Chair of Automatic Control
Department of Mechanical Engineering
Technical University of Munich

Optimal model reduction by tangential interpolation: $\mathcal{H}_2$ and $\mathcal{H}_\infty$ perspectives

Alessandro Castagnotto

Vollständiger Abdruck der von der Fakultät für Maschinenwesen der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs

genehmigten Dissertation.

Vorsitzender:	Prof. Wolfgang Polifke, Ph.D.
Prüfer der Dissertation:	1. Prof. Dr.-Ing. habil. Boris Lohmann
	2. Prof. Serkan Gugercin, Ph.D. Virginia Polytechnic Institute and State University Blacksburg (VA), USA

Die Dissertation wurde am 19. April 2018 bei der Technischen Universität München eingereicht und durch die Fakultät für Maschinenwesen am 04. Juni 2018 angenommen.

Acknowledgments

This manuscript is the result of my doctoral research at Technical University of Munich and would not have come to this form without the support of several people I would hereby like to acknowledge.

First and foremost, I would like to thank my supervisor, Prof. Dr.-Ing. habil. Boris Lohmann, for giving me the opportunity to teach and do research in the field of model reduction. The support and freedom he gave me during my time at the Chair of Automatic Control greatly impacted my work and the results of this thesis. Secondly, I would like to thank Prof. Serkan Gugercin for inviting me to visit him and Prof. Christopher Beattie at Virginia Tech. I have greatly enjoyed and benefited from our joint research project and the many discussions on model reduction and numerical linear algebra. To my mentor, Dr. Jens Saak, I owe a big thank you for his friendly support and for treating me as if I was part of his group. Thanks to Serkan and Jens I have had the chance to meet some great people within the model reduction community. Thanks go to my colleagues from MORLab, Dr.-Ing. Heiko Peuscher, Dr.-Ing. Thomas Wolf, for introducing me to the topic of model reduction; and Maria Cruz Varona, with whom I had the privilege to share this journey and develop our "sss" and "sssMOR" toolboxes. Finally, to all colleagues at the Chair of Automatic Control goes my gratitude for their friendship and the exceptional atmosphere that made going to work every day a pleasure. Last but not least, my deepest gratitude goes to my family: to my parents and sister, who have supported me unconditionally over these past thirty years, and to my wife, who is always at my side and is my greatest source of joy and inspiration.

10 August 2018
Alessandro Castagnotto

Contents

1 Introduction and Problem Statement

With the advent of computerized modeling and simulation-based development, the number of applications in which numerical models are used to create virtual prototypes of a given system of interest has vastly increased, ranging from structural mechanics, electro-magnetics, to fluid dynamics, just to name a few. Numerical simulations not only reduce the cost of building physical prototypes and allow early testing and validation, but also offer an opportunity of using mathematical tools to optimize the design according to given criteria.

Depending on the application at hand, the complexity of numerical models of real-life technical systems can easily become prohibitive, making their numerical evaluation challenging even for modern high-performance computers. This is even more true if, during optimizations, the models need to be evaluated repeatedly until an optimal solution is found. Furthermore, in the era of digitalization, Internet of Things (IoT), digital twins and embedded controllers, there is an increased demand of running numerical models in real-time, often with very limited computational resources, in order, e.g., to estimate the state of the system and perform predictive maintenance, or run feedback controllers to achieve a desired performance. In these latter applications, it is of particular importance that numerical models do not exceed a given admissible complexity, while being the best possible representations of the system of interest.

This thesis is dedicated to finding the best possible approximations for large-scale models of multiple-inputs, multiple-outputs (MIMO) linear time-invariant dynamical systems, given an admissible complexity, represented by the reduced model order. New algorithms are presented to produce reduced-order models that aim at minimizing the approximation error based on two of the most prominent system norms, namely the $\mathcal{H}_2$ and $\mathcal{H}_\infty$ norms. For the former, a new framework by the name of "Model Function" is introduced to decouple the cost of optimization from the cost of reduction, hence generating substantial speedup in $\mathcal{H}_2$-optimal reduction. Based on this result, an efficient implementation of a globalized $\mathcal{H}_2$-optimal model reduction algorithm is presented. For the latter, a rational-interpolation-based scheme available for single-input, single-

output (SISO) systems is extended to the case of MIMO systems. Numerical efficiency is achieved by using data-driven surrogate modeling approaches. For all presented algorithms, fully-documented, open-source numerical implementations are available within the MATLAB[1] Toolbox `sssMOR`, released during this thesis and developed with the support of a number of people. The goal of this toolbox is to make model reduction algorithms available to non-expert users by minimizing the number of required input parameters and delivering reduced-order models at the push of a button. At the same time, all model reduction functions in `sssMOR` allow easy customization for expert users who wish to prescribe a desired behavior and use the toolbox as a reference to compare model reduction algorithms.

This thesis is structured as follows: Chapter 2 gives a brief overview of the fundamentals on model reduction pertinent to this thesis, while Chapter 3 can be considered as the main part, addressing the topics of $\mathcal{H}_2$- and $\mathcal{H}_\infty$-optimal model reduction by tangential interpolation and presenting the main contributions of this thesis. Each topic starts with a brief review of existing results, followed by a short summary of advances achieved during this thesis. The details are available in the original publications reprinted with the publishers' consent or referenced in the Appendix. Section presenting new results achieved during this thesis have been distinguished by a "*" symbol for ease of reference. Chapter 4 introduces numerical tools developed and released during this thesis, while Chapter 5 ends the discussion with a summary of achievements and outlook.

[1] MATLAB and Control System Toolbox 2016b, The MathWorks, Inc., Natick, Massachusetts, United States.

2 A Brief Review on Model Reduction

In this thesis, we consider linear time-invariant (LTI) systems described by generalized state-space models of the form

$$\Sigma \quad \begin{cases} E\,\dot{x}(t) = A\,x(t) + B\,u(t), & \text{(2.1a)} \\ y(t) = C\,x(t) + D\,u(t), & \text{(2.1b)} \end{cases}$$

where $E \in \mathbb{R}^{N\times N}$ is the *descriptor* matrix, $A \in \mathbb{R}^{N\times N}$ is the system matrix and $x \in \mathbb{R}^N$, $u \in \mathbb{R}^m$, $y \in \mathbb{R}^p$ represent the state, input and output vectors of the system, respectively. The matrix $D \in \mathbb{R}^{p\times m}$ directly linking the input vector to the output vector is called the *feed-through* matrix.

Assumptions and restrictions Throughout this thesis, we will consider only models with regular descriptor matrix, satisfying $\det E \neq 0$. Extensions to the case of singular E can be found in [29, 68, 82, 123, 127] and references therein. In the following, for simplicity, we will assume $p, m \ll N$ to hold, meaning that the vectors y (the quantities of interest) and u (external forces) have only few entries, compared to the number of state variables in x. This is not a necessary restriction for applying the method of tangential interpolation (cf. Section 2.2), used within this thesis, per se. Nonetheless, models with a large number of inputs and outputs may be more tedious to approximate and may require additional considerations, cf. [12, 25, 27, 51]. Finally, the discussion will be limited to asymptotically stable systems, for which all generalized eigenvalues of the pencil (E, A) are in the open left-half of the complex plane. This restriction is required to ensure that the norms introduced in Section 3.1 are defined. Whenever unstable systems of the form (2.1) are given, the *anti-stable* part should be retained in the reduced model. This can be achieved by decomposing (2.1) into its asymptotically stable and anti-stable part; the former can then be reduced while the latter is preserved exactly, cf. [19, 23, 92, 121].

Frequency domain description A different—yet equivalent—characterization of the dynamical behavior of a linear system (2.1) from the inputs u to the outputs y can be given in the frequency domain as a relation $y(s) = G(s)\,u(s)$, where $G(s)$ represents the rational transfer function matrix

$$G(s) := C\,(sE - A)^{-1}\,B + D, \tag{2.2}$$

in the variable $s \in \mathbb{C}$, obtained through Laplace transform of (2.1) under the assumption $x(t = 0) = 0$ [4, 78].

Motivation for LTI models Models of the form (2.1) and (2.2) are widely used to describe the dynamical behavior of linear systems, as well as non-linear systems locally around the equilibria and operating points. As such, the theory on this system class is well established and a variety of analysis, control design [78, 80, 86, 125] and model reduction techniques [4, 5] have been developed over the past half century. In addition, most computations can be performed very efficiently exploiting advanced numerical linear algebra routines [17, 39, 41, 115]. Nonetheless, as the problem size N increases, numerical computations become resource demanding, quickly reaching a point where a simple model evaluation becomes challenging—if at all feasible—on a standard computer.

Reduced-order models For this reason, *model order reduction* techniques are being developed to obtain simplified representations, called reduced-order models of the form

$$\Sigma_r \quad \begin{cases} E_r\,\dot{x}_r(t) = A_r\,x_r(t) + B_r\,u(t), & \text{(2.3a)} \\ y_r(t) = C_r\,x_r(t) + D_r\,u(t), & \text{(2.3b)} \end{cases}$$

with a reduced state vector $x_r(t) \in \mathbb{R}^n$ and $n \ll N$. The respective transfer function matrix takes the form

$$G_r(s) := C_r\,(sE_r - A_r)^{-1}\,B_r + D_r. \tag{2.4}$$

The goal of model reduction within this thesis is to construct reduced-order models (2.3), respectively (2.4), that yield a good approximation $y_r(t) \approx y(t)$ for a wide range of admissible inputs $u(t)$. Several approaches have been studied in the literature to find valid approximations of dynamical systems of the form (2.1) or (2.2). Early approaches include approximations of (2.2) by *rational interpolation*, most notably by means of

Padé approximations [30, 102, 119], moment matching or asymptotic waveform evaluation [109], as well as Kalman's work on *realization theory* [80] and *partial realization* [62, 79], targeted at finding minimal state-space realizations for given proper rational matrix functions or sequences of their evaluations at $s \to \infty$, respectively. Analogously, minimal state-space realizations solving the rational interpolation problem can also be obtained by means of the Loewner matrix approach [3, 15, 85, 95]. More recently, as computer-aided modeling and spatial discretizations of partial differential equations generally result in models of the form (2.1), a common approach to generate reduced-order models is given by *projecting* the dynamic equations (2.1) onto appropriately chosen subspaces [132]. This projective formulation is quite general in that it can be used to express many of the most well-known approaches to generate reduced-order models for (2.1), some of which are presented in the next section. Note however that there exist alternative approaches to generate reduced-order models given (2.1), which cannot be formulated in a projective framework. An example is given by reduced-order model parametrizations based on Sylvester equations [7, 8, 76, 134, 136]. Finally, note that recently *data-driven* approaches which generate reduced-order models from data (e.g. measurements or model evaluations), i.e. without requiring explicit knowledge of (2.1) or (2.2), have gained increased interest. This class of methods includes the aforementioned Loewner matrix approach, its time-domain variant [106], as well as Vector Fitting [42, 43, 70] and further approaches, e.g., the recent approach [81], available in the literature on *system identification*, which are not listed for brevity.

2.1 Model Reduction by Projection

The foundation in reducing state-space models (2.1) by projection is given by assuming that the state vector $x(t) \in \mathbb{R}^N$ can be approximated in an n-dimensional subspace, defined by a basis matrix $V \in \mathbb{R}^{N \times n}$, according to the relation

$$x(t) = V\, x_r(t) + e(t). \tag{2.5}$$

Note that our approximation goal is not to minimize the state *reconstruction* error $\|x(t) - V x_r(t)\|$ but rather to determine an appropriate subspace $\mathcal{R}(V)$ such that the output error $\|y(t) - y_r(t)\|$ is small for all admissible inputs $u(t)$. Through a change of

variables in (2.1a), the set of first-order differential equations takes the form

$$E\,V\,\dot{x}_r(t) \;=\; A\,V\,x_r(t) \;+ B\,u(t) + \underbrace{\left(A\,e(t) - E\dot{e}(t)\right)}_{\varepsilon(t)}. \tag{2.6}$$

To ensure solvability of the overdetermined set of differential equations (2.6) and eliminate the residual term $\varepsilon(t)$, the equations are *projected* onto the subspace $\mathcal{R}(EV)$, orthogonally to a second n-dimensional subspace $\mathcal{R}(W)$. This can be achieved by pre-multiplying (2.6) with the projector[1] $\Pi = EV\left(W^\top EV\right)^{-1} W^\top$ satisfying $\Pi^2 = \Pi$. Assuming $\varepsilon(t) \perp \mathcal{R}(W)$ (known as the *Petrov-Galerkin condition*), the projected dynamics in the reduced state vector take the form

$$\Pi E\,V\,\dot{x}_r(t) \;=\; \Pi\,A\,V\,x_r(t) \;+ \Pi B\,u(t). \tag{2.7}$$

Due to regularity of E, the matrix $EV\left(W^\top EV\right)^{-1}$ is of full column-rank, hence the reduced-order model ultimately takes the form

$$W^\top E\,V\,\dot{x}_r(t) \;=\; W^\top A\,V\,x_r(t) \;+\; W^\top B\,u(t), \tag{2.8a}$$

$$y_r(t) \;=\; C\,V\,x_r(t) \;+\; D_r\,u(t), \tag{2.8b}$$

where in (2.8b) we have inserted the ansatz (2.5) and neglected all terms that depend on the state reconstruction error $e(t)$. Note that the feed-through term D_r is not affected by this projection, therefore it appears natural to select $D_r = D$, which often in the literature leads to neglecting this term altogether. However, we will use the term D_r in Section 3.3 explicitly as additional degree of freedom to increase the approximation quality and shall therefore include it in our considerations.

From (2.8) it becomes evident that to obtain a reduced-order model by projection, only appropriate choices of bases matrices (also referred to as *projection* matrices) V, W and—whenever applicable—a reduced feed-through D_r are needed. Note that, in general, only the subspaces $\mathcal{R}(V)$ and $\mathcal{R}(W)$ are of interest, as the reduced transfer function is invariant to a change of bases [16, 118]. Nevertheless, specific choices of bases, e.g. orthonormal or bi-orthogonal, are often used due to numerical considerations. In the following we will briefly introduce some of the most prominent methods to design V and W, before we discuss in more detail in Section 2.2 the reduction method of central relevance to this thesis.

[1] A similar derivation can be conducted using pseudo-inverses, cf. [112].

Modal truncation As eigenvalues and eigenvectors are widely used to characterize the behavior of dynamical systems over time, *modal truncation* [40, 93] is a model reduction approach aimed at preserving certain eigenvalues of interest while neglecting others. It can be interpreted as a projection onto the invariant subspace spanned by the preserved eigenvectors. This approach—and variants thereof—is widely used in structural mechanics [11, 38, 124], where the reconstruction of the displacement $x(t) \approx V\, x_r(t)$ over the whole domain is often of interest and for which lower-frequent eigenvalues often play a dominant role. The question about eigenvalue dominance becomes more involved when approximating input/output maps such as (2.1), as aspects of controllability and observability [78] need to be taken into account. For this reason, several dominance measures have been proposed [90, 94, 113, 130]. Modal truncation has the advantage of explicitly preserving eigenvalues and eigenvectors of the original system, which might be important depending on the application at hand. However, note that for large-scale models, only an estimation of particular regions of the spectrum is possible in a numerical efficient manner, using iterative algorithms such as power methods, inverse iteration or QR-iteration [41], as well as the implicitly restarted Arnoldi [89]. In addition, modal truncation only allows boolean decisions between keeping or neglecting certain eigenvectors in the reduced basis V. The methods that follow offer somewhat higher flexibility by introducing new state-space directions that might be better suited to approximate the input/output behavior of the system.

