \\ &\quad \begin{bmatrix} \alpha_1(p,q)S_1 & & \\ & \alpha_2(p,q)S_2 & \\ & & \alpha_3(p,q)S_3 \end{bmatrix} \begin{bmatrix} I \\ V_2^T V_1 \\ V_3^T V_1 \end{bmatrix} V_1^T \\ &= \begin{bmatrix} U_1 & P \end{bmatrix} K(p,q) V_1^T, \end{aligned} \tag{4.45}$$

where

$$K(p,q) = \begin{bmatrix} \alpha_1(p,q)\mathbf{S_1} + \alpha_2(p,q)\mathbf{U_1^T Z_2 V_1} + \alpha_3(p,q)\mathbf{U_1^T Z_3 V_1} \\ \alpha_2(p,q)\mathbf{P^T}(\mathbf{I} - \mathbf{U_1 U_1^T})\mathbf{Z_2 V_1} + \alpha_3(p,q)\mathbf{P^T}(\mathbf{I} - \mathbf{U_1 U_1^T})\mathbf{Z_3 V_1} \end{bmatrix} \in \mathbb{R}^{(r+n)\times r}. \tag{4.46}$$

Let us denote the thin SVD of $K(p,q)$ by

$$K(p,q) = \Phi(p,q)\Lambda(p,q)\Psi(p,q)^T. \tag{4.47}$$

By (4.45) and (4.47), the SVD of $Z(p,q)$ is therefore

$$Z(p,q) = \Big([U_1\ P]\Phi(p,q)\Big)\Lambda(p,q)\Big(V_1\Psi(p,q)\Big)^T.$$

Accordingly, the basis for the projection subspace, by (4.34), is

$$W(p,q) = W_0V_1\Psi(p,q)\cos(\Lambda(p,q)) + [U_1\ P]\Phi(p,q)\sin(\Lambda(p,q)). \tag{4.48}$$

Now, using the assumption of affine dependence (4.38), the reduced system is constructed similarly to the linear interpolation with single parameter case.

$$\begin{aligned}\hat{E}(p,q) &= W^T(p,q)EW(p,q) \qquad (4.49)\\ &= \sum_{i=1}^{\Phi_E} f_i^E(p,q)\cos(\Lambda(p,q))\Psi(p,q)^T\mathbf{V_1^T}\mathbf{W_0^T}\mathbf{E_i}\mathbf{W_0}\mathbf{V_1}\Psi(p,q)\cos(\Lambda(p,q))\\ &\quad+\sum_{i=1}^{\Phi_E} f_i^E(p,q)\cos(\Lambda(p,q))\Psi(p,q)^T\mathbf{V_1^T}\mathbf{W_0^T}\mathbf{E_i}[\mathbf{U_1}\ \mathbf{P}]\Phi(p,q)\sin(\Lambda(p,q))\\ &\quad+\sum_{i=1}^{\Phi_E} f_i^E(p,q)\sin(\Lambda(p,q))\Phi(p,q)^T[\mathbf{U_1}\ \mathbf{P}]^\mathbf{T}\mathbf{E_i}\mathbf{W_0}\mathbf{V_1}\Psi(p,q)\cos(\Lambda(p,q))\\ &\quad+\sum_{i=1}^{\Phi_E} f_i^E(p,q)\sin(\Lambda(p,q))\Phi(p,q)^T[\mathbf{U_1}\ \mathbf{P}]^\mathbf{T}\mathbf{E_i}[\mathbf{U_1}\ \mathbf{P}]\Phi(p,q)\sin(\Lambda(p,q)).\end{aligned}$$

The matrix $\hat{A}(p,q)$ is constructed analogously. The load matrix and output matrices are

$$\begin{aligned}\hat{B}(p,q) = W^T(p,q)B = &\sum_{i=1}^{\Phi_B} f_i^B(p,q)\cos(\Lambda(p,q))\Psi(p,q)^T\mathbf{V_1^T}\mathbf{W_0^T}\mathbf{B_i} \qquad (4.50)\\ &+\sum_{i=1}^{\Phi_B} f_i^B(p,q)\sin(\Lambda(p,q))\Phi(p,q)^T[\mathbf{U_1}\ \mathbf{P}]^\mathbf{T}\mathbf{B_i}\end{aligned}$$

and

$$\begin{aligned}\hat{C}(p,q) = CW(p,q) = &\sum_{i=1}^{\Phi_C} f_i^C(p,q)\mathbf{C_i}\mathbf{W_0}\mathbf{V_1}\Psi(p,q)\cos(\Lambda(p,q)) \qquad (4.51)\\ &+\sum_{i=1}^{\Phi_C} f_i^C(p,q)\mathbf{C_i}[\mathbf{U_1}\ \mathbf{P}]\Phi(p,q)\sin(\Lambda(p,q)),\end{aligned}$$

respectively. One can realize that all quantities emphasized with bold letters in (4.46), (4.49), (4.50), (4.51) are independent of p, q and therefore can be computed and stored beforehand. We now summarize the procedure in the form of offline-online decomposition as follows.

Offline We compute and store:

- W_0, W_1, W_2, W_3 corresponding to $(p_0, q_0), (p_1, q_0), (p_0, q_1), (p_1, q_1)$.
- $[U_1, \Lambda_1, V_1], [U_2, \Lambda_2, V_2], [U_3, \Lambda_3, V_3]$ representing $Log_{S_0}(S_1)$, $Log_{S_0}(S_2)$, $Log_{S_0}(S_3)$, respectively.
- $P \in \mathbb{R}^{N\times(r+n)}$ by SVD.
- All necessary quantities (in bold letters in (4.46)) for the matrix K.
- All necessary quantities (in bold letters in (4.49), (4.50), (4.51)) for the reduced matrices $\hat{E}, \hat{A}, \hat{B}, \hat{C}$.

Online Given any value $(p, q) \in [p_0, p_1] \times [q_0, q_1]$, we compute

- the matrix K as in (4.46),
- the thin SVD of K: $\Phi\Lambda\Psi^T = K$,
- the reduced system matrices by (4.49), (4.50), (4.51).

The computation cost of the online stage is $\mathcal{O}((r + n)r^2)$ which can be considered as $\mathcal{O}(r^3)$.

Remark The matrix P and therefore the matrix K depend on the choice of ordering of U_1, U_2, U_3 in the sum (4.44). However, $W(p, q)$ in (4.48) always spans the same subspace, since W_0 and the first factor in the second term of $W(p, q)$ are the same with respect to any order of U_1, U_2, U_3.

Using offline-online decomposition in order to deal with highly computational complexity is not a new idea. It has been widely used for parameterized PDEs [118, 117, 159, 132, 149]. For MOR, as mentioned in Section 3.4, it was applied to interpolation on manifolds of reduced system matrices [6, 5, 45, 8]. It was also used in [77], without interpolation, for order reduction of parameter-dependent systems. Consideration of interpolation of projection subspaces as interpolation on Grassmann manifolds was suggested in [7]. In this work, however, neither the offline-online decomposition nor a way for reducing the computational complexity have been used and given. Our strategy obviously eliminates the dependence of the computational complexity in the online stage on the full order, accelerates the computation and therefore enables the algorithm to be used in real time. This will be illustrated in the numerical example. □

4.2.3 Numerical Example

In this section, the applicability of the proposed method is illustrated through an example taken from the Oberwolfach model reduction benchmark collection. This model has been mentioned in Chapter 1. The spatial discretization in space of the heat transfer partial differential equation gives a system of the order 4257:

$$\begin{aligned} E\dot{T}(t) &= (A - h_{top}A_{top} - h_{bot}A_{bot} - h_{sid}A_{sid})T(t) + Bu(t) \\ y(t) &= CT(t), \end{aligned}$$

where E and A, the heat capacity and heat conductivity matrices, are symmetric, B is the load vector, C is the output matrix. We, however, retain only the first row of C in order to simplify the error evaluation. Matrices $A_{top}, A_{bot}, A_{sid}$ are the diagonal matrices derived from the discretization of the convection boundary conditions on the top, at the bottom and on the side with the corresponding film coefficients $h_{top}, h_{bot}, h_{sid}$. These coefficients may change according to the change of the surroundings of the chip and will be treated as the parameters of the model. The unknown T is the vector of temperatures. All system matrices are sparse. The reader is referred to [105, 152] for more details.

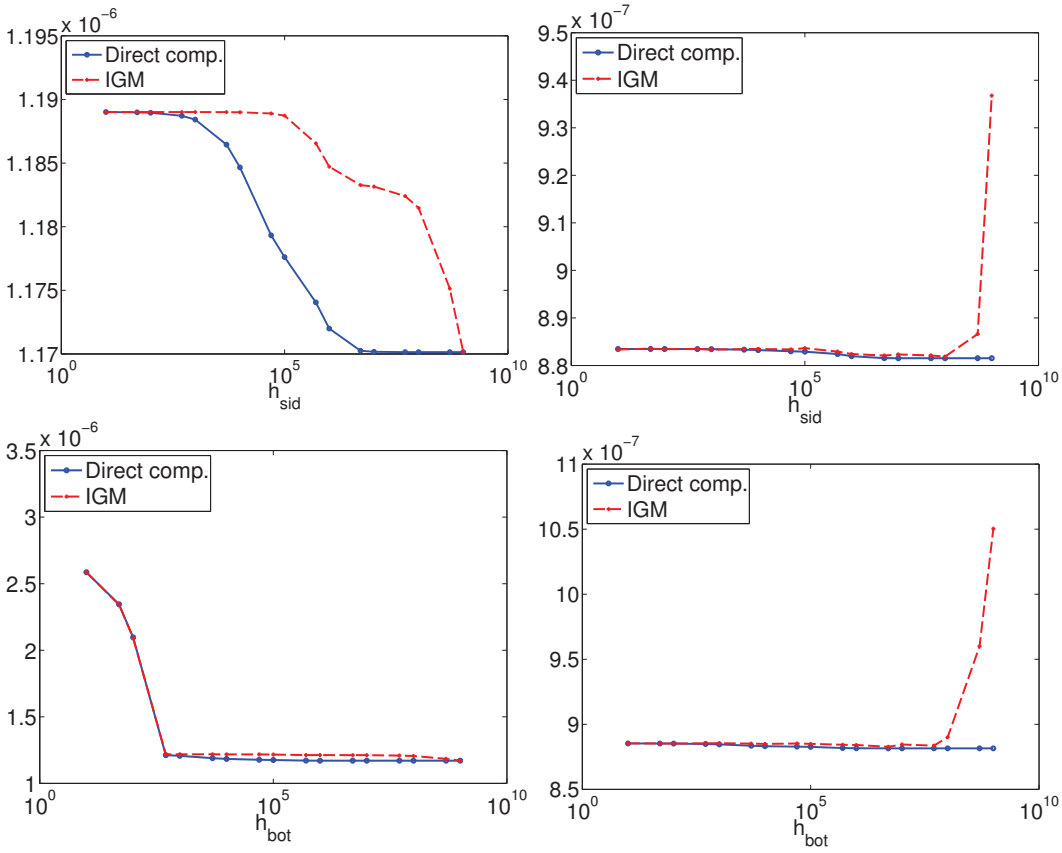

Figure 4.8: Relative errors using IGM vs. direct method; reduced order: 20 (left), 40 (right)

As the first test for linear interpolation of a single parameter, we fix two parameters $h_{top} = 5000, h_{bot} = 200$ and let the left h_{sid} vary from 10 to 10^9. Projection matrices corresponding to $h_{sid} = 10$ and $h_{sid} = 10^9$ are computed by the Krylov subspace method with the intention of matching moments about $s_0 = 100$. The reduced orders are 20 and 40. In both cases, the Krylov subspace at $h_{sid} = 10$ will be chosen as the contact "point". To check the quality of the approximation, we

compute the relative errors, which is defined by

$$\frac{\|H(\cdot) - \hat{H}(\cdot)\|_{\mathcal{H}_\infty}}{\|H(\cdot)\|_{\mathcal{H}_\infty}},$$

of the reduced transfer function at 17 points, $10, 50, 100, 500, 1000, \cdots, 10^9$, henceforth called points of interest. We use an approximation of the form

$$\|H(\cdot)\|_{\mathcal{H}_\infty} \approx \max_{w \in [w_{\min}, w_{\max}]} |H(iw)|,$$

where i denotes the imaginary unit. In our case, the frequency grid/range is chosen to be $-5000 : 10 : 5000$. These relative errors computed by interpolation on Grassmann manifolds (IGM) as aforementioned are then compared with the relative errors caused by direct computation, i.e., the reduced system is constructed by fixing the parameter at points of interest. We then perform the same test with h_{bot}. These errors are plotted in Figure 4.8.

The errors caused by both methods, if all conditions of using IGM are fulfilled, should be identical at the two ends of the parameter interval. As we can see, however, there is difference of the errors at the right end, when h_{sid} and h_{bot} are equal to 10^9. There are two possible reasons for this. From a theoretical point of view, as mentioned before, when using IGM the grid points should not be too far from the contact point. The distance between two points, which are actually two subspaces, can be computed, but one does not know exactly how small the distance should be, since this comes from the requirement for the local existence of the solution of a second order differential equation. If this distance is large, the logarithmic and exponential mappings may not work properly. As a consequence, IGM may not work well. From a computational point of view, in our case, when the parameter is large, the computation of the reduced system of the order 40 by IGM is rather sensitive to the perturbation of the data. Indeed, we perturbed the data by the amount of 10^{-15}, i.e., replaced 10 and 10^9 by $10(1 + i \times 10^{-15})$ and $10^9(1 + i \times 10^{-15}), i = -2, -1, 0, 1, 2$ and looked at the changes in the relative errors of both methods. The direct computation was stable with these changes of the data. For IGM, when the reduced order was 20, the resulting changes in the relative errors at both ends were from 10^{-12} to 10^{-14}. Meanwhile, when the reduced order was 40, the resulting changes in the error at the left end was around 10^{-9} and at the right end was 10^{-7}. This explains that the relative errors at the right end may vary 10^{-7} around its exact value, which can be seen in Figure 4.9. Note also that we use the logarithmic scale for the horizontal axes in all the plots, this also contributes to the intuitive sudden increase in the errors at the right end of the parameter interval. In the end, the amount of 10^{-7} variation in the relative error is not so large and therefore does not seriously affect the quality of the approximation.