Balanced truncation A second class of projection methods is known under the name of *balanced truncation* or *truncated balanced realization*. It is based on finding a *balancing* state-space transformation in which individual state-space directions are equally controllable and observable. This can be characterized in terms of the Gramian matrices for controllability and observability, P and Q respectively, defined in terms of the solutions of following generalized Lyapunov equations

$$A\,P\,E^\top + E\,P\,A^\top + BB^\top = 0, \tag{2.9a}$$

$$A^\top\,\tilde{Q}\,E + E^\top\,\tilde{Q}\,A + C^\top C = 0, \tag{2.9b}$$

where $Q = E^\top \tilde{Q} E$. System-invariant measures, called *Hankel singular values*, for the degree of controllability and observability of balanced state-space directions can be defined as $\varsigma_i := \sqrt{\lambda_i\,(Q\,P)}$, $i = 1, \ldots, N$. Model reduction is then performed by inspecting the relative decay of ς_i and truncating directions corresponding to lower values. Preliminary work is due to Mullis and Roberts [100] and Moore [97, 98]. Several extensions

and derivations have followed. The most relevant in the context of this thesis is given by the *Low-Rank Square-Root Balanced Truncation* (LR-SRBT) method [22, 28, 44, 45, 83, 88, 107, 108, 116, 120, 137], which is based on finding low-rank approximations of the controllability and observability Gramians, e.g. $P \approx Z_p Z_p^\top$ with $Z_p \in \mathbb{R}^{N\times q}$ and $N \gg q > n$. The original method by Moore bears the advantage of guaranteeing preservation of stability and having rigorous, a priori error bounds [49]

$$\|G - G_r\|_{\mathcal{H}_\infty} \leq 2 \sum_{i=n+1}^{N} \varsigma_i, \tag{2.10}$$

where the $\mathcal{H}_\infty$ norm is introduced in Section 3.1. This allows both the estimation of the error and a suitable selection of reduced order n, making this method—in principle—suitable for fully automated model reduction. Unfortunately, rigor is lost when applying LR-SRBT, which is however the only variant applicable to truly large-scale models. In addition, the low-rank solution of the Gramian matrices introduces new parameters, for which often times heuristics are used and an optimal choice is still open [83, 137]. Even so, this model reduction class has been used very effectively and is still considered the gold standard in model reduction. Note that there is a strong link between approximate balanced truncation and the rational interpolation method discussed in Section 2.2 [135].

Optimal Hankel norm approximation Related to the method of balanced truncation is the construction of an optimal reduced model measured in terms of the Hankel norm, which is the 2-induced norm of the *Hankel operator*, a modification of the *convolution operator*, mapping past inputs to future outputs [4]. The main result is due to Adamjan, Arvo and Krein [1] and Glover [61]. This method requires balancing and all-pass dilation of the full-order model and subsequent projection onto the stable eigenspace, which in general involves dense operations. As such, the method is non viable for problems of very large-scale [24]. Nonetheless, stability is preserved and the error bound

$$\|G - G_r\|_{\mathcal{H}_\infty} \leq \sum_{i=n+1}^{N} \varsigma_i \tag{2.11}$$

holds. Even though this type of approximation is not optimal in the $\mathcal{H}_\infty$ norm (the 2-induced norm of the convolution operator, cf. Section 3.1), it is known to yield good approximations also in this norm. For this reason, this method will be used in Section 3.3 for comparison.

2.2 Model Reduction by Tangential Interpolation

Another approach to the model reduction problem is given by rational interpolation, i.e. constructing a reduced-order model, whose (rational) transfer function *interpolates* the original transfer function—and perhaps its derivatives—at selected complex *shifts* $s=\sigma_i$, i.e.

$$G^{(j)}(\sigma_i) = G_r^{(j)}(\sigma_i) \quad i = 1, 2, \ldots, \; j = 0, 1, \ldots, \tag{2.12}$$

where $G^{(j)}(s)$ denotes the j-th derivative of $G(s)$ with respect to s. As the coefficients of the Taylor series expansion of $G(s)$ around a frequency σ are also called *moments*[2], this method is often also referred to as *moment matching*.

Given a rational transfer function $G(s)$, the problem of finding a suitable interpolant has been addressed in different fields, e.g. complex analysis and system theory, under the names of (multi-point) Padé Approximation [10], Moment Matching [84, 142], Asymptotic Waveform Evaluation [109], just to name a few. However, the construction of a reduced transfer function $G_r(s)$ based on the explicit knowledge of the moments is known to be ill-conditioned [52]. In the 1980's, DeVillemagne and Skelton [132] showed how to achieve rational interpolation by projection, i.e. without explicit computation of the moments. In the late 1990's, Grimme [64] used the rational Krylov method by Ruhe [114], a generalization of the shifted and inverted *Arnoldi* algorithm, to compute the projection matrices in a numerically stable way. Similarly, Feldmann and Freund [52] and Gallivan, Grimme and Van Dooren [59] used a *Lanczos* process to compute projection matrices. As to the nomenclature, this class of reduction methods is often denoted in the literature with different names, i.e. *rational interpolation*, *moment matching* or *Krylov subspace methods*. While the main goal of the approaches behind these names is common, i.e. interpolating the transfer function and its derivatives at selected points in the complex plane, there are subtle differences that make the concepts not entirely equivalent, especially for MIMO models. We refer to [16] for a brief discussion on this topic and will use in the remainder of this contribution the most general name, i.e. rational interpolation.

For MIMO systems, an interpolatory approach as in (2.12) might be inefficient, as single entries of the transfer function matrix may have different frequency ranges of interest. For this reason, Gallivan, Vandendorpe and Vandoren [60] proposed approximations by *tangential* interpolation, i.e. interpolating the transfer function matrix only

[2]Note that the literature may differ here in the sign of these coefficients.

along selected *left* and *right tangential directions* $l \in \mathbb{C}^p$, $r \in \mathbb{C}^m$, i.e.

$$G^{(j)}(\sigma_i)\, r = G_r^{(j)}(\sigma_i)\, r, \tag{2.13a}$$

$$l^\top G^{(j)}(\sigma_i) = l^\top G_r^{(j)}(\sigma_i). \tag{2.13b}$$

As this result is the main tool used in the reduction methods presented in the following chapter, it is directly introduced here, omitting all preliminary results briefly mentioned above.

Theorem 1 (Bi-tangential Hermite Interpolation [16, 60])**.** *Consider a full-order model* Σ *as in (2.1) with transfer function* $G(s)$ *and let scalar frequencies* $\sigma_i \in \mathbb{C}$ *and vectors* $r_i \in \mathbb{C}^m$, $l_j \in \mathbb{C}^p$ *be given such that* $\sigma_i E - A$ *is nonsingular for* $i = 1, \ldots, n$. *Consider projection matrices* $V, W \in \mathbb{C}^{N \times n}$ *and let the resulting reduced-order model (2.8) have the transfer function* $G_r(s)$ *and eigenvalues* λ_j *satisfying* $\lambda_j \neq \sigma_i$ *for all* $j = 1, \ldots, n$ *and* $i = 1, \ldots, n$.

1. *If*

$$(A - \sigma_i E)^{-1} B\, r_i \in \mathcal{R}(V), \qquad i = 1, \ldots, n \tag{2.14}$$

then $G(\sigma_i)\, r_i = G_r(\sigma_i)\, r_i$.

2. *If*

$$(A - \sigma_i E)^{-\mathrm{T}} C^\top l_i \in \mathcal{R}(W), \qquad i = 1, \ldots, n \tag{2.15}$$

then $l_i^\top G(\sigma_i) = l_i^\top G_r(\sigma_i)$.

3. *If both (2.14) and (2.15) hold, then, in addition,*

$$l_i^\top G^{(j)}(\sigma_i)\, r_i = l_i^\top G^{(j)}(\sigma_i)\, r_i, \qquad i = 1, \ldots, n. \tag{2.16}$$

This class of methods is inherently predestined for the reduction of large-scale models, as its computation only requires the solution of (generally *sparse*) linear systems of equations (LSE) of the form (2.14) and (2.15), which can be done efficiently even in the large-scale setting. Note that to improve numerical stability, the projection matrices as of (2.14) and (2.15) should be preferably computed using an Arnoldi-type approach, including (modified) Gram-Schmidt orthogonalization after each new direction is computed.

In contrast to balanced truncation and optimal Hankel norm approximation, rigorous and global error bounds are available only for systems in *strictly dissipative* form, cf.

[104]. Note that error expressions have also been provided, e.g., by Bai et al. in [9] and Gugercin in [65]. More recently, Feng, Antoulas and Benner [53] have presented point-wise error bounds in the frequency (and parameter) domain. As already denoted by Antoulas and Soerensen in [6], the *local* nature of interpolatory reduction is at the same time a fundamental problem and a fundamental strength of the methods[3]. Note for completeness that, given a state-space model (2.1), the problem of rational interpolation can be addressed also in a non-projective way by Sylvester-equations-based parametrizations (cp. [7, 8, 76]) or the Loewner matrix approach referenced above.

Once numerically reliable tools are available to produce reduced-order models satisfying (tangential) interpolatory conditions, the question naturally arises as to how to select interpolation points, tangential directions and the number of derivatives to be matched in order to achieve a good approximation. This leads to the question of which selection of parameters results in an *optimal* reduced-order model with respect to some given criteria. The next chapter—and in fact the main part of this thesis—is devoted to this question. After a brief introduction to the approaches already available in literature, the contributions developed within this thesis are presented.

[3]We will come back to this point in Section 3.2.2

3 Optimal Model Reduction by Tangential Interpolation

Whenever a full-order model is given and the requirements on the allowable complexity, expressed in terms of a target reduced order, are known, the question naturally arises as to how to generate a reduced model that is optimal—in some sense—for this given reduced order. In this chapter, we introduce the two most prominent system norms used to quantify the approximation error. The remainder of the chapter will then be devoted to develop interpolatory reduction methods targeted at generating reduced models with the lowest possible error, while being computationally efficient also for very large-scale problems.

3.1 System Norms

In the context of generating approximate models, a very fundamental question is to determine how large the approximation error is allowed to be and subsequently quantify how large the resulting approximation error really is. For dynamical systems (2.1) reduced by projection (2.8), typical characterizations are given in terms of the output error $e_y(t) := y(t) - y_r(t)$ or the error in state reconstruction $e_x(t) := x(t) - Vx_r(t)$ for all admissible inputs $u(t) \in \mathcal{U}$ in some set $\mathcal{U}$. The work in this thesis focuses on *control systems*, where the modeling intent is to represent the input/output dynamics of a given system and the evolution of state variables can be seen as a necessary intermediate step. In this context, a correct approximation of the state vector is at most a subordinate goal, whereas the primary goal is to minimize the output error $e_y(t)$. In the frequency domain, this error can be represented as

$$\mathscr{L}\{y(t) - y_r(t)\} = y(s) - y_r(s) = \underbrace{(G(s) - G_r(s))}_{G_e(s)} u(s), \tag{3.1}$$

where $\mathscr{L}\{\cdot\}$ denotes the Laplace transform for zero initial conditions. As it can be seen from (3.1), the output error for all admissible inputs can be characterized by the

difference in transfer function matrices $G_e(s) = G(s) - G_r(s)$, therefore minimizing the output error $e_y(t)$ can be translated to minimizing the approximation error $G_e(s)$. The distance between two transfer function matrices can be measured in terms of *Hardy* norms ($\mathcal{H}_p$), where $\mathcal{H}_p^{p\times m}$ spaces are spaces of $p \times m$ complex-valued function which are analytic in the open right-half of the complex plane. See [4] for a more detailed discussion on normed spaces.

$\mathcal{H}_2$ norm Consider a $p\times m$ complex-valued function G, analytic in the open right-half of the complex plane. Then its $\mathcal{H}_2$ norm is defined as [4, 14]

$$\|G\|_{\mathcal{H}_2} := \left(\frac{1}{2\pi} \int_{-\infty}^{\infty} \operatorname{tr}\left(G^H(-j\omega) G(j\omega) \right) \mathrm{d}\omega \right)^{1/2}. \tag{3.2}$$

We note that $\mathcal{H}_2^{p\times m}$ is a Hilbert space as the norm (3.2) can be defined as $\|G\|_{\mathcal{H}_2} = \sqrt{\langle G,\, G\rangle_{\mathcal{H}_2}}$ by means of the inner product

$$\langle G,\, H\rangle_{\mathcal{H}_2} := \frac{1}{2\pi} \int_{-\infty}^{\infty} \operatorname{tr}\left(G^H(-j\omega) H(j\omega) \right) \mathrm{d}\omega. \tag{3.3}$$

In addition, given the controllability and observability Gramians P and Q, respectively, defined in (2.9), the $\mathcal{H}_2$ norm can be computed by [4, 14]

$$\|G\|_{\mathcal{H}_2} = \sqrt{\operatorname{tr}\left(B^\top Q B\right)} = \sqrt{\operatorname{tr}\left(C P C^\top\right)}. \tag{3.4}$$

It was shown in [65, 69] that the $\mathcal{H}_2$ norm can be evaluated also using the residue theorem according to following result (for SISO models)

$$\|G\|_{\mathcal{H}_2} = \sqrt{\sum_{i=1}^{N} \operatorname{res}\left[G(s), \lambda_i\right] G(-\lambda_i)} \tag{3.5}$$

where we assumed the poles of $G(s)$ to be simple for ease of notation. Finally note that for systems with single-input ($m = 1$) and/or single-output ($p = 1$), the $\mathcal{H}_2$ norm can be interpreted as an induced norm of the underlying convolution operator according to [16]

$$\|G\|_{\mathcal{H}_2} = \sup_{u\in\mathcal{L}_2} \frac{\|y\|_{\mathcal{L}_\infty}}{\|u\|_{\mathcal{L}_2}}. \tag{3.6}$$

There exists a direct relation between the approximation error in the frequency domain in terms of the $\mathcal{H}_2$ norm and a bound for the Lebesgue $\mathcal{L}_\infty$ norm of the output error in the time domain, also for the more general case of MIMO models, according to [67]

$$\|y - y_r\|_{\mathcal{L}_\infty} \leq \|G - G_r\|_{\mathcal{H}_2} \|u\|_{\mathcal{L}_2}. \tag{3.7}$$

A low $\mathcal{H}_2$ error in the frequency domain hence corresponds to a low maximum output error in the time domain for all $\mathcal{L}_2$-bounded inputs.

$\mathcal{H}_\infty$ norm Consider a $p \times m$ complex-valued function G, analytic in the open right-half of the complex plane. Then its $\mathcal{H}_\infty$ norm is defined as [4]

$$\|G\|_{\mathcal{H}_\infty} := \sup_{\omega \in \mathbb{R}} \varsigma_{max} \left(G(j\omega) \right), \tag{3.8}$$

where $\varsigma_{max}(\cdot)$ denotes the largest singular value. As opposed to $\mathcal{H}_2^{p\times m}$, the Hardy space $\mathcal{H}_\infty^{p\times m}$ is not an Hilbert space. Nonetheless, it can be shown that the $\mathcal{H}_\infty$ norm is an induced norm according to [4, 16]

$$\|G\|_{\mathcal{H}_\infty} = \sup_{U \neq 0} \frac{\|GU\|_{\mathcal{H}_2}}{\|U\|_{\mathcal{H}_2}} = \sup_{u \in \mathcal{L}_2} \frac{\|y\|_{\mathcal{L}_2}}{\|u\|_{\mathcal{L}_2}}, \tag{3.9}$$

where $U := \mathscr{L}\{u\}$ and u is an $\mathcal{L}_2$ function over the positive real axis [4]. A bounded $\mathcal{H}_\infty$ error in the frequency domain results in an uniformly bounded output error in the time domain, according to

$$\|y - y_r\|_{\mathcal{L}_2} \leq \|G - G_r\|_{\mathcal{H}_\infty} \|u\|_{\mathcal{L}_2}. \tag{3.10}$$

3.2 $\mathcal{H}_2$-Optimal Model Reduction

Given a desired fixed order n, the goal of this section is to find a reduced-order model satisfying

$$G_r(s) = \arg \min_{\deg \widehat{G}_r = n} \|G - \widehat{G}_r\|_{\mathcal{H}_2}. \tag{3.11}$$

As the optimization problem in (3.11) is non-convex, at first the ambition will be to find a local solution. This problem has been studied time and again over the course of the past 50 years, cf. e.g. [13, 31, 58, 66, 67, 75, 87, 96, 122, 128, 133, 139, 140, 141], and several characterizations of optimality solutions have been derived. A good

overview can be found in the recent manuscript by Beattie and Gugercin [16]. Considering the problem of minimizing the mean-squared output error for a given stochastic input, Meier and Luenberger derived in [96] first-order optimality conditions for SISO models in terms of rational interpolatory conditions. Further, optimality conditions in terms of Lyapunov equations were derived by Wilson [133] and later by Hyland and Bernstein [75]. Gugercin, Antoulas and Beattie [67] linked all of the above, showing their equivalence to structured orthogonality conditions, based on optimal approximations in Hilbert spaces. The generalization of the interpolatory conditions to MIMO models presented in the following is based on the works [31, 67, 128].