In order to verify the computational reduction, the reduced system in state space representation form is computed at different parameter values. All the computations are performed with Matlab R2010b on a computer, using Linux/Debian 5.0,

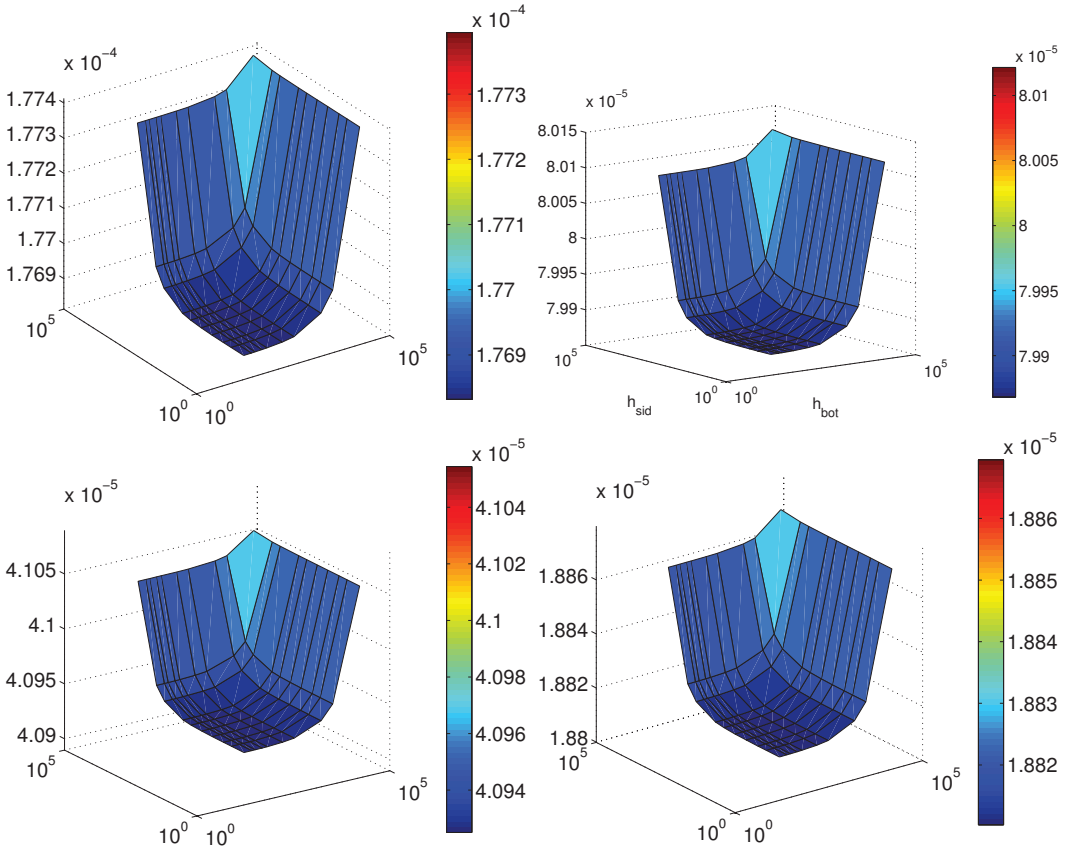

Figure 4.9: Relative errors using bilinear interpolation; reduced order: 10 (top-left), 20 (top-right), 30 (bottom-left), 40 (bottom-right)

and equipped with processor 2GHz 2GB AMD Athlon 64 X2. Since the computation time can slightly vary from point to point, we evaluate the reduced system at 99 points in $10 < h_{sid} < 10000$. The time, counted in seconds, consumed by the procedure with offline-online decomposition and that without offline-online decomposition, i.e., the original method proposed in [7] at different reduced orders are listed in Table 4.2. The acceleration factor is computed as the ratio between the time consumed by the two methods.

In the second test, we fix h_{top} and let h_{bot} and h_{sid} vary from 50 to 5×10^4 and 5 to 5×10^4, respectively. We examine the reduced system at a total of 100 grid points corresponding to typical values of parameters h_{bot} and h_{sid} mentioned in [105]. First of all, we compute 4 projection subspaces at $(h_{bot}, h_{sid}) = (50, 5), (5 \times 10^4, 5), (50, 5 \times 10^4)$ and $(5 \times 10^4, 5 \times 10^4)$ with the intention of matching moments about $s_0 = 100$. The reduced orders are 10, 20, 30 and 40. The subspace at $(50, 5)$ will be used as the contact point. The relative errors of the reduced models are plotted in Figure 4.9. The computation time is listed in Table 4.3.

The decrease in the error when the reduced order increases shows that our pro-

Table 4.2: Computation time: linear interpolation

Reduced order	10	20	30	40
With off-on decomp.	0.0479	0.0508	0.0563	0.0675
Without off-on decomp.	0.9468	3.0626	5.6700	7.2910
Acceleration factor	19.7851	60.3121	100.7854	107.9589

gram works properly. However, the fact that this decrease is not so considerable suggests that, in this case, the effort to increase the reduced order does not bring much achievement.

We can realize that the advantage of using the proposed method is different in the linear case and general case as the reduced order varies. In the linear case, the higher the reduced order is, the bigger the acceleration factor is, while in the general case, it gets smaller. The reason is that in the linear case, the procedure is simple, we do not have to compute matrix K as well as its SVD. Therefore, when the reduced order is higher, we can take advantage of this fact. Meanwhile, in the general case, the computation of K and its SVD slows down the online stage as the reduced order increases.

Table 4.3: Computation time: bilinear interpolation

Reduced order	10	20	30	40
With off-on decomp.	0.0674	0.1982	0.4562	0.8372
Without off-on decomp.	1.0480	3.0708	5.8994	7.5586
Acceleration factor	15.5415	15.4934	12.9309	9.0287

Chapter 5

Conclusion

In this thesis, model order reduction of parameter-dependent systems has been investigated. All methods are based on the extension of standard MOR methods or a combination of one of them with an interpolation technique. We have focused on the second direction.

As the first effort, we have combined the balanced truncation method with spline interpolation to symbolically preserve the dependence of the considered model on parameters. This approach does not require an explicit expression of the dependence. However, it is applicable only for reachable, observable and stable systems. We have shown that the error between the original system and the reduced system is bounded from above, and this bound is theoretically explicit and *a priori*. It is the sum of, up to a factor, the error caused by balanced truncation and one caused by interpolation. If the considered system is highly varying, the derivative of its transfer function is large; in such case, we have suggested that this method should not be applied. In addition, the stability is preserved during the reduction process. Although the actual process produces the external description of the reduced system, a state space representation for the resulting reduced system is constructed by appropriately choosing the end conditions and some computations.

Our second effort concentrated more on computational aspects. We have projected the original system on Krylov subspaces. For parameter-dependent systems, these subspaces vary with the parameter and it turns out that they lie on a Grassmann manifold. To deal with the dependence on parameters, we have interpolated a set of pre-computed projection subspaces. However, the standard interpolation procedure does not work since one has to maintain the rank of the bases of these subspaces. We had to first map the data on the underlying Grassmann manifold to a tangent space, then interpolate on that space and finally map the interpolated data back to the Grassmann manifold. The connection between a Grassmann manifold and its tangent spaces is determined by exponential and logarithmic mappings. By exploiting the structure of these mappings, analyzing the structure of sums of SVDs and by decomposing the process into offline and online stages, we have considerably reduced the computation cost of the online stage and therefore enabled this procedure to be usable in real time.

In the following we analyze some directions which may be investigated in the coming time.

As mentioned before, it is still a challenge to derive an error bound of the method using cubic spline interpolation for the MIMO case. We presume that sophisticated new linear algebra results will be needed in order to solve this problem.

The error of using spline interpolation can probably satisfy a given tolerance thanks to controlling the local error and determining the region of validity on which the error is still less than a given number regardless the change of parameter. This needs further investigations into the application of results on the effect of perturbation on ROMs.

Our second result on interpolating on Grassmann manifolds is applicable for one-sided projection reduction methods. In some case, a MOR method can only be formulated as a two-sided projection such as balanced truncation. In other cases, two-sided projection always gives a better result than one-sided projection does. It is therefore a need to extend the interpolation on Grassmann manifold framework for such methods. A Gram-Schmidt-like method may be an option but the way to imbed it in the algorithm such that the online stage is able to be used in real time is still unknown.

The sensitivity of computing ROMs using IGM needs to be investigated in detail. Especially, one needs to know in each specific case the distance between the interpolation points and the contact point in which the procedure is still effective.

Bibliography

[1] T. Abrudan, J. Eriksson, and V. Koivunen. Conjugate gradient algorithm for optimization under unitary matrix constraint. *Signal Processing*, 89:1704–1714, 2009.

[2] P. A. Absil, R. Mahony, and R. Sepulchre. Riemannian geometry of Grassmann manifolds with a view on algorithmic computation. *Acta Appl. Math.*, 80:199–220, 2004.

[3] E. Acar, S. Nassif, Y. Liu, and L. T. Pileggi. Time-domain simulation of variational interconnect models. In *Proceedings Inter. Symp. Quality Elec. Desi.*, pages 419–424, 2002.

[4] J. H. Ahlberg and E. N. Nilson. Convergence properties of the spline fit. *J. SIAM*, 11(1):95–104, 1963.

[5] D. Amsallem. *Interpolation on Manifolds of CFD-based Fluid and Finite Element-based Structural Reduced-Order Models for On-line Aeroelastic Predictions*. PhD thesis, Stanford University, 2010.

[6] D. Amsallem, J. Cortial, K. Carlberg, and C. Farhat. A method for interpolating on manifolds structural dynamics reduced-order models. *Int. J. Numer. Meth. Engng*, 80(9):1241–1258, 2009.

[7] D. Amsallem and C. Farhat. Interpolation method for adapting reduced-order models and application to aeroelasticity. *AIAA Journal*, 46(7):1803–1813, 2008.

[8] D. Amsallem and C. Farhat. An online method for interpolating linear reduced-order models. *SIAM J. Sci. Comput.*, 33(5):2169–2198, 2011.

[9] C. A. Andrews, J. M. Davies, and G. R. Schwarz. Adaptive data compression. In *Proceeding of the IEEE*, pages 267–277, 1967.

[10] A. C. Antoulas. *Approximation of Large-scale Dynamical Systems*. Advances in Design and Control DC-06. SIAM, Philadelphia, 2005.

[11] A. C. Antoulas, D. C. Sorensen, and Y. Zhou. On the decay rate of Hankel singular values and related issues. *Syst. Control Lett.*, 46(5):323–342, 2002.

[12] W. E. Arnoldi. The principle of minimized iterations in the solution of the matrix eigenvalue problem. *Q. Appl. Math.*, 9:17–29, 1951.

[13] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache. Geometric means in a novel vector space structure on symmetric positive-definite matrices. *SIAM J. Matrix Anal. A.*, 29(1):328–347, 2007.

[14] U. Baur, C. Beattie, P. Benner, and S. Gugercin. Interpolatory projection methods for parameterized model reduction. *SIAM J. Sci. Comput.*, 33(5):2489–2518, 2011.

[15] U. Baur and P. Benner. Factorized solution of Lyapunov equations based on hierarchical matrix arithmetic. *Computing*, 78(3):211–234, 2006.

[16] U. Baur and P. Benner. Parametrische Modellreduktion mit dünnen Gittern. In *GMA FA 1.30 Workshop*, pages 262–271, 2008.

[17] U. Baur and P. Benner. Modellreduktion für parametrisierte Systeme durch balanciertes Abscheiden und Interpolation. *Automatisierungtechnik*, 57(8):411–419, 2009.

[18] P. Benner. Solving large-scale control problems. *IEEE Contr. Syst. Mag.*, 24(1):44–59, 2004.

[19] P. Benner and T. Breiten. On $\mathcal{H}_2$-model reduction of linear parameter-varying systems. In *Proceedings in Applied Mathematics and Mechanics*, 2011.

[20] A. Berman and R. J. Plemmons. *Nonnegative Matrices in the Mathematical Sciences*. Classics in Applied Mathematics. SIAM, Philadelphia, 1994.

[21] G. Berzook, P. Holmes, and J. L. Lumley. On the relation between low-dimensional models and the dynamics of coherent structures in the turbulent wall layer 1. *Theoret. Comput. Fluid Dynamics*, 4:255–269, 1993.

[22] G. Berzook, P. Holmes, and J. L. Lumley. The proper orthogonal decomposition in the analysis of turbulent flows. *Annu. Rev. Fluid Mech.*, 25:539–575, 1993.

[23] Å. Björck and C. C. Paige. Solving linear least squares problems by Gram-Schmidt orthogonalization. *BIT*, 7(1):1–21, 1967.

[24] Å. Björck and C. C. Paige. Loss and recapture of orthogonality in the modified Gram-Schidt algorithm. *SIAM J. Mat. Anal. Appl.*, 13(1):176–190, 1992.

[25] B. N. Bond and L. Daniel. A piecewise-linear moment-matching approach to parameterized model-order reduction for highly nonlinear systems. *IEEE T. Comput. Aid. D.*, 26(12):1467–1480, 2007.

[26] C. D. Boor. *A Practical Guide to Splines*. Applied Mathematical Sciences. Springer-Verlag, New York, 1978.

[27] W. M. Boothby. *An Introduction to Differentiable Manifolds and Riemannian Geometry*. Pure and Applied Mathematics. Academic Press, London, 1975.

[28] M. Brand. Fast low-rank modifications of the thin singular value decomposition. *Linear Algebra. Appl.*, 415:20–30, 2006.

[29] T. Bui-Thanh, K. Willcox, and O. Ghattas. Parametric reduced-order models for probabilistic analysis of unsteady aerodynamic applications. *AIAA Journal*, 46(10):2520–2529, 2008.

[30] H. J. Bungartz and M. Griebel. Sparse grids. *Acta Numerica*, 13:147–269, 2004.

[31] F. M. Callier and C. A. Desoer. *Linear System Theory*. Springer Texts in Electrical Engineering. Springer, New York, 1991.