Theorem 2 ([31, 67, 128])**.** *Consider a full-order model (2.1) with transfer function $G(s)$. Consider a reduced-order model with transfer function $G_r(s) = \sum_{i=1}^{n} \frac{\hat{c}_i \hat{b}_i}{s-\lambda_{r,i}}$ with reduced poles $\lambda_{r,i} \in \mathbb{C}$ and input, output residue directions $\hat{b}_i^\top \in \mathbb{C}^m$, $\hat{c}_i \in \mathbb{C}^p$, respectively.*

If $G_r(s)$ satisfies (3.11) locally, then

$$G(-\bar{\lambda}_{r,i})\,\hat{b}_i^\top = G_r(-\bar{\lambda}_{r,i})\,\hat{b}_i^\top \tag{3.12a}$$

$$\hat{c}_i^\top\, G(-\bar{\lambda}_{r,i}) = \hat{c}_i^\top\, G_r(-\bar{\lambda}_{r,i}) \tag{3.12b}$$

$$\hat{c}_i^\top\, G^{(1)}(-\bar{\lambda}_{r,i})\,\hat{b}_i^\top = \hat{c}_i^\top\, G_r^{(1)}(-\bar{\lambda}_{r,i})\,\hat{b}_i^\top \tag{3.12c}$$

for $i = 1, \ldots, n$.

The extension to the case of poles with higher multiplicities is omitted here for brevity and can be found in [129]. In general, previous approaches to the $\mathcal{H}_2$-optimal model reduction problem referenced above either required the solution of sequences of Lyapunov equations or the computation of transfer functions numerator and denominator polynomials, hence becoming intractable for large-scale problems,. Gugercin, Antoulas, and Beattie introduced in [67] an algorithm to iteratively construct a reduced-order model by rational interpolation—hence being numerically tractable also in the large-scale setting—until the optimality conditions (3.12) are satisfied.

The Iterative Rational Krylov Algorithm and its derivates If the reduced order poles and residue directions were known a priori, then by Theorem 1 it would be clear how to construct a reduced-order model satisfying the optimality conditions (3.12). However, as reduced poles and residue directions depend on the interpolation data (shifts and tangential directions), Gugercin, Antoulas and Beattie proposed in [67] the Iterative Rational Krylov Algorithm (IRKA) to iteratively adapt the interpolation data, until (hopefully) convergence. A sketch of the MIMO version of IRKA is given in

Algorithm 1 and will be a central element of all further discussions within this thesis.

Algorithm 1 MIMO $\mathcal{H}_2$-Optimal Tangential Interpolation (IRKA)

Input: Σ; Initial interpolation data $\{\sigma_i\}_{i=1}^n$, $\{r_i\}_{i=1}^n$, $\{l_i\}_{i=1}^n$
Output: locally $\mathcal{H}_2$-optimal reduced model Σ_r

1: **while** not converged **do**
2: Compute V, W according to (2.14) and (2.15)
3: $\Sigma_r \leftarrow W^\top \Sigma V$ // compute reduced model by projection
4: $[X, D, Y] = \text{eig}(\Sigma_r)$ // eigendecomposition
5: $\sigma_i \leftarrow -\overline{\lambda}_i(D)$; $r_i \leftarrow B_r^\top Y e_i$; $l_i \leftarrow C_r X e_i$ // update interpolation data
6: **end while**

IRKA has experienced a great success mainly due to its simplicity and effectiveness in producing high-quality reduced-order models for very large-scale problems. Even though no results are available in terms of convergence and stability of the reduced-order models in general (cf. [56] for the case of state-space symmetric systems), it has been shown to be effective in practice. It has therefore been extended in different contexts, e.g. in the data-driven approximation within the Loewner framework [15], structure preserving reduction of port-Hamiltonian systems, systems in 2nd order form [138], general co-prime factorizations [14], DAEs [68], and recently even nonlinear systems [2, 18, 20, 21, 55]. In addition, in [15] Beattie and Gugercin introduced a residue-correction step to accelerate the convergence in presence of large input/output spaces.

For completeness, note that also different approaches have been presented to efficiently solve the optimal $\mathcal{H}_2$ reduction problem. For instance, as opposed to the fixed-point iterations mentioned so far, trust-region descent algorithms for $\mathcal{H}_2$-optimal reduction have been derived by Beattie and Gugercin [15] and Panzer et al. in [103, 105]. A detailed description of these algorithms is left out for brevity. However, note that many of the approaches presented in this thesis also apply to these alternatives.

3.2.1 Initialization*

As for any local optimization problem, *initialization* plays a crucial role in determining convergence behavior and ultimately the quality of the optimal solution. The same holds true also for the case of $\mathcal{H}_2$-optimal model reduction as presented in the previous section[1]. This is briefly discussed in [65, 66, 67, 103] and more extensively in the student thesis by Michael Ott [101] and Siyang Hu [74]—who I had the privilege to supervise—as well as the manuscript [33]. Indeed, not only the convergence speed (and hence the

[1] We will show in Section 3.2.3 how the dependency on appropriate initialization can be mitigated by introducing *globalized* $\mathcal{H}_2$-optimal reduction strategies.

reduction cost), but also stability and approximation error[2] highly depend on the initial choice of $\{\sigma_i\}_{i=1}^n$, $\{r_i\}_{i=1}^n$, $\{l_i\}_{i=1}^n$. Even though there have been different studies in the literature trying to determine *a priori*, i.e. before any sort of reduction, valid choices of initial parameters, to date there is no initialization strategy that is guaranteed to yield good results in all cases.

The problem of identifying good shifts σ_i, be it for direct reduction by rational interpolation or as initialization for an $\mathcal{H}_2$-optimal reduction method, has been studied by different researchers. Grimme [64] studied the problem of choosing appropriate shifts and derived heuristics relations when choosing interpolation points on the real or imaginary axis. Eid [46, 47] used relationships resulting from time-domain model reduction to propose one optimal interpolation point, which can be computed using the algorithm ICOP, available in the `sssMOR` toolbox presented in Chapter 4. Gugercin, Antoulas and Beattie propose in [65, 67, 69] to choose mirror images of the most dominant eigenvalues of the full-order model, inspired by the expression of the $\mathcal{H}_2$ error. Also note that the problem of finding good frequencies has been dealt also by Penzl [108], Benner, Kürschner, and Saak [28, 83], as well as Druskin, Simoncini, and Zavslavski [44, 45] in the context of approximate solution of Lyapunov equations. As demonstrated in [135], the strong relationship between these methods and reduction by rational interpolation make these approached interesting also to the problem at hand.

In [101], Ott has compared a large number of shift initialization strategies for SISO models and assessed their suitability for initializing $\mathcal{H}_2$-optimal reduction algorithms. Noting that the amount of numerical simulations was not enough to have statistical relevance, during this work no affirmative answer to the original question could be found. Nonetheless, the initialization strategies that appear to bear the highest potential are:

I.1 Computing a few eigenvalues/eigenvector of the original model using iterative methods (cp. `eigs` in MATLAB) and using the mirrored spectrum, as well as input/output residual directions as initial data.

I.2 Initialize all data at $\sigma_i = 0$ (and, for MIMO models, $r_i = \begin{bmatrix}1 & 1 & \dots & 1\end{bmatrix}^\top$, $l_i = 1\begin{bmatrix}1 & 1 & \dots & 1\end{bmatrix}^\top$ of appropriate dimensions).

The first approach (I.1) is certainly the most valid, as it uses information from the original model and is motivated by $\mathcal{H}_2$ considerations (cf. [69]). In practice, this approach generally yields quick convergence and good reduced-order models, even though

[2] It turns out that the selection of *reduced order* may also have a significant impact on the optimization. In fact, a variation of ± 1 in n can make a difference, e.g., in terms of convergence and stability. An appropriate selection of reduced order lies outside the scope of this thesis and is topic of ongoing research. Cf., e.g., the CUmulative REduction (CURE) scheme in [103, 105].

exceptions are known. In general, it is numerically inexpensive, as the computation of a few eigenvalues can be done efficiently by means of power iterations. Nonetheless, its efficiency may decrease with increasing number of eigenvalues to be computed. In addition, it may sometimes suffer from convergence issues, especially when computing eigenvalues with largest/smallest real or imaginary part. In general, one is interested in determining the first n most dominant eigenvalues, which could be done e.g. by applying the dominant pole algorithm of [113]. The second approach (I.2) is more trivial in that it requires no knowledge of the original model. At the same time, this is as well one of its advantages, requiring no additional computational effort. In addition, an initial reduction with this choice of parameters is particularly convenient from a computational perspective, as only LSEs with same left hand-sides need to be solved. The choice of tangential directions as vectors of 1s makes sure that all input/output channels are considered in the reduction. Both initialization strategies are implemented in the `sssMOR` toolbox presented in Chapter 4 within the function `initializeShifts`.

3.2.2 The Model Function Framework*

Depending on the effectiveness of the initialization, convergence to a local $\mathcal{H}_2$-optimal reduced model may require a large amount of optimization steps until convergence. As obvious from Algorithm 1, this implies the repeated reduction of the original, full-order model, at every step of the optimization. In some sense, the cost of reduction and the cost of optimization are tied together.

To alleviate the high cost resulting from repeated reduction, Panzer introduced in [103] the concept of *Model Function* Σ_μ, i.e., a surrogate model to be used during $\mathcal{H}_2$-optimization instead of the full model. As the author states, the main idea for this approach originated during the master thesis of Stefan Jaensch [77], who he supervised. This concept was further studied during this thesis in [34, 35], where it was shown that under certain *update conditions*, the optimization based on the Model Function would result in a reduced-order model satisfying the $\mathcal{H}_2$-optimality conditions with respect to the original model. Considerations on the initialization and the update of the Model Function have been presented and the idea has been extended to a general framework applying to interpolatory $\mathcal{H}_2$-optimal reduction for different system classes. Numerical investigations have demonstrated the substantial speedup that can be achieved by applying the framework to large-scale models.

The main motivation for the new framework arises from simple considerations on the *locality* of model reduction by tangential interpolation, as partly pointed out also by

Antoulas and Sorensen in [6]. In fact,

1. tangential interpolation only guarantees to yield a good approximation *locally* around the shifts $\{\sigma_i\}_{i=1}^n$, tangentially along directions $\{r_i\}_{i=1}^n$ and $\{l_i\}_{i=1}^n$,
2. as $\|G-G_r\|_{\mathcal{H}_2}$ is a non-convex function, in general only *local* optima can be found.

As such, the main idea behind the Model Function framework is to generate a good approximation of the full-order model, locally with respect to the initial interpolation data—and run surrogate optimization with respect to this local approximation. The new (optimal) interpolation data thus found is used to update the Model Function. Provided the optimal interpolation data converges, then the reduced-order model is guaranteed to satisfy the optimality conditions with respect to the full-order model, cf. [34, Theorem 4.2]. While LSEs with respect to the full-order model Σ are needed only when creating and updating the Model Function Σ_μ, the $\mathcal{H}_2$ optimization is performed on the generally much smaller model function Σ_μ, thus making the *optimization cost* negligible compared to the *reduction cost*. The details, derivations and proofs can be found in [35] for SISO and [34] for MIMO LTI models. Figure 1 gives a graphical overview of the Model Function framework, where n_μ^i represents the order of the Model Function at iteration i and k_μ is the number of iterations required for the Model Function framework to converge. The left side shows conventional $\mathcal{H}_2$-optimal reduction methods, while the right side illustrates the Model Function framework.

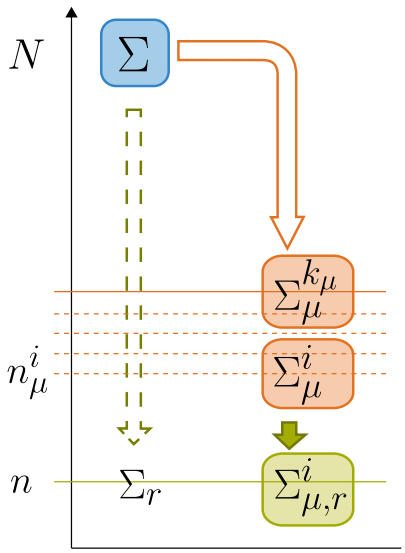


Figure 1: Schematic overview of the Model Function framework

As discussed in [34], this approach truly is a new framework for $\mathcal{H}_2$-optimal model reduction and not just an additional $\mathcal{H}_2$-optimal reduction algorithm. In fact, this framework can be applied to various interpolatory $\mathcal{H}_2$ reduction methods, such as IRKA, the trust region method of [14], and SPARK [103, 105]. First indications on how to apply this framework even to different system classes (e.g. nonlinear systems), for which interpolatory $\mathcal{H}_2$-optimal/-inspired algorithms are available, are given in [36] . When applied to IRKA, the resulting algorithm Confined IRKA (CIRKA) takes the form described in Algorithm 2.

Algorithm 2 Confined IRKA (CIRKA) [34]

Input: Σ; Initial interpolation data $\{\sigma_i\}_{i=1}^n$, $\{r_i\}_{i=1}^n$, $\{l_i\}_{i=1}^n$.
Output: reduced model $\Sigma_{\mu,r}$, Model Function $\Sigma_\mu^{k_\mu}$, error estimation $\tilde{\epsilon}_{\mathcal{H}_2}$

1: $k \leftarrow 0$; $\left[\Sigma_\mu^k, \{\sigma_{\mu,j}\}, \{r_{\mu,j}\}, \{l_{\mu,j}\}\right] \leftarrow$ empty; // Initialization
2: $\{\sigma_{*,i}\}_{i=1}^n \leftarrow \{\sigma_i\}_{i=1}^n$, $\{r_{*,i}\}_{i=1}^n \leftarrow \{r_i\}_{i=1}^n$, $\{l_{*,i}\}_{i=1}^n \leftarrow \{l_i\}_{i=1}^n$;
3: **while** not converged **do**
4: $k \leftarrow k+1$
5: $\Sigma_\mu^k \leftarrow$updateModelFunction$\left(\Sigma, \Sigma_\mu^{k-1}, S_*^{\text{tot}}, R_*^{\text{tot}}, L_*^{\text{tot}}, S_*^{k-1}, R_*^{k-1}, L_*^{k-1}\right)$
6: $\left[\Sigma_{\mu,r}, S_*^k, R_*^k, L_*^k\right] \leftarrow$ IRKA $(\Sigma_\mu^k, S_*^{k-1}, R_*^{k-1}, L_*^{k-1})$ // $\mathcal{H}_2$ optimization
7: $S_*^{\text{tot}} \leftarrow$blkdiag$\left(S_*^{\text{tot}}, S_*^k\right)$; $R_*^{\text{tot}} \leftarrow \left[R_*^{\text{tot}}, R_*^k\right]$; $L_*^{\text{tot}} \leftarrow \left[L_*^{\text{tot}}, L_*^k\right]$
8: **end while**
9: $k_\mu \leftarrow k$
10: $\tilde{\epsilon}_{\mathcal{H}_2} \leftarrow$norm$\left(G_\mu^{k_\mu} - G_{\mu,r}\right)$

Besides decoupling the cost of reduction from the cost of optimization and thus often accelerating $\mathcal{H}_2$-optimal reduction (there more so, the higher the original order N), this new framework naturally—i.e. at no additional cost—yields a middle-sized surrogate model, the Model Function Σ_μ, that can be used for a variety of purposes, first and foremost the estimation of the approximation error

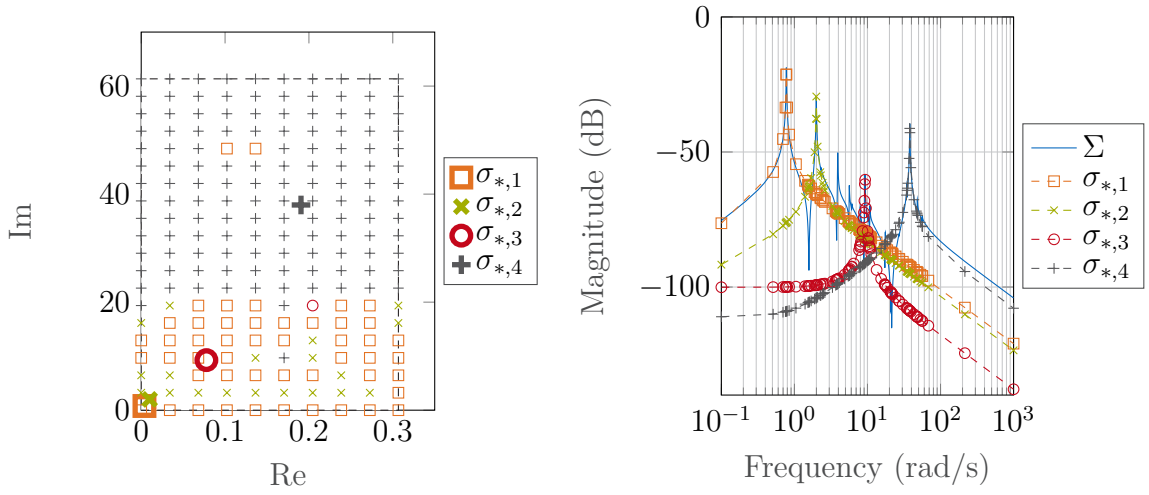
$$\|G - G_r\|_{\mathcal{H}_p} \approx \|G_\mu - G_r\|_{\mathcal{H}_p}. \tag{3.13}$$

Interpretations of the Model Function framework The Model Function framework presented above can be given different interpretations. On the one hand, it is a form of *surrogate optimization* (cf. e.g. [111]) in that the $\mathcal{H}_2$ optimization is not conducted on the actual cost function $\mathcal{J} = \|G - G_r\|_{\mathcal{H}_2}$ but on an approximation $\mathcal{J} \approx \widehat{\mathcal{J}} = \|G_\mu - G_r\|_{\mathcal{H}_2}$. In fact, the framework itself is an application of *reduced-model-based optimization*, cf. e.g. [26, 71]. On the other hand, the framework can be seen as $\mathcal{H}_2$ optimization in a subspace, defined by the matrices V_μ and W_μ, that is updated

at every iteration. In fact, this technique could be interpreted as a *subspace acceleration* method (cf. e.g. [113]) to recycle information obtained in a previous optimization step. Traditionally, subspace acceleration methods in numerics are used to ameliorate the convergence of iterative methods. In our setting, this becomes the convergence to a set of optimal reduction parameters. Finally, one may think of it as a *restarted* $\mathcal{H}_2$ optimization, where e.g. IRKA is restarted in a higher-dimensional subspace after convergence.