[32] D. Celo, P. Gunupudi, R. Khazaka, D. Walkey, T. Smy, and M. Nakhla. Fast simulation of steady-state temperature distributions in electronic components using multidimensional model reduction. *IEEE T. Compon. Pack. T.*, 28(1):70–79, 2005.

[33] Y. Chahlaoui and P. Van Dooren. A collection of benchmark examples for model reduction of linear time invariant dynamical systems. *MIMS Eprint* *http://eprints.ma.man.ac.uk/1040/*, 2002.

[34] Y. Chahlaoui, D. Lemonnier, A. Vandendorpe, and P. Van Dooren. Second-order balanced truncation. *Linear Algebra. Appl.*, 415(2-3):373–384, 2006.

[35] E. Chiprout and M. S. Nakhla. *Asymptotic Waveform Evaluation and Moment Matching for Interconnect Analysis*. The Springer International Series in Engineering and Computer Science 252. Springer, Berlin, 1994.

[36] C. R. Cockrell and F. B. Beck. Asymptotic waveform evaluation technique for frequency domain electromagnetic analysis. *NASA Technical Memorandum*, pages 1–13, 1996.

[37] L. Codecasa. A novel approach for generating boundary condition independent compact dynamic thermal networks of packages. *IEEE T. Compon. Pack. T.*, 28(4):593–604, 2005.

[38] M. Condon and R. Ivanov. Empirical balanced truncation of nonlinear systems. *J. Nonlinear Sci.*, 14(5):405–414, 2004.

[39] M. J. Corless and A. E. Frazho. *Linear System and Control: An Operator perspective*. Pure and Applied Mathematics. Marcel Dekker, Inc., New York Basel, 2003.

[40] J. Cullum and W. E. Donath;. A block Lanczos algorithm for computing the q algebraically largest eigenvalues and a corresponding eigenspace of large, sparse, real symmetric matrices. In *IEEE Conf. Decision and Control, incl. 13th Symp. Adaptive Processes*, pages 505–509, 1974.

[41] J. Cullum and T. Zhang. Two-sided Arnoldi and nonsymmetric Lanczos algorithms. *SIAM J. Matrix Anal. A.*, 24(2):303–319, 2002.

[42] M. I. Curtis. *Matrix groups*. Springer New York, 1984.

[43] L. Daniel, O. C. Siong, L. S. Chay, K. H. Lee, and J. White. A multiparameter moment-matching model-reduction approach for generating geometrically parameterized interconnect performance models. *IEEE T. Comput. Aid. D.*, 23(5):678–693, 2004.

[44] P. Davies and N. Higham. A Schure-Parlett algorithm for computing matrix functions. *SIAM J. Matrix Anal. A.*, 25(2):464–485, 2003.

[45] J. Degroote, J. Vierendeels, and K. Willcox. Interpolation among reduced-order matrices to obtain parameterized models for design, optimization and probabilistic analysis. *Int. J. Numer. Meth. Fl.*, 63:207–230, 2010.

[46] J. W. Demmel. *Applied Numerical Linear Algebra*. SIAM, Philadelphia, 1997.

[47] A. Edelman, T. A. Arias, and S. T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM J. Matrix Anal. A..*, 20(2):303–353, 1998.

[48] I. M. Elfadel and D. D. Ling. A block rational Arnoldi algorithm for multipoint passive model-order reduction of multiport RLC networks. In *Digest of Technical Papers of 1997 IEEE/ACM Inter. Conf. Computer-Aided Design*, pages 66–71, 1997.

[49] R. Engelking. *Outline of General Topology*. North-Holland, Philadelphia; PWN-Polish Scientific, 1968.

[50] D. F. Enns. Model reduction with balanced realization: An error bound and a frequency weighted generalization. In *Proceeding of the 23rd IEEE Conference on Decision and Control, 1984*, pages 127–132, 1984.

[51] O. Farle, V. Hill, P. Nickel, and R. Dyczij-Edlinger. An Arnoldi-type algorithm for parametric finite element modelling of microwave components. In *PAMM: Proceedings Appl. Math Mech.*, pages 655–656, 2005.

[52] H. Fassbender and P. Benner. Passivity preserving model reduction via a structured Lanczos method. In *IEEE Conf. Comp. Aided Cont. Syst. Degs.*, pages 8–13, 2006.

[53] P. Feldmann and R. W. Freund. Efficient linear circuit analysis by Padé approximation via Lanczos process. *IEEE T. Comput. Aid. D.*, 14(5):639–649, 1995.

[54] P. Feldmann and R. W. Freund. Reduced-order modeling of large linear subcircuits via a block Lanczos algorithm. In *DAC '95. 32nd Conf. Design Automation*, pages 474–479, 1995.

[55] L. Feng, E. B. Rudnyi, and J. G. Korvink. Preserving the film coefficient as a parameter in the compact thermal model for fast electrothermal simulation. *IEEE T. Comput. Aid. D.*, 24(12):1838–1847, 2005.

[56] J. Ferrer, M. I. García, and F. Peurta. Differentiable families of subspaces. *Linear Algebra Appl.*, 199:229–252, 1994.

[57] R. W. Freund. Krylov subspace methods for reduced-order modeling in circuit simulation. *J. Comp. Appl. Math.*, 123(1-2):395–421, 2000.

[58] R. W. Freund, M. H. Gutknecht, and N. M. Nachtigal. An implementation of the look-ahead Lanczos algorithm for non-Hermitian matrices. *SIAM J. Sci. Comput.*, 14(1):137–158, 1993.

[59] K. Gallivan, E. Grimme, and P. Van Dooren. Asymptotic waveform evaluation via a Lanczos method. *Appl. Math. Lett.*, 7(5):75–80, 1994.

[60] K. Gallivan, E. Grimme, and P. Van Dooren. A rational Lanczos algorithm for model reduction. *Numer. Algorithms*, 12:33–63, 1996.

[61] K. Gallivan, A. Vandendorpe, and P. Van Dooren. Model reduction of MIMO systems via tangential interpolation. *SIAM J. Matrix Anal. A.*, 26(2):328–349, 2004.

[62] E. Gallopoulos and Y. Saad. Efficient solution of parabolic equations by Krylov approximation methods. *SIAM J. Sci. Stat. Comp.*, 13(5):1236–1264, 1992.

[63] Y. M. Gao and X. H. Wang. Criteria for generalized diagonally dominant and M-matrices. *Linear Algebra Appl.*, 268:257–268, 1992.

[64] Y. M. Gao and X. H. Wang. Criteria for generalized diagonally dominant and M-matrices ii. *Linear Algebra Appl.*, 248:339–353, 1996.

[65] W. K. Gawronski. *Advanced Structural Dynamics and Active Control of Structures*. Mechanical Engineering. Springer, New York, 2004.

[66] L. Giraud and J. Langou. The loss of orthogonality in the Gram-Schmidt orthogonality process. *Comput. Math. Appl.*, 50:1069–1075, 2005.

[67] K. Glover. All optimal Hankel-norm approximations of linear multivariable systems and their l^∞-error bound. *Int. J. Control*, 39(6):1115–1193, 1984.

[68] G. H. Golub and C. F. Van Loan. *Matrix Computations, Third edition*. Johns Hopkins University Press, Baltimore, 1996.

[69] W.B. Gragg and A. Lindquist. On the partial realization problem. *Linear Algebra Appl.*, 50:277–319, 1983.

[70] E. J. Grimme. *Krylov Projection Methods for Model Reduction*. PhD thesis, University of Illinois at Urbana-Champaign, 1997.

[71] E. J. Grimme, D. C. Sorensen, and P. Van Dooren. Model reduction of state space systems via an implicitly restarted lanczos method. *Numer. Algorithms*, 12:1–31, 1996.

[72] S. Gugercin. An iterative SVD-Krylov based method for model reduction of large-scale dynamical systems. *Linear Algebra Appl.*, 428(8-9):1964–1986, 2008.

[73] S. Gugercin and A. C. Antoulas. A survey of model reduction by balanced truncation and some new results. *Int. J. Control*, 77(8):748–766, 2004.

[74] R. C. Gunning and H. Rossi. *Analytic Functions of Several Complex Variables*. Prentice-Hall Series in Modern Analysis. Prentice-Hall, Englewood Cliffs, N. J., 1965.

[75] P. Gunupudi, R. Khazaka, and M. Nakhla. Analysis of transmission line circuits using multidimensional model reduction techniques. *IEEE T. Adv. Pack.*, 25(2):174–180, 2002.

[76] P. Gunupudi and M. Nakhla. Multi-dimensional model reduction of VLSI interconnects. In *Proceedings of IEEE 2000 Custom Inte. Circ. Conference*, pages 499–502, 2000.

[77] B. Haasdonk and M. Ohlberger. Efficient reduced models and a posteriori error estimation for parametrized dynamical systems by offline/online decomposition. *Math. Comp. Model. Dyn.*, 17(2):145–161, 2011.

[78] C. A. Hall. On error bound for spline interpolation. *J. Approx. Theory*, 1:209–218, 1968.

[79] G. Hämmerlin and K. H. Hoffmann. *Numerical Analysis*. Undergraduate Texts in Mathematics. Springer, New York, 1991.

[80] C. Hartmann, V. M. Vulcanov, and C. Schütte. Balanced truncation of linear second-order systems: a Hamiltonian approach. *Multiscale Model. Sim.*, 8(4):1348–1367, 2010.

[81] A. Hay, J. Borggaard, I. Akhtar, and D. Pelletier. Reduced-order models for parameter dependent geometries based on shaped sensitivity analysis. *J. Comput. Phys.*, 229:1327–1353, 2010.

[82] S. Helgason. *Differential Geometry, Lie Groups, and Symmetric Spaces*. Pure and applied mathematics. Academic Press, New York, 1978.

[83] N. J. Higham. The scaling and squaring method for the matrix exponential revisited. *SIAM J. Matrix Anal. A.*, 26(4):1179–1193, 2005.

[84] D. Hinrichsen and A. J. Pritchard. *Mathematical Systems Theory I*. Text in applied mathematics. Springer, Berlin Heidelberg, 2005.

[85] P. Holmes. Can dynamical systems approach turbulence? *Lect. Notes. Phys.*, 357:195–249, 1990.

[86] P. Holmes, J. L. Lumley, and G. Bekooz. *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*. Cambridge Monographs on Mechanics. Cambridge University Press, 1996.

[87] C. Homescu, L. R. Petzold, and R. Serban. Error estimation for reduced-order models of dynamical systems. *SIAM J. Numer. Anal.*, 43(4):1693–1714, 2005.

[88] Y. S. Hung and K. Glover. Optimal Hankel-norm approximation of stable systems with first-order stable weighting functions. *Syst. Control Lett.*, 7:165–172, 2004.

[89] I. M. James. *The Topology of Stiefel manifolds*. London Mathematical Society Lecture Note Series. Cambridge University Press, 1976.

[90] Z. X. Jia. A variation on the block Arnoldi method for large unsymmetric matrix eigenproblems. *Acta Math. Appl. Sin-E*, 14(4):425–432, 1998.

[91] E. Jonckheere and R. Li. Generalization of optimal Hankel-norm and balanced model reduction by bilinear mapping. *Int. J. Control*, 45(5):1751–1769, 1987.

[92] M. Kahlbacher and S. Volkwein. Galerkin proper orthogonal decomposition methods for parameter dependent elliptic systems. *Discussiones Mathematicae: Differential Inclusions, Control and Optimization*, 27(1):95–117, 2007.

[93] K. Karhunen. Zur Spektraltheorie stochastischer Prozesse. *Ann. Acad. Sci. Finnicae, Ser. A1*, 1:34 pages, 1946.

[94] H. M. Kim and R. R. Craig Jr. Structural dynamics analysis using an unsymmetric block Lanczos algorithm. *Int. J. Numer. Meth. Eng.*, 26(10):2305–2318, 1988.

[95] L. Yu. Kolotilina. Bounds for infinity norm of the inverse for certain M- and H-matrices. *Linear Algebra Appl.*, 430:692–702, 2009.

[96] D. D. Kosambi. Statistics in function space. *J. Indian Math. Soc.*, 7:76–88, 1943.

[97] D. Kubalińska. *Optimal Interpolation-Based Model Reduction*. PhD thesis, Universität Bremen, 2008.

[98] K. Kunisch and S. Volkwein. Control of the Burgers equation by a reduced-order approach using proper orthogonal decomposition. *J. Optimiz. Theory. App.*, 102(2):345–371, 1999.

[99] K. Kunisch and S. Volkwein. Optimal snapshot location for computing POD basis functions. *ESAIM-Math. Model. Num.*, 44(3):509–529, 2010.

[100] S. Lall, P. Krysl, and J. E. Marsden. Structure-preserving model reduction for mechanical systems. *Physica D*, 184:304–318, 2003.

[101] S. Lall, J. E. Marsden, and S. Glavaški. A subspace approach to balanced truncation for model reduction of nonlinear control systems. *Int. J. Robust Nonlin.*, 12(6):519–535, 2002.

[102] C. Lanczos. Asymptotic waveform evaluation via a Lanczos method. *J. Res. Nat. Bur. Stand.*, 45(4):255–282, 1950.

[103] S. Lang. *Differential and Riemannian Manifolds*. Graduate Texts in Mathematics. Springer, New York, 1995.

[104] B. Larangot, C. Rossi, T. Camps, A. Bertholds, P. Q. Pham, D. Briand, N. F. de Rooij, M. Puig-Vidal, P. Miribel, E. Montané, E. López, and J. Samitier. Solid propellant micro rockets - towards a new type of power MEMS. In *Nanotech '02, AIAA paper*, page 9, 2002.

[105] C. J. M. Lasance. Two benchmarks to facilitate the study of compact thermal modeling phenomena. *IEEE T. Compon. Pack. T.*, 24(4):559–565, 2001.

[106] S. Lefteriu, A. C. Antoulas, and A. C. Ionita. Parametric model order reduction from measurements. In *The IEEE 19th Conference on Electrical Performance of Electronic Packaging and Systems*, pages 193–196, 2010.