3.2.3 Globalized $\mathcal{H}_2$-Optimal Model Reduction*

The $\mathcal{H}_2$-optimal reduction problem defined in (3.11) is non-convex, meaning that in general one can only hope to find a local minimizer. However, when computational resources available to evaluate the models are limited, it is of particular interest to find *the best* $\mathcal{H}_2$-optimal reduced-order model for a given complexity, i.e. the *global* minimum. To analyze the impact of different initializations, Figure 2 summarizes the results of running IRKA for a SISO version of the ISS model, taken from the collection [37], generating optimal reduced models of order $n = 2$. A detailed description of the numerical setup can be found in [33]. Figure 2a depicts all initial shifts (small markers), as well as the respective IRKA fixed points, indicated by large markers of the same type. Figure 2b compares the local optimal reduced models to the full-order model in terms of Bode magnitude diagrams. As it can be seen, several local minima are found with very different approximation qualities.



(a) Initialization and fixed points.

(b) Bode diagrams of local optima.

Figure 2: The impact of different intializations in $\mathcal{H}_2$-optimal reduction of the ISS model.

To alleviate this dependency, [33] introduces a *globalized* approach to $\mathcal{H}_2$-optimal model reduction, based on performing an initial *global sampling* of the search space [73, 110] and subsequent (parallel) localized reduction using the $\mathcal{H}_2$ reduction algorithms of the previous sections. Preliminary work is documented in the masters thesis of Hu [74]. Even though this approach cannot guarantee, in general, that the global optimum will be found, in practical terms it increases the chances of finding it, with probability tending towards one as the coverage of the initial global sampling increases. In essence, what this approach does it to determine the best local optimum within the set of all local optima found. As the computation of the $\mathcal{H}_2$ norm of the error is not feasible for large-scale models (cf. (3.4)), a different strategy is required. By using the Hilbert projection theorem, it is shown in [33] that the best minimizer is the one with largest $\mathcal{H}_2$ norm. As this model has the reduced order n, its norm is readily computed. Further, the burden of repeatedly performing local $\mathcal{H}_2$ optimization from all initial samples, e.g. by IRKA, is alleviated by exploiting the Model Function framework of Section 3.2.2. In this way, the localized optimization can be performed efficiently with respect to the surrogate model Σ_μ. Details on effective initialization and update of the Model Function in this globalized setting, e.g. using *k-means clustering* [91], are given in [33].

3.3 $\mathcal{H}_\infty$-Optimal Model Reduction

As briefly mentioned in Section 2.1, the reduction methods of balanced truncation and optimal Hankel norm approximations (OHNA) both deliver global, rigorous error bounds on the $\mathcal{H}_\infty$ norm of the approximation error. Recall that, for these error bounds to hold, dense operations are required, making these approaches viable only for middle-sized problems. Noticing that the error bound (2.11) for the OHNA is smaller by a factor of two than the one by balanced truncation (2.10), it is not surprising that in practice OHNAs are known to yield small $\mathcal{H}_\infty$ approximation errors. Even so, neither of the two approaches is aimed at finding a reduced-order model with *minimal* $\mathcal{H}_\infty$ error, i.e., solving the problem

$$G_r(s) = \arg \min_{\deg \widehat{G}_r = n} \|G - \widehat{G}_r\|_{\mathcal{H}_\infty}. \tag{3.14}$$

The problem of $\mathcal{H}_\infty$-optimal reduction still lacks approaches that are both numerically efficient for large-scale problems and guarantee (at least local) optimality of the reduced-order model. Existing approaches include the linear matrix inequality (LMI)-based formulations in [63, 72, 131]. Unfortunately, these inequalities can be solved only

for very small problems. It was noted by Flagg [57] and later in Flagg, Beattie and Gugercin [54] that there is a connection between sufficient conditions for $\mathcal{H}_\infty$ optimality due to Trefethen [126] and the $\mathcal{H}_2$ interpolatory conditions of Meier and Luenberger [96]. In fact, a SISO reduced model that interpolates the original model at $2n+1$ interpolation points in the open right-half of the complex plane and yields an error with constant modulus over the imaginary axis is known to be an $\mathcal{H}_\infty$-optimal approximation. In the SISO case, $2n$ interpolation conditions can be achieved by means of rational interpolation, while an additional interpolation point is generally attained at $s \to \infty$ when $D_r = D$ is chosen. By modifying the reduced feed-through while preserving the original interpolation points, the additional interpolation point in the right half-plane can be introduced, while at the same time trying to achieve a nearly constant modulus of the error over the imaginary axis. The parametrization in terms of D_r of all reduced order models satisfying particular interpolatory conditions is due to Mayo and Antoulas [95] and Bettie and Gugercin [15]. While in these works particular projection bases are required, the result has been generalized to arbitrary bases during this thesis in [36]. In addition, constant modulus of the error system over the imaginary axis can be interpreted from approaches in *potential theory* in the complex plane as an equipotential with charges distributed symmetrically with respect to the imaginary axis. When charges are interpreted as transfer function poles and sinks as transfer function zeros, a link to $\mathcal{H}_2$-optimal reduction can be drawn, where the Meier-Luenberger conditions impose exactly this type of symmetric distribution with respect to the imaginary axis of poles and zeros (i.e. interpolation points).

In the following section we will discuss the extension of the Ansatz by Flagg, Beattie and Gugercin [54] to MIMO models that was presented in [36]. Before that, note that for any n^{th}-order approximation $G_r(s)$, following lower bound on the $\mathcal{H}_\infty$ error holds

$$\|G - G_r\|_{\mathcal{H}_\infty} \geq \varsigma_{n+1}, \tag{3.15}$$

where ς_{n+1} corresponds to the $n+1^{\text{st}}$ Hankel singular value. This lower bound can be used in a sufficient way to have an indication of "how far" a given reduced-order model is at most from the optimal $\mathcal{H}_\infty$ approximation.

3.3.1 MIMO Interpolatory $\mathcal{H}_\infty$ Approximations*

The manuscript [36] presents an interpolatory approach to generate near-optimal $\mathcal{H}_\infty$ approximations for MIMO models. In the MIMO case, the $\mathcal{H}_\infty$ norm of the error (i.e. the maximum singular value of the transfer function matrix $G(s) - G_r(s)$) may

not be analytic along the imaginary axis. For this reason, there does not seem to be a straightforward extension of Trefethen's result on sufficient conditions for $\mathcal{H}_\infty$ optimality. Nevertheless, it is easy to argue that, assuming the maximum singular value of the approximation error were constant along the imaginary axis, then a change in approximation parameters is likely to increase the maximum singular value at some frequency while possibly reducing it for some other. Following this thought (which will be to some extent confirmed by the numerical examples at the end of this section), also in the MIMO case it appears beneficial to start the generation of $\mathcal{H}_\infty$-optimal reduced-order models by first performing $\mathcal{H}_2$-optimal reduction, e.g., by IRKA.

Minimization of the $\mathcal{H}_\infty$ error with respect to D_r Once an $\mathcal{H}_2$-optimal reduced-order model is obtained, it is preferable to preserve the favorable selection of tangential interpolatory conditions. That said, the reduced-order model is uniquely identified except for the remaining $p+m$ parameters in the reduced feed-through matrix. Using the Sherman-Morrison-Woodbury formula, the modified reduced-order model can be decomposed into the sum of the IRKA reduced model and the change induced by the reduced feed-through, i.e.

$$G_r^D(s, D_r) = G_r^0(s) + \Delta G_r^D(s, D_r), \tag{3.16}$$

where the expressions for $G_r^D(s, D_r)$ are given in [36]. Accordingly, the $\mathcal{H}_\infty$-optimal reduced order model can be found by solving

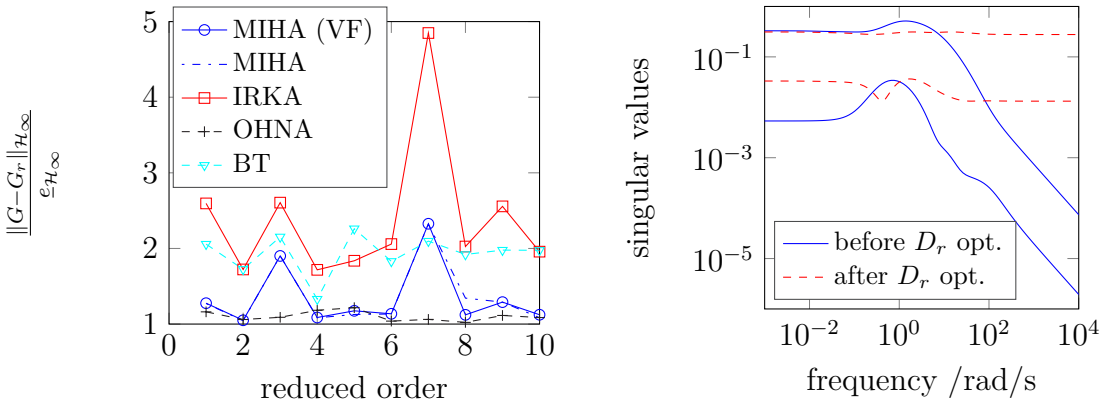
$$D_r^* = \arg\min_{D_r} \|G - G_r^D(D_r)\|_{\mathcal{H}_\infty} \tag{3.17}$$

with the additional constraint of requiring $G_r^D(s, D_r)$ to be stable.

Solving the $\mathcal{H}_\infty$ optimization efficiently If we could solve (3.17) efficiently, then the resulting reduced-order model would indeed be (locally) $\mathcal{H}_\infty$-optimal. However, as the error system $G(s) - G_r^D(s, D_r)$ is of order $N+n$, this computation is not feasible for large-scale models. To alleviate this problem, surrogate optimization is proposed. In fact, by using the samples of the transfer function of the full-order model taken during IRKA, *data-driven* approaches such as the Lowner framework [95] or vector fitting [43, 70] can be used to build a middle-sized surrogate model to be used during optimization. This makes this approach tractable even for very large-scale models. Nevertheless, as surrogate optimization is implemented, the resulting reduced-order model cannot be

expected to be optimal, but hopefully close to optimal.

Numerical results Figure 3 shows numerical results obtained with the `heatmodel` example. The approximation errors in Figure 3a are divided by the theoretical lower



(a) Comparison of different reduction algorithms.

(b) Singular value plot of the error before and after optimization.

Figure 3: Numerical results using MIHA on the `heatmodel` example.

bound (3.15), meaning that values close to 1 are actually (at least close to) optimal. Also note how the surrogate optimization with the vector fitting [43, 70] surrogate (MIHA VF) has only minimal influence on the quality of the reduction. For the same model, Figure 3b shows the singular values over the imaginary axis before and after the additional D_r optimization for a reduced order $n = 2$. Notice how the maximum singular value for the MIHA reduced model is almost perfectly flat over the imaginary axis, confirming our intuition on $\mathcal{H}_\infty$-optimal approximations.

4 Numerical Tools for Model Reduction

It is impossible to talk about the theory of model reduction without making a direct link to the numerical aspects of putting the developed algorithms into practice. In fact, reduction algorithms that are promising from a theoretical standpoint are of no use if they cannot be ultimately applied efficiently for the large-scale problems that arise in real-life applications, often exceeding the order of magnitude $\mathcal{O}(N) \approx 10^5 \dots 10^6$. While this chapter cannot go into the details of numerical linear algebra as required for model reduction, its scope is to raise the awareness to some particular topics that should be kept in mind and ultimately reference to existing software tools.

One central aspect in the numerical treatment of large-scale models is that the main limiting factor is determined by the available storage, while the required computational time to perform operations takes a subordinate role. Luckily, large-scale models often come in sparse representations [115], as generally the elements of the state vector are only influenced by neighboring elements. Therefore, it is of primary importance to preserve the sparsity throughout the numerical computations, as otherwise the resulting *fill-in* may cause to exceed the available memory. This is the primary reason why, e.g. the Lyapunov equations required for the balanced truncation reduction cannot be computed using direct methods such as Bartel-Stewart or Hammarling. Indeed, one of the few computations that can be performed efficiently for large, sparse models are LSEs. In fact, through reordering and clever pivoting, LU decompositions can be tailored to reduce the fill-in [39]. Even so, the size or the structure of the problem may be prohibitive even for sparse direct methods, requiring *indirect* methods for the solution of the LSEs, which involve only matrix-vector multiplications [115].

4.1 The `sss` and `sssMOR` Toolboxes*

During this thesis, the `sss` and `sssMOR` MATLAB toolboxes have been developed and released to make classic as well as modern efficient model reduction algorithms accessible to a vast audience of both expert and non-expert users. An introduction is given in [32].

While `sss` is focused on the definition and analysis of sparse large-scale models, `sssMOR` provides functions to efficiently reduce `sss` models. The toolboxes are characterized by exploiting syntax and functionality of MATLAB's Control System Toolbox[1], which by itself is unfortunately unfit to deal with large-scale models as the sparsity of the system matrices is destroyed. The separation of the two toolboxes bears the advantage of allowing the use of `sss` models also in other applications where reduction may not be required. An example of this is the taX toolbox[2] [48] for the definition and analysis of thermo-acustic networks.

Dynamical models can be defined using the command `sys = sss(A,B,C,D,E);` and based on these `sss` objects, a large number of functions known from the Control System Toolbox, such as `bode, step, impulse, isstable, norm` etc. has been included in the `sss` toolbox to analyze large-scale models in an efficient way. In addition, classic and state-of-the-art model reduction routines such as `projectiveMor, modalMor, tbr, rk, irka, isrk, cirka, gcirka, hinfMor` etc. are implemented to obtain reduced-order models preserving certain properties of the original model and targeted at obtaining minimal approximation errors with respect to different error measures.

The development has always had *usability* and *customizability* as the two primary goals. The former is achieved by consistency to MATLAB's built-in functions, comprehensive documentation and automatic choice of meaningful reduction parameters, forcing unexperienced users to select only as few reduction parameters as necessary. For example, the function `tbr` can be called with only the respective `sss` model as input, in which case a plot with the relative decay of Hankel singular values will appear, prompting the user to select a reduced order. As another example, the function `irka` can be called with as little as the `sss` model and a desired reduced order, performing internally all required initializations and selections of execution parameters. The latter is achieved by allowing the definition of a large number of execution parameters, which is achieved by simply passing one additional input, allowing experienced users an in-depth tailoring of the function execution.

The development of the `sss` and `sssMOR` toolboxes has involved overall a large number of people, starting from the initiators Heiko Peuscher (né Panzer), Rudy Eid and Sylvia Cremer who had the idea of using a "sparse state space" class to extend the capabilities of the built-in Control System Toolbox, and who programmed the core functionality of `sss` and `sssMOR` long before the research related to this thesis was

[1] MATLAB and Control System Toolbox 2016b, The MathWorks, Inc., Natick, Massachusetts, United States.

[2] Available at https://www.tfd.mw.tum.de/index.php?id=33

started. The desire to make the functions available to a vast audience is what has driven me to take on this project from the very beginning of my doctoral studies. From simple retouching and commenting, the desire to improve and increase the functionality has resulted in a large number of modifications and improvements, which ultimately led to the current form of the toolboxes as they are available on the homepage of the Chair of Automatic Control[3], on the File Exchange in MATLAB Central[4] and on the development platform GitHub[5]. This entire effort would not have been nearly as successful without the dedicated work of my colleague Maria Cruz Varona, who embarked this project enthusiastically right from the start, and a team of students that I would like to mention explicitly in chronological order: Lisa Jeschek, Jorge Luiz Moreira Silva, Rodrigo Mancilla, Siyang Hu, Michael Ott, Max Gille, Jonathan Seiti Miura, Jonas Ferber, Maximilian Loderer, Niklas Kochdumper and Paul Heidenreich. In addition, during the cooperation with the Professur of Thermofluiddynamik at Technical University of Munich, Stefan Jaensch and Thomas Emmert have contributed to the development of `sss`. Furthermore, the automatic generation of MATLAB documentation has been greatly supported by the cooperation with Philip Holzwarth and Nico-Philipp Walz from the MOREMBS[6] team. Last but not least, the integration of the M-M.E.S.S.[7] Toolbox [117] into `sss` to efficiently solve large-scale matrix equations has been supported by Jens Saak.

`sss` and `sssMOR` are free, open-source software tools with over 1500 downloads by the time this manuscript was written. The development has been moved to the well-known platform GitHub with the hope to engage the model reduction community in contributing, maintaining and improving the toolboxes, adding further reduction methods and hence providing a common platform to

a) compare different methods and reproduce numerical results from literature,

b) increase the number of available analysis and reduction methods for expert and non-expert users.

Finally, note that an extension for the analysis and reduction of parametric models has been recently added in form of the `psssMOR` toolbox[8].

[3] Available at https://www.rt.mw.tum.de/?sssMOR
[4] Available at https://de.mathworks.com/matlabcentral/fileexchange/59169-sssmor-toolbox
[5] Available at https://github.com/MORLab/sssMOR
[6] Available at http://www.itm.uni-stuttgart.de/research/morembs/
[7] Available at https://www.mpi-magdeburg.mpg.de/projects/mess
[8] Available at https://www.rt.mw.tum.de/forschung/morlab/software/psssmor/

4.2 Other Tools

In recent years, many efforts have been done in the model reduction community to develop efficient model reduction tools and making them publicly available. An updated overview of model reduction software can be found in the MORWiki [99] at https://morwiki.mpi-magdeburg.mpg.de/morwiki/index.php/Category:Software and is therefore omitted at this point for brevity. In addition, conceptual work has been put forward by Jörg Fehr, Jan Heiland, Christian Himpe and Jens Saak on how to generally increase replicability, reproducibility, reusability of numerical examples [50].

5 Summary of Achievements and Outlook

In this thesis, current advances in the fields of $\mathcal{H}_2$- and $\mathcal{H}_\infty$-optimal model reduction for linear time-invariant multiple-input multiple-output dynamical systems are presented. A new framework is introduced that can significantly reduce the cost of $\mathcal{H}_2$-optimal model reduction, making it even more efficient in the reduction of very large-scale models. At the same time, optimality is guaranteed by observing particular update conditions. Moreover, the Model Function resulting from this new framework has been used to estimate the reduction error. Furthermore, possible extensions of this framework to other system classes have been introduced. In addition, a first numerically efficient approach targeted at finding the *globally* $\mathcal{H}_2$-optimal reduced order model of fixed order has been presented. Moreover, a numerically efficient method to obtain reduced-order models with nearly-optimal $\mathcal{H}_\infty$-error for system with multiple-inputs multiple-outputs has been presented. Finally, numerical tools in the form of the `sss` and `sssMOR` MATLAB toolboxes have been presented, which deliver state-of-the-art model reduction algorithms to both expert and non-expert users.

The problem of finding *guaranteed* $\mathcal{H}_\infty$-optimal reduced-order models is still open and requires further investigation. More generally, the task to estimate rigorously and globally the reduction error is still open and could not be tackled in this thesis. Related to this is the problem of finding a suitable reduced order for model reduction by rational interpolation. That said, the Model Function resulting from the homonymous framework could be used in combination with the Cumulative Reduction (CURE) framework [103, 134] to construct reduced order models in a cumulative fashion while recycling and using the Model Function to estimate the reduction error. First investigations have been conducted in [74] and in unpublished work by the author of this thesis. Finally, numerical investigations on the effectiveness of applying the Model Function framework to other system classes are still open.

Acknowledgment of foreign scientific contributions

Following people have been involved to some extent in the work presented in this thesis:

- Stephan Jaensch and Heiko Peuscher (né Panzer) introduced the preliminary idea to use $\mathcal{H}_2$-surrogate optimization as a heuristic in the SPARK algorithm. To show the link to this first idea, the name "model function" denoting the surrogate model has been preserved also in this thesis, though the concept has been revised, improved, and extended into a rigorous framework.
- Maria Cruz Varona has contributed to the development of the `sss` and `sssMOR` toolboxes as well as to the respective paper.
- The main ideas behind the extension of the "Interpolatory $\mathcal{H}_\infty$ Approximation" algorithm to MIMO models were developed in close cooperation with Serkan Gugercin and Christopher Beattie during my visit at Virginia Tech.
- Following students have actively contributed to developing and maintaining the `sss` and `sssMOR` toolboxes: Lisa Jeschek, Jorge Luiz Moreira Silva, Rodrigo Mancilla, Siyang Hu, Michael Ott, Max Gille, Jonathan Seiti Miura, Jonas Ferber, Maximilian Loderer, Niklas Kochdumper and Paul Heidenreich.
- Michael Ott has contributed in his Bachelors Thesis to the analysis of different initialization strategies for $\mathcal{H}_2$-optimal model reduction.
- Siyang Hu has contributed in his Masters Thesis to the development of a globalized $\mathcal{H}_2$ reduction approach and to first investigations regarding the recycling of the Model Function within the Cumulative Reduction (CURE) framework to estimate a suitable reduced order exploiting the Model Function.

References

Own Publications

[32] A. Castagnotto, M. Cruz Varona, L. Jeschek, and B. Lohmann. sss & sssMOR: Analysis and Reduction of Large-Scale Dynamic Systems in MATLAB. *at-Automatisierungstechnik,* 65.2 (2017), 134–150 (cf. p. 27).

[33] A. Castagnotto, S. Hu, and B. Lohmann. An Approach for Globalized $\mathcal{H}_2$-Optimal Model Reduction. *IFAC-PapersOnLine,* 51.2 (2018), 196–201 (cf. pp. 17, 22, 23).

[34] A. Castagnotto and B. Lohmann. A New Framework for $\mathcal{H}_2$-Optimal Model Reduction. *Mathematical and Computer Modelling of Dynamical Systems,* (2018), 1–22 (cf. pp. 19–21).

[35] A. Castagnotto, H. K. F. Panzer, and B. Lohmann. Fast $\mathcal{H}_2$-Optimal Model Order Reduction Exploiting the Local Nature of Krylov-Subspace Methods. In: *2016 European Control Conference (ECC).* Aalborg, Denmark, 2016, 1958–1969 (cf. pp. 19, 20).

[36] A. Castagnotto, C. A. Beattie, and S. Gugercin. Interpolatory Methods for $\mathcal{H}_\infty$ Model Reduction of Multi-Input/Multi-Output Systems. In: *Model Reduction of Parametrized Systems.* Ed. by P. Benner, M. Ohlberger, A. Patera, G. Rozza, and K. Urban. Vol. 17. MS&A (Modeling, Simulation and Applications). Cham: Springer International Publishing, 2017, 349–365 (cf. pp. 21, 24, 25).

Other Publications

[1] V. Adamjan, D. Arov, and M. Krein. Infinite Hankel block matrices and related extension problems. *American Mathematical Society Translations,* 111 (1978), 133–156 (cf. p. 8).

[2] M. I. Ahmad, P. Benner, and P. Goyal. Krylov subspace-based model reduction for a class of bilinear descriptor systems. *J. Comput. Appl. Math.* 315 (2017), 303–318 (cf. p. 17).

[3] B. D. O. Anderson and A. C. Antoulas. Rational interpolation and state-variable realizations. *Linear Algebra Appl.* 137/138 (1990), 479–509 (cf. p. 5).

[4] A. C. Antoulas. *Approximation of Large-Scale Dynamical Systems.* Vol. 6. Advances in Design and Control. Philadelphia, PA: SIAM Publications, 2005 (cf. pp. 4, 8, 14, 15).

[5] A. C. Antoulas, D. C. Sorensen, and S. Gugercin. A survey of model reduction methods for large-scale systems. *Contemp. Math.* 280 (2001), 193–219 (cf. p. 4).

[6] A. Antoulas and D. Sorensen. Approximation of large-scale dynamical systems: An overview. *Int. J. Appl. Math. Comput. Sci.* 11.5 (2001), 1093–1121 (cf. pp. 11, 20).

[7] A. Astolfi. A new look at model reduction by moment matching for linear systems. In: *46th IEEE Conference on Decision and Control*. 2007, 4361–4366 (cf. pp. 5, 11).

[8] A. Astolfi. Model reduction by moment matching for linear and nonlinear systems. *IEEE Transactions on Automatic Control,* 55.10 (2010), 2321–2336 (cf. pp. 5, 11).

[9] Z. Bai, R. D. Slone, W. T. Smith, and Q. Ye. Error Bound for Reduced System Model by Padé Approximation via the Lanczos Process. *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.* 18.2 (1999), 133–141 (cf. p. 11).

[10] G. A. Baker and P. Graves-Morris. *Padé approximants*. Vol. 59. Cambridge University Press, 1996 (cf. p. 9).

[11] M. C. C. Bampton and R. R. Craig Jr. Coupling of substructures for dynamic analyses. *AIAA Journal,* 6.7 (1968), 1313–1319 (cf. p. 7).

[12] N. Banagaaya, L. Feng, W. Schoenmaker, P. Meuris, A. Wieers, R. Gillon, and P. Benner. Model order reduction for nanoelectronics coupled problems with many inputs. In: *Proceedings of the 2016 Conference on Design, Automation & Test in Europe*. 2016, 313–318 (cf. p. 3).

[13] L. Baratchart, M. Cardelli, and M. Olivi. Identification and rational L 2 approximation A gradient algorithm. *Automatica,* 27.2 (1991), 413–417 (cf. p. 15).

[14] C. A. Beattie and S. Gugercin. A trust region method for optimal $\mathcal{H}_2$ model reduction. In: *IEEE Conference on Decision and Control*. 2009 (cf. pp. 14, 17, 21).

[15] C. A. Beattie and S. Gugercin. Realization-independent $\mathcal{H}_2$-approximation. In: *51st IEEE Conference on Decision and Control*. 2012, 4953–4958 (cf. pp. 5, 17, 24).

[16] C. A. Beattie and S. Gugercin. Model Reduction by Rational Interpolation. In: *Model Reduction and Approximation: Theory and Algorithms*. SIAM, 2017, 297–334 (cf. pp. 6, 9, 10, 14–16).

[17] P. Benner. Numerical Linear Algebra for Model Reduction in Control and Simulation. *GAMM-Mitt.* 29.2 (2006), 275–296 (cf. p. 4).

[18] P. Benner and T. Breiten. Interpolation-based $\mathcal{H}_2$-model reduction of bilinear control systems. *SIAM J. Matrix Anal. Appl.* 33.3 (2012), 859–885 (cf. p. 17).

[19] P. Benner, M. Castillo, E. Quintana-Ortí, and G. Quintana-Ortí. Parallel Model Reduction of Large-Scale Unstable Systems. In: *Parallel Computing: Software Technology, Algorithms, Architectures & Applications*. Ed. by G. R. Joubert, W. E. Nagel, F. J. Peters, and W. V. Walter. Vol. 13. Advances in Parallel Computing. Elsevier, 2004, 251–258 (cf. p. 3).

[20] P. Benner and P. Goyal. Multipoint Interpolation of Volterra Series and $\mathcal{H}_2$-Model Reduction for a Family of Bilinear Descriptor Systems. *Syst. Control Lett.* 97 (2016), 1–11 (cf. p. 17).

[21] P. Benner, P. Goyal, and S. Gugercin. *$\mathcal{H}_2$-Quasi-Optimal Model Order Reduction for Quadratic-Bilinear Control Systems*. arXiv e-prints 1610.03279. Cornell University, 2016 (cf. p. 17).

[22] P. Benner, P. Kürschner, and J. Saak. A Reformulated Low-Rank ADI Iteration with Explicit Residual Factors. *Proc. Appl. Math. Mech.* 13.1 (2013), 585–586 (cf. p. 8).

[23] P. Benner and E. S. Quintana-Ortí. Model Reduction Based on Spectral Projection Methods. In: *Dimension Reduction of Large-Scale Systems*. Ed. by P. Benner, V. Mehrmann, and D. Sorensen. Vol. 45. Lect. Notes Comput. Sci. Eng. Springer-Verlag, Berlin/Heidelberg, Germany, 2005, 5–45 (cf. p. 3).

[24] P. Benner, E. S. Quintana-Ortí, and G. Quintana-Ortí. Computing Optimal Hankel Norm Approximations of Large-Scale Systems. In: *Proc. 43rd IEEE Conf. Decision Contr.* Omnipress, Madison, WI, 2004, 3078–3083 (cf. p. 8).

[25] P. Benner and A. Schneider. Model Reduction for Linear Descriptor Systems with Many Ports. In: *Progress in Industrial Mathematics at ECMI 2010*. Ed. by M. Günther, A. Bartel, M. Brunk, S. Schöps, and M. Striebel. Vol. 17. Mathematics in Industry. Berlin: Springer-Verlag, 2012, 137–143 (cf. p. 3).

[26] P. Benner, Z. Tomljanović, and N. Truhar. Optimal damping of selected eigenfrequencies using dimension reduction. *Numer. Lin. Alg. Appl.* 20.1 (2013), 1–17 (cf. p. 21).

[27] P. Benner, L. Feng, and E. B. Rudnyi. Using the superposition property for model reduction of linear systems with a large number of inputs. In: *Proceedings of the 18th International Symposium on Mathematical Theory of Networks & Systems*. 2008 (cf. p. 3).

[28] P. Benner, P. Kürschner, and J. Saak. Self-generating and efficient shift parameters in ADI methods for large Lyapunov and Sylvester equations. *Electronic Transactions on Numerical Analysis,* 43 (2014), 142–162 (cf. pp. 8, 18).

[29] P. Benner and T. Stykel. Model Order Reduction for Differential-Algebraic Equations: A Survey. In: *Surveys in Differential-Algebraic Equations IV*. Ed. by A. Ilchmann and T. Reis. Differential-Algebraic Equations Forum. Cham: Springer International Publishing, 2017, 107–160 (cf. p. 3).

[30] C. Brezinski. *Padé-type approximation and general orthogonal polynomials*. Springer, 1980 (cf. p. 5).

[31] A. Bunse-Gerstner, D. Kubalinska, G. Vossen, and D. Wilczek. h_2-norm optimal model reduction for large scale discrete dynamical MIMO systems. *J. Comput. Appl. Math.* 233.5 (2010), 1202–1216 (cf. pp. 15, 16).

[37] Y. Chahlaoui and P. Van Dooren. *A collection of benchmark examples for model reduction of linear time invariant dynamical systems*. Tech. rep. 2002–2. SLICOT Working Note, 2002 (cf. p. 22).

[38] R. R. Craig Jr. Substructure Methods in Vibration. *Journal of Vibration and Acoustics,* 117.B (1995), 207–213 (cf. p. 7).

[39] T. Davis. *Direct Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, 2006 (cf. pp. 4, 27).

[40] E. J. Davison. A method for simplifying linear dynamic systems. *IEEE Trans. Autom. Control,* AC–11 (1966), 93–101 (cf. p. 7).

[41] J. Demmel. *Applied Numerical Linear Algebra*. Society for Industrial and Applied Mathematics, 1997 (cf. pp. 4, 7).

[42] Z. Drmač, S. Gugercin, and C. A. Beattie. Quadrature-Based Vector Fitting for Discretized $\mathcal{H}_2$ Approximation. *SIAM Journal on Scientific Computing,* 37.2 (2015), A625–A652 (cf. p. 5).

[43] Z. Drmač, S. Gugercin, and C. A. Beattie. Vector Fitting for Matrix-valued Rational Approximation. *SIAM Journal on Scientific Computing,* 37.5 (2015), A2346–A2379 (cf. pp. 5, 25, 26).

[44] V. Druskin and V. Simoncini. Adaptive rational Krylov subspaces for large-scale dynamical systems. *Syst. Control Lett.* 60.8 (2011), 546–560 (cf. pp. 8, 18).