[107] A. T. Leung and R. Khazaka. Parametric model order reduction technique for design optimization. In *IEEE Int. Symp. Circ. Syst. ISCAS 2005*, volume 2, pages 1290–1293, 2005.

[108] X. Li, P. Li, and L. T. Pileggi. Parameterized interconnect order reduction with explicit-and-implicit multi-parameter moment matching for inter/intra-die variations. In *2005 IEEE/ACM Int. Conf. Comp. Aid. Desi.*, pages 805–811. IEEE Computer Society, 2005.

[109] Y. T. Li, Z. Bai, and Y. Su. A two-directional Arnoldi process and its application to parametric model order reduction. *J. Comput. Appl. Math.*, 226:10–21, 2009.

[110] Y. T. Li, Z. Bai, Y. Su, and X. Zeng. Model order reduction of parameterized interconnect networks via a two-directional Arnoldi process. *IEEE T. Comput. Aid. D.*, 27(9):1571–1582, 2008.

[111] T. Lieu and C. Farhat. Adaptation of aeroelastic reduced-order models and application to an F-16 configuration. *AIAA Journal*, 45(6):1244–1257, 2007.

[112] Y. Liu and B. D. O. Anderson. Singular perturbation approximation of balanced systems. In *Proceedings of the 28th IEEE Conf. Decisions and Control*, pages 1355–1360, 1989.

[113] Y. Liu, L.T. Pileggi, and A.J. Strojwas. Model order-reduction of RC(L) interconnect including variational analysis. In *Proceedings 36. Desi. Auto. Conf. 1999*, pages 201–206, 1999.

[114] M. Loève. Functions aleatoire de second ordre. *C. R. Acad. Sci. Paris.*, 220, 1945.

[115] P. Losse and V. Mehrmann. Controllability and observability of second order descriptor systems. *SIAM J. Control Optim.*, 47(3):1351–1379, 2008.

[116] J. L. Lumley. The structure of inhomogeneous turbulent flows. *In: A.M. Yaglom and V.I. Tatarski, Editors, Atmospheric Turbulence and Radio Wave Propagation*, pages 166–178, 1967.

[117] Y. Maday. Reduced basis method for the rapid and reliable solution of partial differential equations. In *Proceedings of the 10th International Congress of Mathematicians,*, volume 3, pages 1255–1270. European Mathematical Society, 2006.

[118] Y. Maday, A.T. Patera, and D.V. Rovas. A blackbox reduced-basis output bound method for noncoercive linear problems. *Stud. Math. Appl.*, 31:533–569, 2002.

[119] J. H. Manton. Optimization algorithms exploiting unitary constraints. *IEEE T. Signal. Proces.*, 50(3):635–650, 2002.

[120] M. Moakher. A differential geometric approach to the geometric mean of symmetric positive-definite matrices. *SIAM J. Matrix Anal. A.*, 26(3):735–747, 2005.

[121] B. C. Moore. Principal component analysis in linear systems: controllability, observability, and model reduction. *IEEE T. Automat. Contr.*, AC-26(1):17–32, 1981.

[122] C. Moosmann. *ParaMOR- Model Order Reduction for Parameterized MEMS applications*. PhD thesis, Albert-Ludwigs-University of Freiburg, 2007.

[123] N. Morača. Upper bounds for the infinity norm of the inverse of SDD and S-SDD matrices. *J. Comput. Appl. Math.*, 206:666–678, 2007.

[124] N. Morača. Bound for norms of the matrix inverse and the smallest singular value. *Linear Algebra Appl.*, 429:2589–2601, 2008.

[125] K. Ogata. *Systems Dynamics*. Pearson Education, Inc., New Jersey, 2004.

[126] A. Ohara, N. Suda, and S. Amari. Dualistic differential geometry of positive definite matrices and its applications to related problems. *Linear Algebra Appl.*, 247(1):31–53, 1996.

[127] K. H. A. Olsson. *Model Order Reduction in FEMLAB by Dual Rational Arnoldi*. Thesis for the Degree of Licentiate of Engineering. Chalmers Uni. Technology and Göteborg Uni., 2002.

[128] P. C. Opdenacker and E. A. Jonckheere. A contraction mapping preserving balanced reduction scheme and its infinity norm error bounds. *IEEE T. Circuits Syst.*, 35(2):184–189, 1988.

[129] A. M. Ostrowski. Über die Determinanten mit überwiegender Hauptdiagonale. *Comment. Math. Helv.*, 10:69–96, 1937.

[130] H. Panzer, J. Mohring, R. Eid, and B. Lohmann. Parametric model order reduction by matrix interpolation. *Automatisierungtechnik*, 58(8):475–484, 2010.

[131] B. N. Parlett, D. R. Taylor, and Z. A. Liu. A look-ahead Lanczos algorithm for unsymmetric matrices. *Math. Comput.*, 44(169):105–124, 1985.

[132] A. T. Patera and G. Rozza. *Reduced Basis Approximation and A Posteriori Error Estimation for Parametrized Partial Differential Equations, Ver. 1.0*. MIT Pappalardo Graduate Monographs in Mechanical Engineering. MIT, Massachusetts, 2007.

[133] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philos Mag.*, 2(6):559–572, 1901.

[134] T. Penzl. A cyclic low rank smith method for large spare Lyapunov equations. *SIAM J. Sci. Comput.*, 21(4):1401–1418, 2000.

[135] J. R. Phillips, L. Daniel, and L. M. Silveira. Guaranteed passive balancing transformations of model order reduction. *IEEE T. Comput. Aid. D.*, 22(8):1027–1041, 2003.

[136] L. T. Pillage and R. A. Rohrer. Asymptotic waveform evaluation for timing analysis. *IEEE T. Comput. Aid. D.*, 9(4):352–366, 1990.

[137] S. Prajna. POD model reduction with stability guarantee. In *Proceedings of 42nd IEEE Conference on Decision and Control*, volume 5, pages 5254–5258, 2003.

[138] L. Qi. Some simple estimates for singular values of a matrix. *Linear Algebra Appl.*, 56:105–119, 1984.

[139] I. U. Rahman, I. Drori, and V. C. Stodden. Multiscale representations for manifold-valued data. *Multiscale Model. Sim.*, 4:1201–1232, 2005.

[140] M. Rathinam and L. R. Petzold. A new look at proper orthogonal decomposition. *SIAM J. Numer. Anal.*, 41(5):1893–1925, 2003.

[141] T. Reis and T. Stykel. Balanced truncation model reduction of second-order systems. *Math. Comp. Model. Dyn.*, 14(5):391–406, 2008.

[142] T. Reis and T. Stykel. Pabtec: passivity-preserving balanced truncation for electrical circuits. *IEEE T. Comput. Aid. D.*, 29(9):1354–1367, 2010.

[143] T. Reis and T. Stykel. Positive real and bounded real balancing for model reduction of descriptor systems. *Int. J. Control*, 83(1):74–88, 2010.

[144] A. Rosenfeld and A. C. Kak. *Digital Picture Processing, 2nd Edit.* Academic Press, Inc. Orlando, FLorida, USA, 1982.

[145] C. Rossi. Micropropellant for space - A survey of MEMS-based micro thruster and their solid propellant technology. *Sensors Update*, 10:997–1013, 2002.

[146] C. Rossi, S. Orieux, B. Lanrangot, T. Do Conto, and D. Estève. Design, fabrication and modeling of solid propellant microrocket-application to micropropulsion. *Sensor. Actuator.*, 9:125–133, 2002.

[147] C. W. Rowley. Model reduction for fluids using balanced proper orthogonal decomposition. *Int. J. Bifurcat. Chaos*, 15(3):997–1013, 2005.

[148] C. W. Rowley, T. Colonius, and R. M. Murray. Model reduction for compressible flows using POD and Galerkin projection. *Physica D: Nonlinear Phenomena*, 189(1-2):115–129, 2004.

[149] G. Rozza, T. M. Lassila, and A. Manzoni. Reduced basis approximation for shape optimization in thermal flows with a parametrized polynomial geometric map. *Lecture Notes in Computational Science and Engineering*, 74:307–315, 2011.

[150] E. B. Rudnyi, T. Bechtold, J. G. Korvink, and C. Rossi. Solid propellant microthruster: Theory of operation and modelling strategy. In *Strategy, Nanotech 2002 - At the Edge of Revolution, 2002*, 2002.

[151] E. B. Rudnyi, L. H. Feng, M. Salleras, S. Marco, and J. G. Korvink. Error indicator to automatically generate dynamic compact parametric thermal models. In *Proceedings of 11th International Workshop on Thermal Investigations ICs and Systems*, pages 139–145, 2005.

[152] E. B. Rudnyi and J. G. Korvink. Thermal model. In *Oberwolfach Model Reduction Benchmark Collection* `http://portal.uni-freiburg.de/imteksimulation/downloads/benchmark`. 2008.

[153] A. Ruhe. Implementation aspects of band Lanczos algorithms for computation of eigenvalues of large sparse symmetric matrices. *Math. Comput.*, 33(146):680–687, 1979.

[154] A. Ruhe. The two-sided Arnoldi algorithm for nonsymmetric eigenvalue problems. *Lect. Notes Math.*, 973:104–120, 1983.

[155] Y. Saad. Analysis of some Krylov subspace approximations to the matrix exponential operator. *SIAM J. Numer. Anal.*, 29(1):209–228, 1992.

[156] M. Sadkane. Block-Arnoldi and Davidson methods for unsymmetric large eigenvalue problems. *Numer. Math.*, 64(1):195–211, 1993.

[157] B. Salimbahrami, B. Lohmann, T. Bechtold, and J.G. Korvink. A two-sided Arnoldi algorithm with stopping criterion and MIMO selection procedure. *Math. Comp. Model. Dyn.*, 11(1):79–93, 2005.

[158] R. Samara, I. Postlethwaitea, and D. Gu. Model reduction with balanced realizations. *Int. J. Control*, 62(1):33–64, 1995.

[159] S. Sen, K. Veroy, D. B. P. Huynh, S. Deparis, N. C. Nguyen, and A. T. Patera. Natural norm a posteriori error estimators for reduced basis approximations. *J. Comput. Phys.*, 217:37–62, 2006.

[160] R. Serban, C. Homescu, and L. R. Petzold. The effect of problem perturbations on nonlinear dynamical systems and their reduced order models. *SIAM J. Sci. Comput.*, 29(6):2621–2643, 2007.

[161] L. Sirovich. Turbulence and the dynamics of coherent structures. *Q. Appl. Math.*, XLV(3):561–590, 1987.

[162] L. Sirovich, K. S. Ball, and R. A. Handler. Propagating structures in wall-bounded turbulent flows. *Theor. Comp. Fluid Dyn.*, 2:307–317, 1991.

[163] E. D. Sontag. *Mathematical Control Theory: Deterministic Finite Dimensional Systems*. Texts in Applied Mathematics. Springer, New York, 1990.

[164] D. C. Sorensen. Implicit application of polynomial filters in a k-step Arnoldi method. *SIAM. J. Matrix Anal. A.*, 13(1):357–385, 1992.

[165] D. C. Sorensen. Deflation techniques for an implicitly restarted Arnoldi algorithm. *SIAM. J. Matrix Anal. A.*, 17(4):789–821, 1996.

[166] P. Spiteri. A new characterization of M-matrices and H-matrices. *BIT*, 43(5):1010–1032, 2003.

[167] M. Spivak. *A Comprehensive Introduction to Differential Geometry, vol. 1.* Publish or Perish, Inc., Berkerley, 1979.

[168] V. Sreeram and P. Agathoklis. Model reduction using balanced realizations with improved low frequency behaviour. *Syst. Control. Lett.*, 12(1):33–38, 1989.

[169] T. Stykel. Gramian based model reduction for descriptor systems. *Math. Control. Signal*, 16:297–319, 2004.

[170] K. Unnenland, P. Van Dooren, and O. Egeland. A novel scheme for positive real balanced truncation. In *Proceedings of the 2007 American Control Conference*, pages 947–952, 2007.

[171] J. M. Varah. A lower bound for the smallest singular value of a matrix. *Linear Algebra Appl.*, 11:3–5, 1975.

[172] K. Veroy, C. Prud'home, and A. T. Patera. Reduced-basis approximation of the viscous Burgers equation: rigorous *a posteriori* error bounds. *C. R. Math.*, 337(9):619–624, 2003.

[173] C. D. Villemagne and R. E. Skelton. Model reduction using a projection formulation. *Int. J. Control*, 46(6):2141–2169, 1987.

[174] S. Volkwein. Model reduction using proper orthogonal decomposition. *Lecture notes at Graz University, available online at* *http://www.math.uni-konstanz.de/numerik/personen/volkwein/teaching/scripts.php*, pages 1–42, 2008.

[175] S. Volkwein and A. Hepberger. Impedance identification by POD model reduction techniques. *Automatisierungstechnik*, 56(8):437–446, 2008.

[176] D. S. Weile, E. Michielssen, E. J. Grimme, and K. Gallivan. A method for generating rational interpolant reduced order models of two-parameter linear systems. *Appl. Math. Lett.*, 12(3):93–102, 1999.

[177] N. Wong and V. Balakrishnan. Fast positive-real balanced truncation via quadratic alternating direction implicit iteration. *IEEE T. Comput. Aid. D.*, 26(9):1725–1731, 2007.

[178] Y. C. Wong. Differential geometry of Grassmann manifolds. In *Proceedings of National Acad. Sci. USA.*, volume 57, pages 589–594, 1967.

[179] Y. Xu and T. Zeng. Optimal $\mathcal{H}_2$ model reduction for large scale MIMO system via tangential interpolation. *Int. J. Numer. Anal. Mod.*, 8(1):174–188, 2007.

[180] B. Yan, S. X. D. Tan, P. Liu, and B. McGaughy. Passive interconnect macromodeling via balanced truncation of linear systems in descriptor form. In *Design Automation Conference, 2007. ASP-DAC '07. Asia and South Pacific*, pages 355–360, 2007.

[181] Z. Zhang, Q. Wang, N. Wong, and L. Daniel. A moment-matching scheme for the passivity-preserving model order reduction of indefinite descriptor systems with possible polynomial parts. In *16th Asia and South Pacific Design Automation Conference (ASP-DAC)*, pages 49–54, 2011.