[45] V. Druskin, V. Simoncini, and M. Zaslavsky. Adaptive Tangential Interpolation in Rational Krylov Subspaces for MIMO Dynamical Systems. *SIAM J. Matrix Anal. Appl.* 35.2 (2014), 476–498 (cf. pp. 8, 18).

[46] R. Eid. Time domain model reduction by moment matching. Dissertation. Munich, Germany: Technical University of Munich, 2009 (cf. p. 18).

[47] R. Eid, H. K. F. Panzer, and B. Lohmann. *How to choose a single expansion point in Krylov-based model reduction*. Technical Reports on Automatic Control 2. Institute of Automatic Control, Technical University of Munich, 2009 (cf. p. 18).

[48] T. Emmert, S. Jaensch, C. Sovardi, and W. Polifke. taX—A Flexible Tool for Low-Order Duct Acoustic Simulation in Time and Frequency Domain. In: *Forum Acusticum, Krakow, Poland*. 2014, 7–12 (cf. p. 28).

[49] D. F. Enns. Model reduction with balanced realizations: An error bound and a frequency weighted generalization. In: *Proc. 23rd IEEE Conf. Decision Contr.* Vol. 23. 1984, 127–132 (cf. p. 8).

[50] J. Fehr, J. Heiland, C. Himpe, and J. Saak. Best Practices for Replicability, Reproducibility and Reusability of Computer-Based Experiments Exemplified by Model Reduction Software. *AIMS Mathematics,* 1.3 (2016), 261–281 (cf. p. 30).

[51] P. Feldmann. Model order reduction techniques for linear systems with large numbers of terminals. In: *Proceedings Design, Automation and Test in Europe Conference and Exhibition*. Vol. 2. 2004, 944–947 Vol.2 (cf. p. 3).

[52] P. Feldmann and R. W. Freund. Efficient linear circuit analysis by Padé approximation via the Lanczos process. *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.* 14 (1995), 639–649 (cf. p. 9).

[53] L. Feng, A. C. Antoulas, and P. Benner. Some a posteriori error bounds for reduced-order modelling of (non-)parametrized linear systems. *ESAIM: M2AN,* 51.6 (2017), 2127–2158 (cf. p. 11).

[54] G. M. Flagg, C. A. Beattie, and S. Gugercin. Interpolatory $\mathcal{H}_\infty$ Model Reduction. *Syst. Control Lett.* 62.7 (2013), 567–574 (cf. p. 24).

[55] G. M. Flagg and S. Gugercin. Multipoint Volterra Series Interpolation and $\mathcal{H}_2$ Optimal Model Reduction of Bilinear Systems. *SIAM J. Numer. Anal.* 36.2 (2015), 549–579 (cf. p. 17).

[56] G. Flagg, C. A. Beattie, and S. Gugercin. Convergence of the iterative rational Krylov algorithm. *Syst. Control Lett.* 61 (2012), 688–691 (cf. p. 17).

[57] G. M. Flagg. An Interpolation-Based Approach to the Optimal $\mathcal{H}_\infty$ Model Reduction. MA thesis. Virginia Polytechnic Institute and State University, 2009 (cf. p. 24).

[58] P. Fulcheri and M. Olivi. Matrix Rational H2 Approximation: A Gradient Algorithm Based on Schur Analysis. *SIAM Journal on Control and Optimization,* 36.6 (1998), 2103–2127. eprint: https://doi.org/10.1137/S0363012995284230 (cf. p. 15).

[59] K. Gallivan, E. Grimme, and P. Van Dooren. A rational Lanczos algorithm for model reduction. *Numer. Algorithms,* 12 (1996), 33–63 (cf. p. 9).

[60] K. Gallivan, A. Vandendorpe, and P. Van Dooren. Model reduction of MIMO systems via tangential interpolation. *SIAM J. Matrix Anal. Appl.* 26.2 (2004), 328–349 (cf. pp. 9, 10).

[61] K. Glover. All optimal Hankel-norm approximations of linear multivariable systems and their L^∞-error norms. *Internat. J. Control,* 39.6 (1984), 1115–1193 (cf. p. 8).

[62] W. B. Gragg and A. Lindquist. On the partial realization problem. *Linear Algebra Appl.* 50 (1983), 277–319 (cf. p. 5).

[63] K. M. Grigoriadis. Optimal $\mathcal{H}_\infty$ model reduction via linear matrix inequalities: continuous and discrete-time cases. In: *Proceedings of the 34th IEEE Conference on Decision and Control*. Vol. 3. 1995, 3074–3079 (cf. p. 23).

[64] E. J. Grimme. Krylov projection methods for model reduction. PhD thesis. Univ. of Illinois at Urbana-Champaign, USA, 1997 (cf. pp. 9, 18).

[65] S. Gugercin. Projection methods for model reduction of large-scale dynamical systems. PhD thesis. Rice University, 2002 (cf. pp. 11, 14, 17, 18).

[66] S. Gugercin and A. C. Antoulas. A survey of balancing methods for model reduction. In: *Proc. European Control Conf. ECC 2003, Cambridge, UK*. 2003 (cf. pp. 15, 17).

[67] S. Gugercin, A. C. Antoulas, and C. A. Beattie. $\mathcal{H}_2$ Model Reduction for Large-Scale Dynamical Systems. *SIAM J. Matrix Anal. Appl.* 30.2 (2008), 609–638 (cf. pp. 15–18).

[68] S. Gugercin, T. Stykel, and S. Wyatt. Model Reduction of Descriptor Systems by Interpolatory Projection Methods. *SIAM J. Sci. Comput.* 35.5 (2013), B1010–B1033 (cf. pp. 3, 17).

[69] S. Gugercin and A. Antoulas. An $\mathcal{H}_2$ error expression for the Lanczos procedure. In: *Proceedings. 42nd IEEE Conference on Decision and Control*. Vol. 2. 2003, 1869–1872 (cf. pp. 14, 18).

[70] B. Gustavsen and A. Semlyen. Rational approximation of frequency domain responses by vector fitting. *Power Delivery, IEEE Transactions on,* 14.3 (1999), 1052–1061 (cf. pp. 5, 25, 26).

[71] J. S. Han, E. B. Rudnyi, and J. G. Korvink. Efficient optimization of transient dynamic problems in MEMS devices using model order reduction. *Journal of Micromechanics and Microengineering,* 15.4 (2005), 822 (cf. p. 21).

[72] A. Helmersson. Model reduction using LMIs (1994) (cf. p. 23).

[73] R. Horst and P. M. Pardalos. Handbook of global optimization, volume 2 of Nonconvex Optimization and its Applications. *Kluwer Academic Publishers,* 2.3 (1995), 4 (cf. p. 23).

[74] S. Hu. Global $\mathcal{H}_2$-Optimal Model Reduction with Adaptive Choice of Reduced Order. MA thesis. Technical University of Munich, 2017 (cf. pp. 17, 23, 31).

[75] D. C. Hyland and D. Bernstein. The optimal projection equations for model reduction and the relationships among the methods of Wilson, Skelton, and Moore. *IEEE Transactions on Automatic Control,* 30.12 (1985), 1201–1211 (cf. pp. 15, 16).

[76] T. C. Ionescu, A. Astolfi, and P. Colaneri. Families of moment matching based, low order approximations for linear systems. *Systems & Control Letters,* 64 (2014), 47–56 (cf. pp. 5, 11).

[77] S. Jaensch. $\mathcal{H}_2$-optimale Entwicklungspunktwahl bei der Modellordnungsreduktion mit Krylov-Unterraum-Verfahren. MA thesis. Technical University of Munich, 2012 (cf. p. 19).

[78] T. Kailath. *Linear Systems*. Prentice-Hall, Inc., New Jersey, 1980 (cf. pp. 4, 7).

[79] R. Kalman. On partial realizations, transfer functions and canonical forms. *Acta Polytechnica Scandinavica,* 31 (1979), 9–32 (cf. p. 5).

[80] R. E. Kalman, P. L. Falb, and M. A. Arbib. *Topics in mathematical system theory*. Vol. 1. McGraw-Hill New York, 1969 (cf. pp. 4, 5).

[81] B. Kramer and S. Gugercin. Tangential interpolation-based eigensystem realization algorithm for MIMO systems. *Math. Comput. Model. Dyn. Syst.* 22.4 (2016), 282–306 (cf. p. 5).

[82] P. Kunkel and V. Mehrmann. *Differential-algebraic equations.* EMS textbooks in mathematics. European Math. Soc., Zürich, 2006 (cf. p. 3).

[83] P. Kürschner. Efficient Low-Rank Solution of Large-Scale Matrix Equations. PhD thesis. Shaker Verlag Aachen, 2016 (cf. pp. 8, 18).

[84] M. Lal and R. Mitra. Simplification of large system dynamics using a moment evaluation algorithm. *IEEE Transactions on Automatic Control,* 19.5 (1974), 602–603 (cf. p. 9).

[85] S. Lefteriu and A. C. Antoulas. A new approach to modeling multiport systems from frequency-domain data. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* 29.1 (2010), 14–27 (cf. p. 5).

[86] D. J. Leith and W. E. Leithead. Survey of gain-scheduling analysis and design. *International journal of control,* 73.11 (2000), 1001–1025 (cf. p. 4).

[87] A. Lepschy, G. Mian, G. Pinato, and U. Viaro. Rational L2 approximation: a non-gradient algorithm. In: *32nd IEEE Conference on Decision and Control.* Vol. 3. 1991, 2321–2323 (cf. p. 15).

[88] J.-R. Li and J. White. Reduction of large circuit models via low rank approximate gramians. *Int. J. Appl. Math. Comput. Sci.* 11.5 (2001), 1151–1171 (cf. p. 8).

[89] J. Liesen and Z. Strakos. *Krylov subspace methods: principles and analysis.* Numerical Mathematics and Scientific Computation. Oxford University Press, 2012 (cf. p. 7).

[90] L. Litz. *Reduktion der Ordnung linearer Zustandsraummodelle mittels modaler Verfahren.* Vol. 4. Hochschulsammlung Ingenieurwissenschaft : Datenverarbeitung. Stuttgart: Hochschul-Verlag, 1979, 164 (cf. p. 7).

[91] S. Lloyd. Least squares quantization in PCM. *IEEE transactions on information theory,* 28.2 (1982), 129–137 (cf. p. 23).

[92] C. Magruder, C. A. Beattie, and S. Gugercin. Rational Krylov methods for optimal $\mathcal{L}_2$ model reduction. In: *49th IEEE Conference on Decision and Control (CDC).* 2010, 6797–6802 (cf. p. 3).

[93] S. A. Marshall. An Approximate Method for reducing the Order of a Linear System. *Internat. J. Control,* 10 (1966), 642–643 (cf. p. 7).

[94] N. Martins, L. T. G. Lima, and H. J. C. P. Pinto. Computing dominant poles of power system transfer functions. *IEEE Trans. Power Syst.* 11.1 (1996), 162–170 (cf. p. 7).

[95] A. J. Mayo and A. C. Antoulas. A framework for the solution of the generalized realization problem. *Linear Algebra Appl.* 425.2-3 (2007), 634–662 (cf. pp. 5, 24, 25).

[96] L. Meier and D. G. Luenberger. Approximation of Linear Constant Systems. *IEEE Transactions on Automatic Control,* 12.5 (1967), 585–588 (cf. pp. 15, 16, 24).

[97] B. C. Moore. Principal Component Analysis in Nonlinear Systems: Preliminary Results. In: *18th IEEE Conference on Decision and Control including the Symposium on Adaptive Processes.* Vol. 2. 1979, 1057–1060 (cf. p. 7).

[98] B. C. Moore. Principal component analysis in linear systems: controllability, observability, and model reduction. *IEEE Trans. Autom. Control,* AC–26.1 (1981), 17–32 (cf. p. 7).

[99] The MORwiki Community. *MORWiki - Model Order Reduction Wiki.* http://modelreduction.org (cf. p. 30).

[100] C. Mullis and R. Roberts. Synthesis of minimum roundoff noise fixed point digital filters. *IEEE Transactions on Circuits and Systems,* 23.9 (1976), 551–562 (cf. p. 7).

[101] M. Ott. Strategien zur Initialisierung der Entwicklungspunkte für $\mathcal{H}_2$-Optimale Modellordnungsreduktion. Bachelor Thesis. Technical University of Munich, 2016 (cf. pp. 17, 18).

[102] H. Padé. *Sur la représentation approchée d'une fonction par des fractions rationnelles.* 740. Gauthier-Villars et fils, 1892 (cf. p. 5).

[103] H. K. F. Panzer. Model Order Reduction by Krylov Subspace Methods with Global Error Bounds and Automatic Choice of Parameters. Dissertation. Munich, Germany: Technical University of Munich, 2014 (cf. pp. 17–19, 21, 31).

[104] H. K. F. Panzer, T. Wolf, and B. Lohmann. H_2 and H_∞ Error Bounds for Model Order Reduction of Second Order Systems by Krylov Subspace Methods. In: *Proc. European Control Conf. ECC 2013, Zurich.* 2013, 4484–4489 (cf. p. 11).

[105] H. Panzer, S. Jaensch, T. Wolf, and B. Lohmann. A Greedy Rational Krylov Method for H_2-Pseudooptimal Model Order Reduction with Preservation of Stability. In: *Proceedings of the American Control Conference.* 2013, 5512–5517 (cf. pp. 17, 18, 21).

[106] B. Peherstorfer, S. Gugercin, and K. Willcox. Data-Driven Reduced Model Construction with Time-Domain Loewner Models. *SIAM Journal on Scientific Computing,* 39.5 (2017), A2152–A2178 (cf. p. 5).

[107] T. Penzl. Algorithms for model reduction of large dynamical systems. *Linear Algebra Appl.* 415.2–3 (2006), 322–343 (cf. p. 8).

[108] T. Penzl. A cyclic low-rank Smith method for large sparse Lyapunov equations. *SIAM Journal on Scientific Computing,* 21.4 (2000), 1401–1418 (cf. pp. 8, 18).

[109] L. T. Pillage and R. A. Rohrer. Asymptotic waveform evaluation for timing analysis. *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.* 9.4 (1990), 352–366 (cf. pp. 5, 9).

[110] J. D. Pintér. *Global optimization in Action.* Dordrecht-Boston-London: Kluwer Academic Publishers, 1996, 54–63 (cf. p. 23).

[111] N. V. Queipo, R. T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, and P. K. Tucker. Surrogate-based analysis and optimization. *Progress in aerospace sciences,* 41.1 (2005), 1–28 (cf. p. 21).

[112] R. Rao and S. K. Mitra. Further Contributions to the Theory of Generalized Inverses of Matrices and Its Applicaitons. *Sankhya: The Indian Journal of Statistics,* 33.3 (1971), 289–300 (cf. p. 6).

[113] J. Rommes and N. Martins. Efficient Computation of Multivariable Transfer Function Dominant Poles Using Subspace Acceleration. *IEEE Transactions on Power Systems,* 21.4 (2006), 1471–1483 (cf. pp. 7, 19, 22).

[114] A. Ruhe. Rational Krylov algorithms for nonsymmetric eigenvalue problems. In: *Recent Advances in Iterative Methods.* Ed. by G. Golub, M. Luskin, and A. Greenbaum. Vol. 60. The IMA Volumes in Mathematics and its Applications. Springer, New York, 1994, 283–295 (cf. p. 9).

[115] Y. Saad. *Iterative methods for sparse linear systems.* Second. Society for Industrial and Applied Mathematics, 2003 (cf. pp. 4, 27).

[116] J. Saak. Efficient Numerical Solution of Large Scale Algebraic Matrix Equations in PDE Control and Model Order Reduction. PhD thesis. Chemnitz University of Technology, 2009 (cf. p. 8).

[117] J. Saak, M. Köhler, and P. Benner. M-M.E.S.S.-1.0.1 - The Matrix Equation Sparse Solver Library (2016) (cf. p. 29).

[118] B. Salimbahrami and B. Lohmann. *Krylov Subspace Methods in Linear Model Order Reduction: Introduction and Invariance Properties.* Tech. rep. Institute of Automation, University of Bremen, 2002 (cf. p. 6).

[119] Y. Shamash. Model reduction using the Routh stability criterion and the Padé approximation technique. *International Journal of Control,* 21.3 (1975), 475–484 (cf. p. 5).

[120] V. Simoncini. A new iterative method for solving large-scale Lyapunov matrix equations. *SIAM Journal on Scientific Computing,* 29.3 (2007), 1268–1288 (cf. p. 8).

[121] K. Sinani. Iterative Rational Krylov Algorithm for Unstable Dynamical Systems and Genaralized Coprime Factorizations. MA thesis. Virginia Tech, 2016 (cf. p. 3).

[122] J. T. Spanos, M. H. Milman, and D. L. Mingori. A new algorithm for L2 optimal model reduction. *Automatica,* 28.5 (1992), 897–909 (cf. p. 15).

[123] T. Stykel. Gramian-based model reduction for descriptor systems. *Math. Control Signals Systems,* 16.4 (2004), 297–319 (cf. p. 3).

[124] T.-J. Su and R. R. Craig jr. Model reduction and control of flexible structures using Krylov vectors. *Journal of Guidance, Control, and Dynamics,* 14.2 (1991), 260–267 (cf. p. 7).

[125] T. Takagi and M. Sugeno. Fuzzy identification of systems and its applications to modeling and control. In: *Readings in Fuzzy Sets for Intelligent Systems*. Elsevier, 1993, 387–403 (cf. p. 4).

[126] L. Trefethen. Rational Chebyshev approximation on the unit disk. *Numerische Mathematik,* 37.2 (1981), 297–320 (cf. p. 24).

[127] M. M. Uddin. Computational methods for model reduction of large-scale sparse structured descriptor systems. PhD thesis. Magdeburg, Germany: Otto-von-Guericke-Universität, 2015 (cf. p. 3).

[128] P. Van Dooren, K. Gallivan, and P.-A. Absil. $\mathcal{H}_2$-optimal model reduction of MIMO systems. *Appl. Math. Lett.* 21 (2008), 1267–1273 (cf. pp. 15, 16).

[129] P. Van Dooren, K. A. Gallivan, and P.-A. Absil. $\mathcal{H}_2$-optimal model reduction with higher-order poles. *SIAM Journal on Matrix Analysis and Applications,* 31.5 (2010), 2738–2753 (cf. p. 16).

[130] A. Varga. Enhanced modal approach for model reduction. *Math. Model. Syst.* 1.2 (1995), 91–105 (cf. p. 7).

[131] A. Varga and P. Parrilo. Fast algorithms for solving $\mathcal{H}_\infty$-norm minimization problems. In: *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on*. Vol. 1. 2001, 261–266 (cf. p. 23).

[132] D. Villemagne and R. E. Skelton. Model Reduction using a Projection Formulation. *Internat. J. Control,* 46 (1987), 2141–2169 (cf. pp. 5, 9).

[133] D. A. Wilson. Optimum Solution of Model-Reduction Problem. *Proceedings of the Institution of Electrical Engineers,* 117.6 (1970), 1161–1165 (cf. pp. 15, 16).

[134] T. Wolf. H_2 Pseudo-Optimal Model Order Reduction. Dissertation. Munich, Germany: Technical University of Munich, 2015 (cf. pp. 5, 31).

[135] T. Wolf and H. Panzer. The ADI iteration for Lyapunov equations implicitly performs $\mathcal{H}_2$ pseudo-optimal model order reduction. *International Journal of Control,* 89.3 (2016), 481–493 (cf. pp. 8, 18).

[136] T. Wolf, H. K. Panzer, and B. Lohmann. $\mathcal{H}_2$ pseudo-optimality in model order reduction by Krylov subspace methods. In: *European Control Conference (ECC)*. 2013, 3427–3432 (cf. p. 5).

[137] T. Wolf, H. K. Panzer, and B. Lohmann. ADI iteration for Lyapunov equations: A tangential approach and adaptive shift selection. *Applied Numerical Mathematics,* 109 (2016), 85–95 (cf. p. 8).

[138] S. Wyatt. Issues in Interpolatory Model Reduction: Inexact Solves, Second Order Systems and DAEs. PhD thesis. Blacksburg, Virginia, USA: Virginia Polytechnic Institute and State University, 2012 (cf. p. 17).

[139] Y. Xu and T. Zeng. Optimal $\mathcal{H}_2$ Model Reduction for Large Scale MIMO Systems via Tangential Interpolation. *International Journal of Numerical Analysis and Modeling,* 8.1 (2011), 174–188 (cf. p. 15).

[140] Y. Xu and T. Zeng. Fast optimal $\mathcal{H}_2$ model reduction algorithms based on Grassmann manifold optimization. *International Journal of Numerical Analysis and Modeling,* 10 (2013), 972–991 (cf. p. 15).

[141] W.-Y. Yan and J. Lam. An approximate approach to $\mathcal{H}_2$ optimal model reduction. *IEEE Transactions on Automatic Control,* 44.7 (1999), 1341–1358 (cf. p. 15).

[142] V. Zakian. Simplification of linear time-invariant systems by moment approximants. *International Journal of Control,* 18.3 (1973), 455–460 (cf. p. 9).

Appendix A

Reproduction of publications

A.1 Fast $\mathcal{H}_2$-Optimal Model Order Reduction Exploiting the Local Nature of Krylov-Subspace Methods

Summary: This contribution presents a new approach, based on the heuristic Model Function idea presented by Heiko K. F. Panzer in his doctoral thesis, to accelerate $\mathcal{H}_2$-optimal model reduction while guaranteeing optimality at convergence. This paper revises the surrogate optimization based on intermediate reduced-order models, namely the Model Functions, and introduces proof of satisfaction of first-order $\mathcal{H}_2$-optimality conditions if certain update conditions are ensured. The motivation in this paper was taken from observing that the well-known IRKA method for $\mathcal{H}_2$-optimal reduction can sometimes be computationally more demanding than dense methods such as balanced truncation. Exploiting the local nature of rational interpolation (depending on the context sometimes also referred to as *Krylov-subspace method*), surrogate optimization is motivated and introduced. The paper results in a modified version of IRKA by the name of "Confined IRKA", as the validity of the method is confined locally around the interpolation points chosen to define the Model Function. In preliminary considerations, it is shown how CIRKA can *decouple* the cost of optimization from the cost of reduction, hence allowing for significant speedup. This contribution focuses exclusively on models with single input and single output.

Contribution(s): Derivations, analysis, software development, numerical examples and writing have been conducted predominantly by the first author.

2016 European Control Conference
June 29 - July 1, 2016. Aalborg, Denmark

Fast $\mathcal{H}_2$-Optimal Model Order Reduction Exploiting the Local Nature of Krylov-Subspace Methods*

Alessandro Castagnotto, Heiko K. F. Panzer, and Boris Lohmann

***Abstract*— Rational Krylov-subspace methods are a predestined candidate in the reduction of very-large-scale linear models due to their moderate computational cost and memory requirements. However, in order to achieve good approximation results, state-of-the-art Krylov algorithms like IRKA iteratively search for a set of locally $\mathcal{H}_2$-optimal reduction parameters. This search requires the repeated reduction of the high-dimensional model and can therefore still account for significant computational cost, especially in case of slow convergence. In this contribution, we investigate the cost of $\mathcal{H}_2$-optimal rational Krylov methods and propose an enhanced reduction framework, based on the local nature of such methods, to reduce the computational effort while guaranteeing optimality at convergence. The improvement achieved through this framework is analyzed theoretically and validated numerically on a modified IRKA algorithm.**

I. Introduction

A. Motivation and problem statement

The accurate description of dynamical systems can result in high-fidelity models of prohibitively large dimensions. This is the case, for example, in models resulting from the spatial discretization of partial differential equations over a fine grid. Simulations and optimizations based on these large models are computationally expensive, if at all feasible. To resolve this issue, model reduction techniques that generate good approximated models in a numerically efficient manner are required.

Krylov-subspace methods (also known as *rational interpolation* or *moment matching* techniques) are known to be suitable for the reduction of very-large-scale models, since they require only the solution of shifted linear systems of equations (LSE). In fact, storage requirements can be reduced significantly even when applying direct methods like LU decomposition by exploiting the sparsity that typically characterizes the matrices. However, the reduction quality of these methods relies heavily on the choice of appropriate reduction parameters, like reduced order, matching frequencies and tangential directions. Fortunately, in the past decade a range of numerically tractable Krylov-subspace methods that find locally optimal parameters were developed. In practice, however, it turns out that other methods like *balanced truncation* or *modal reduction* might outperform the $\mathcal{H}_2$-optimal Krylov methods in terms of computational time.

*The work related to this contribution is supported by the German Research Foundation (DFG), Grant LO408/19-1.

Chair of Automatic Control, Technical University of Munich, Boltzmannstr. 15, D-85748 Garching. e-mail: a.castagnotto@tum.de

In this contribution, we will inspect this intrinsic issue of $\mathcal{H}_2$-optimal reduction procedures by analyzing the computational cost of the methods. Subsequently, we will propose a framework that reduces the computational complexity of finding optimal parameters by introducing the concept of *model functions*. Finally, we will prove that this procedure effectively achieves $\mathcal{H}_2$-optimality.

For simplicity of exposure, the discussion will be restricted to the case of systems with a single input and a single output (SISO). The results can be further generalized to include the multiple input, multiple output (MIMO) case, as will be discussed in the full paper.

B. Notation

$\Sigma = \{E, A, B, C, D\}$ denotes a dynamical system by its realization, i.e. the set of matrices defining state and output equations. Σ^μ denotes a *model function*, a concept we will introduce in Section IV-A. Accordingly, all related quantities will be indicated by a superscript μ. The amount of iterations required until convergence will be represented by k, while a subscript i as in σ_i denotes either an element within a set or the value at a certain iteration depending on the context. $\mathcal{C}_{N,n}(\cdot)$ is used to express the asymptotic arithmetic complexity of a mathematical operation with problem size N and reduced order n. The image of a matrix A, i.e. the subspace spanned by its columns, is denoted by $\mathrm{Im}(A)$.

II. Preliminaries

A. Model reduction by projection

Consider a high-order linear state-space model

$$\begin{aligned} E\,\dot{x} &= A\,x + B\,u, \\ y &= C\,x + D\,u, \end{aligned} \tag{1}$$

where $E \in \mathbb{R}^{N\times N}$ is the regular *descriptor matrix*, $A \in \mathbb{R}^{N\times N}$ is the system matrix and $x \in \mathbb{R}^N$, $u \in \mathbb{R}^1$, $y \in \mathbb{R}^1$ represent the state, input and output of the system respectively. The goal of model reduction within this contribution is to reduce the full order model (FOM) (1) to a reduced order model (ROM) obtained by a *Petrov-Galerkin* projection

$$\begin{aligned} \overbrace{W^\top E V}^{E_r}\,\dot{x}_r &= \overbrace{W^\top A V}^{A_r}\,x_r + \overbrace{W^\top B}^{B_r}\,u, \\ y_r &= \underbrace{C V}_{C_r}\,x_r + \underbrace{D}_{D_r}\,u, \end{aligned} \tag{2}$$

where $x_r \in \mathbb{R}^n$ $(n \ll N)$ represents the reduced state vector. For brevity, we will refer to the ROM in (2) through Σ_r and will use the shorthand notation $\Sigma_r = W^\top \Sigma V$ to underline

the projective nature. In fact, in this framework the task of finding a good ROM translates to finding appropriate reduction matrices W and V.

Note that in a control setting, the main goal is not the reconstruction of the whole state vector x but rather the approximation of the input-output behavior of the system, which can be characterized in the frequency domain by the rational transfer function

$$G(s) := \frac{\mathcal{L}\{y(t)\}}{\mathcal{L}\{u(t)\}} = C\left(sE - A\right)^{-1} B + D, \tag{3}$$

i.e. the Laplace transform of the impulse response.

B. Balanced truncation

A well-known method and probably gold standard for the computation of W, V is given by *balanced truncation* (BT) [1], [2]. This technique is based on obtaining a balanced realization, i.e. a realization in which every state is equally controllable and observable. In general, it requires the computation of the Cholesky factors of the controllability and observability Gramians $P = S^\top S$ and $Q = R^\top R$ defined by

$$APE^\top + EPA^\top + BB^\top = 0, \tag{4a}$$
$$A^\top QE + E^\top QA + C^\top C = 0, \tag{4b}$$

as well as the singular value decomposition of the product $SER^\top$. The reduction is then performed by truncating the states corresponding to small singular values, for they are both hard to reach and to observe.

This technique requires the solutions of generalized Lyapunov equations, which are dense and hence prohibitively large to store. Extensions of this procedure that are more suitable for large-scale systems are available, e.g. in [3].

C. Implicit moment matching by Krylov-subspace methods

A different choice of W, V is given by matrices spanning appropriate Krylov subspaces. In fact, if V, W span n^{th}-order *input* and *output rational Krylov (RK) subspaces*

$$\mathcal{K}_n\left((A - \sigma E)^{-1} E, (A - \sigma E)^{-1} B\right), \tag{5a}$$
$$\mathcal{K}_n\left((A - \sigma E)^{-\top} E^\top, (A - \sigma E)^{-\top} C^\top\right), \tag{5b}$$

then the reduced order model is guaranteed to match $2n$ moments of the FOM, that is Taylor series coefficients about $\sigma \in \mathbb{C}$ [4]. Note that an analogous result applies to the case where σ is not a scalar value but a set of complex frequencies $\sigma = \{\sigma_i\}_{i=1}^q$ with associated dimensions of the respective Krylov subspaces $\{n_i\}_{i=1}^q$ where $\sum_{i=1}^q n_i = n$. In this case, known as *multipoint Padé*, the projection subspace is given by the union of all individual rational Krylov subspaces. In the following we will make no distinction in the notation for the sake of simplicity.

The special case where both input and output Krylov subspaces are computed with the same shifts σ is called *Hermite* interpolation and will play a central role in the following. In fact, by the moment matching theorem, if $(A - \sigma E)^{-1} B \in \text{Im}(V)$ and $(A - \sigma E)^{-\top} C^\top \in \text{Im}(W)$, then it holds

$$G(\sigma) = G_r(\sigma), \tag{6a}$$
$$G'(\sigma) = G_r'(\sigma), \tag{6b}$$

where $G'(s)$ denotes the first derivative with respect to s [5]. In the following, we shall refer to a Hermite interpolation as stated above through the shorthand notation $\Sigma_r = \text{RK}(\Sigma, \sigma)$.

Note that given a set σ of complex shifts, the Krylov subspaces defined in (5) can be characterized equivalently by the following Sylvester equations

$$AV - EVS_V - BT = 0, \tag{7a}$$
$$A^\top W - E^\top W S_V^\top - C^\top T = 0, \tag{7b}$$

where S_V is a Jordan matrix that encodes the shifts and T is a row vector of zeros and ones [6].

In this setting, the task of model reduction translates to finding appropriate shifts σ and reduced order n.

D. $\mathcal{H}_2$-optimal reduction

The quality of the ROM obtained through Krylov-subspace methods relies heavily on the choice of reduction parameters. For this reason, the task of finding optimal parameters has been studied extensively in the literature [5], [7], [8].

It appears that the most tractable algorithms aim at minimizing the $\mathcal{H}_2$-norm of the error, i.e.

$$G_r(s) = \underset{\dim(\widetilde{G}_r) = n}{\text{argmin}} \left\| G(s) - \widetilde{G}_r(s) \right\|_{\mathcal{H}_2}. \tag{8}$$

Often times $\mathcal{H}_2$-optimal ROMs are characterized by the *Meier-Luenberger* necessary conditions they satisfy, i.e.

$$G(-\overline{\lambda}_{r,i}) = G_r(-\overline{\lambda}_{r,i}), \tag{9a}$$
$$G'(-\overline{\lambda}_{r,i}) = G_r'(-\overline{\lambda}_{r,i}), \tag{9b}$$

where $\lambda_{r,i}$ are the eigenvalues of the ROM.

The most simple and prominent procedure aimed at constructing $\mathcal{H}_2$-optimal ROMs is the Iterative Rational Krylov Algorithm (IRKA). It represents a fixed point iteration resulting from a simplified Newton procedure [5].

Algorithm 1 Iterative Rational Krylov Algorithm (IRKA)

Input: $\Sigma, \sigma, \text{tol}$
Output: locally $\mathcal{H}_2$-optimal reduced model Σ_r, σ^*

```
while relative change of σ > tol do
    Σ_r ← RK(Σ, σ)                    // Hermite interpolation
    σ ← −λ̄(Σ_r)                       // mirror reduced eigenvalues
end while
σ* ← σ                                // return optimal shifts
```

A trust-region method by the same authors that uses gradient and Hessian to guarantee a monotonic decrease of the error in each iteration is given in [7]. Moreover, the procedure in [8], [9] uses a trust-region method and further allows for an adaptive selection of the reduced order.

Due to its simplicity and widespread acceptance, we shall focus on the IRKA algorithm in the following as a representative for $\mathcal{H}_2$-optimal MOR techniques.

III. The cost of $\mathcal{H}_2$-optimal reduction

The observation that $\mathcal{H}_2$-optimal reduction methods can be outperformed in terms of computation time by other methods like balanced truncation motivates us to inspect the operations involved in the procedures and try to analyze the factors responsible for the main burden.

The computational efficiency of an algorithm can be assessed according to different criteria (e.g. usage of resources, execution time) and is affected by a variety of factors [10]. While the amount of operations (flops) required to obtain a desired solution gives an indication about the amount of work to be processed, the implementation on hardware plays an ever more important role. Research areas such as computational science and engineering (CSE) and high-performance computing focus on this issue [11]. Going into the algorithmic implementation details would exceed the scope of this treatise by far. Instead, in this section we look at $\mathcal{H}_2$-optimal reduction methods from a theoretical standpoint and try to understand which operations are the most intensive with regards to arithmetics.

A. Comparison between BT and IRKA

The predominant cost in the reduction by RK lies in the solution of large-scale linear systems of equations (LSE) needed to determine a basis for the Krylov subspaces (5). For this task *sparse direct* and *iterative* solvers are available to deal with the dimensionality of large-scale problems. On the other hand, the reduction by BT requires the computation of the Cholesky factors of the Gramian matrices solving generalized Lyapunov equations (4) (LyapChol). Using direct methods, this involves operations on dense matrices and is thus only feasible for middle-sized models. Iterative solvers like LRCF-ADI [3] or RK-subspace methods [12] increase the feasible problem size significantly.

Regardless of the individual choice of algorithm, a comparison between RK and BT yields following ratio for the asymptotic arithmetic complexities $\mathcal{C}_{N,n}(\cdot)$ when reducing a model of full order N to a reduced order n:

$$\frac{\mathcal{C}_{N,n}(\text{RK})}{\mathcal{C}_{N,n}(\text{BT})} \approx \underbrace{\frac{\mathcal{C}_N(\text{LSE})}{\mathcal{C}_N(\text{LyapChol})}}_{\alpha} \cdot n. \tag{10}$$

The numerical value of α clearly depends on the choice of algorithms for the solution of LSE and LyapChol, as well as problem specific quantities such as sparsity and structure of the system matrices at hand. Assuming dense direct solvers, for instance, the ratio becomes $\alpha = \frac{1}{136}$ [13], [14], but is generally significantly smaller when applying sparse direct solvers [15], [16]. This is demonstrated in Figure 1, where α is estimated by comparing the execution times of `mldivide` and `lyapchol` in MATLAB, for some sparse benchmark models from the collection [17].

Whichever the chosen low level algorithms and hence the value for α, when applying $\mathcal{H}_2$-optimal reduction methods the reduction by RK has to be repeated as many times as needed until a set of optimal parameters is found. For instance, if IRKA requires k_{IRKA} steps to reach convergence,

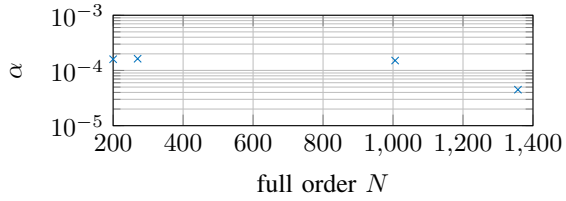


Fig. 1. Complexity ratio α, estimated for direct sparse LSE solvers on a selection of benchmark models.

then the arithmetic complexity ratio is given as

$$\frac{\mathcal{C}_{N,n}(\text{IRKA})}{\mathcal{C}_{N,n}(\text{BT})} \approx \frac{\mathcal{C}_{N,n}(\text{RK}) \cdot k_{IRKA}}{\mathcal{C}_{N,n}(\text{BT})} \approx \alpha \cdot n \cdot k_{IRKA}. \tag{11}$$

From this comparison we see that the search for optimal parameters is weighted with the full cost of performing a reduction by RK and that if k_{IRKA} becomes large, the advantage with respect to BT might be lost. As it is shown in the next section, the two problems of *reduction* (i.e. obtaining information from the FOM) and *optimization* (i.e. finding suitable reduction parameters) can be somewhat decoupled by exploiting the local nature of Krylov-subspace methods.

IV. Model Functions in $\mathcal{H}_2$-optimal MOR

To cope with the main computational burden of finding locally optimal reduction parameters in this section we propose a reduction framework for *any* $\mathcal{H}_2$-optimal reduction procedure that starts from an initial guess of the optimal shifts σ_0 and converges to a locally optimal solution σ^*. Note that we restrict the discussion to SISO systems for brevity and that the generalization for MIMO will be given in the full paper.

A. The main idea - Model Function MOR

Assume we are given a FOM of high order N and would like to obtain an $\mathcal{H}_2$-optimal ROM of much smaller order n. In the pursuit of this goal, at every step i along the way to the optimal solution σ^*, we need to reduce the FOM at σ_i and decide how to proceed (evaluate gradient and Hessian or compute the reduced eigenvalues). Depending on how many steps (k) are required to converge, this might require several reductions and is responsible for the predominant cost of such techniques.

Recall that RK methods have a local validity (as opposed to e.g. BT) in the sense that all we know is that we might expect the ROM to be a good approximation of the FOM around chosen frequencies σ. Since the minimization of the $\mathcal{H}_2$-norm of the error is a non-convex problem, all we might hope for is to find a local optimum around σ. Therefore, it might be enough to use a local approximation of the FOM around these frequencies and run the optimizer on it. This observation leads us to the definition of a *model function* Σ^μ.

Definition 1: Let Σ be the FOM of order N we wish to reduce. The model function Σ^μ of order n^μ is defined by

applying Hermite interpolation of Σ about some complex frequencies σ^μ, i.e.

$$\Sigma^\mu := \mathrm{RK}\left(\Sigma, \sigma^\mu\right), \tag{12}$$

whereby typically $N \gg n^\mu > n$.

The purpose of Σ^μ is to be a good local approximation of the FOM and allow for a faster optimization. Reduction strategies using a middle-sized surrogate model Σ^μ, similarly to what we described so far, are known as *two-step* approaches [18]. The drawback of this strategy is that the reduction quality depends heavily on the intermediate model and therefore the choice of shifts. In addition, it is generally not possible to establish any optimality or other rigorous quality measures with respect to the original problem.

For this reason, in the proposed procedure we go further and exploit the information gained from the optimization at step i with respect to the model function Σ_i^μ. In fact, the optimization of the model function indicates new frequency regions that might be relevant. We make use of this information and *update* the model function Σ_{i+1}^μ with information of the FOM about the new optimal frequencies σ_i^*. The optimization is repeated with respect to this new model function, that now represents the FOM well also locally around the new shifts. The iteration is repeated until the optimal shifts have converged, i.e. $\sigma_i^* = \sigma_{i+1}^*$.

B. Optimality of the reduced model function Σ_r^μ

What has been presented so far is up to now merely a heuristic procedure that tries to reduce the optimization cost by replacing the FOM with a local approximation. What still remains unclear is what type of properties one can expect from the resulting ROM. In the following we will prove that the ROM obtained in this procedure actually is a local $\mathcal{H}_2$-optimal approximation of the FOM and, in addition, that it is exactly the same ROM as one would obtain by reducing the FOM directly at the optimal shifts σ^*.

Lemma 1: Assume $\Sigma^\mu = \mathrm{RK}\left(\Sigma, \sigma^\mu\right)$, where σ^μ is the set of shifts used to create the model function. Further, assume that we obtained a reduced model function that minimizes the $\mathcal{H}_2$-error locally with respect to the model function, i.e. $\Sigma_r^\mu = \mathrm{RK}\left(\Sigma^\mu, \sigma^*\right)$. Then, if $\sigma^* \subseteq \sigma^\mu$, the reduced model function Σ_r^μ is a locally $\mathcal{H}_2$-optimal approximation of the FOM Σ.

Proof: The proof is obtained by showing that the Meier-Luenberger conditions in (9) are satisfied. By optimality of Σ_r^μ at σ^* we get that the reduced eigenvalues satisfy the condition $-\overline{\lambda_{r,i}^\mu} = \sigma_i^* \quad \forall i = 1, \ldots, n$. Further, the condition $\sigma^* \subseteq \sigma^\mu$ implies that the model function contains, amongst other, the information relative to a Hermite interpolation of Σ at σ^*, that is the transfer function value and its derivative. Connecting all these results completes the proof. ∎

This shows that if the model function is continuously updated with information about the new optimal shifts σ_i^* at each step, at convergence Σ_r^μ will be optimal even with respect to the original model. A trivial consequence of this is that the transfer functions $G_r(s)$ and $G_r^\mu(s)$ are the same.

Corollary 1: Consider the assumptions of Lemma 1. Then the transfer function $G_r^\mu(s)$ corresponding to the realization Σ_r^μ is the same as $G_r(s)$, i.e. the one we would obtain by approximating $G(s)$ directly.

Proof: The proof follows directly from the fact that both $G_r^\mu(s)$ and $G_r(s)$ are transfer functions of a dynamical system of order n and that by fixing $2n$ parameters we get uniqueness. Note that by (2) it is implicitly assumed that $D = D_r = D_r^\mu$. ∎

It turns out that we can even go one step further and show that the state-space representation Σ_r^μ resulting from the model function framework is *the same* as Σ_r.

Theorem 1: Assume $\Sigma^\mu = \mathrm{RK}\left(\Sigma, \sigma^\mu\right)$. Further assume that we obtained an optimal reduced model function, i.e. $\Sigma_r^\mu = \mathrm{RK}\left(\Sigma^\mu, \sigma^*\right)$. Then, if $\sigma^* \subseteq \sigma^\mu$, the realization of the reduced model function Σ_r^μ is the same as Σ_r, i.e. the realization we would obtain by directly reducing Σ at σ^*.

Proof: For brevity of exposition, this contribution will focus on the case of SISO systems and will exclude multiplicities in σ^* as well as the case of different bases for the Krylov subspaces. Note, however, that generalizations in this direction are available and will be explained in the full paper.

Recall that the input Krylov subspace used for the direct reduction of Σ about σ^* satisfies

$$AV_r - EV_r S_V^* - BT = 0, \tag{13}$$

where $\lambda_i(S_V^*) = \sigma_i^*$. Consider the Sylvester equation for V_r^μ

$$A^\mu V_r^\mu - E^\mu V_r^\mu S_V^* - B^\mu T = 0, \tag{14}$$

which can be rewritten as

$$\begin{aligned} \left(W^\mu\right)^\top \left(AV^\mu V_r^\mu - EV^\mu V_r^\mu S_V^* - BT\right) &= 0, \\ \left(W^\mu\right)^\top \left(A\widetilde{V}_r - E\widetilde{V}_r S_V^* - BT\right) &= 0. \end{aligned} \tag{15}$$

A comparison between (15) and (13) reveals that $\widetilde{V}_r = V_r$ is sufficient to satisfy (14). However, to get equivalence, we need to show also the necessity of this condition.

For this purpose, assume that the expression in the bracket in (15) does not vanish. Another way of interpreting the equation is by seeing it as a product of n^μ row vectors and n column vectors

$$\begin{pmatrix} \vdots \\ CA_{\sigma_i^\mu}^{-1} \\ \vdots \end{pmatrix} \cdot \begin{pmatrix} \cdots & A_{\sigma_j^*}\widetilde{V}_{r,j} - B & \cdots \end{pmatrix} = 0, \tag{16}$$

where $A_{\sigma_i} := A - \sigma_i E$. Now, we can use the condition $\sigma^* \subseteq \sigma^\mu$ since it implies that for every σ_j^*, we can find the row-vector i with $\sigma_i^\mu = \sigma_j^*$. For each of these pairs, the product simplifies to

$$C\left(\widetilde{V}_{r,j} - \left(A - \sigma_j^* E\right)^{-1} B\right) = 0,$$

which has to hold true for all C. Therefore, we get the equality

$$\widetilde{V}_{r,j} = \left(A - \sigma_j^* E\right)^{-1} B = V_{r,j}. \tag{17}$$

Repeating this procedure for all σ_j^* yields $V^\mu V_r^\mu = V_r$, and applying the dual argumentation yields $W^\mu W_r^\mu = W_r$, which completes the proof. ∎

C. Update of the model function

From the results we have just seen, the main condition the model function has to satisfy at convergence is $\sigma^* \subseteq \sigma^\mu$, i.e. the model function Σ^μ must contain at least the information related to a Hermite interpolation of Σ at the optimal frequencies σ^*. This suggests the following update procedures:

1) The simplest strategy is to update the model function Σ_i^μ at each step with Krylov directions about *all* frequencies σ^*. In *multiset* notation this yields $\sigma_i^\mu = \cup_{j=1}^{i} \sigma_j^*$.
2) Another strategy would update the model function only with the *new* optimal shifts not already included. In *set* notation this yields $\sigma_i^\mu = \cup_{j=1}^{i} \sigma_j^*$. This procedure makes sure the size of the model function does not grow too fast but preserves the information already computed.
3) The most lean procedure would be to keep *only* the information that is required to ensure optimality at convergence, that is $\sigma_i^\mu = \sigma_i^*$. This last option keeps the model function size to a minimum but destroys information about the FOM already computed. Note also that the model function needs to be larger than the ROM, therefore some additional information needs to be incorporated.

D. The cost of the model function strategy

One question still remains open at this point: will this procedure really reduce the arithmetic complexity of the reduction as hoped? To answer the question we study the complexity $\mathcal{C}_{N,n}^\mu(\cdot)$ of any given $\mathcal{H}_2$-optimal MOR function that reduces a model of full order N to an approximant of reduced order n with the proposed model function framework:

$$\mathcal{C}_{N,n}^\mu(\cdot) = \underbrace{\sum_{i=1}^{k^\mu} \mathcal{C}_{N,n_i^{\mu,+}}(\mathrm{RK})}_{\Sigma^\mu \text{ update}} + \underbrace{\sum_{i=1}^{k^\mu} \mathcal{C}_{n_i^\mu,n}(\cdot)}_{\text{optimization}}, \tag{18}$$

where $n_i^{\mu,+}$ represents the update size of the model function Σ^μ and n_i^μ represents the order of the respective model function at each iteration i. The total number of iterations needed until convergence is denoted by k^μ.

This representation reveals the effect of the model function approach, i.e. decoupling to cost of *reduction* (i.e. retrieving information from the FOM) and *optimization* (i.e. finding suitable frequencies). Further it *shifts* the cost from the actual MOR function—in our case the search of $\mathcal{H}_2$-optimal reduction parameters—to the update of the model function. Therefore, this strategy is expected to be particularly effective if the model function updates are significantly less than the number of reduction steps required for the $\mathcal{H}_2$-optimal search. The numerical examples of Section VI illustrate the effectiveness of this framework.

V. Confined IRKA

The proposed model function framework is general in the sense that it can be applied to *any* $\mathcal{H}_2$-optimal reduction procedure that is initialized about some complex frequencies σ and yields a locally optimal solution σ^*. In this contribution, we will show how this can be applied to the well-known IRKA procedure. Since the validity of the optimization is confined to the region of complex frequencies covered by the model function, we shall call this procedure Confined Iterative Rational Krylov Algorithm (CIRKA). The update of the model function can be conducted in several ways as discussed in Section IV-C.

Algorithm Confined IRKA (CIRKA)

Input: $\Sigma, \sigma, \mathrm{tol}$
Output: locally $\mathcal{H}_2$-optimal reduced model Σ_r

```
Initialize Σ^μ to empty
while relative change of σ > tol do
    Σ^μ ← updateModelFct(Σ, Σ^μ, σ)
    [Σ_r^μ, σ*] ← IRKA(Σ^μ, σ)
    σ ← σ*
end while
Σ_r ← Σ_r^μ
```

A comparison of the complexity between IRKA and CIRKA is given in Table I.

TABLE I
Comparison between IRKA and CIRKA

Algorithm	$\mathcal{C}_{N,n}(\cdot)$
IRKA	$k_{IRKA} \cdot \mathcal{C}_{N,n}(\mathrm{RK})$
CIRKA	$\sum_{i=1}^{k^\mu} \mathcal{C}_{N,n_i^{\mu,+}}(\mathrm{RK}) + \sum_{i=1}^{k^\mu} k_{IRKA,i} \cdot \mathcal{C}_{n_i^\mu,n}(\mathrm{RK})$

Assuming $\mathcal{C}_{n_i^\mu,n}(\mathrm{RK}) \ll \mathcal{C}_{N,n}(\mathrm{RK})$ we see that the cost is shifted from the optimization to the model function update. This is illustrated also in Figure 2, where IRKA and CIRKA are compared in terms of their advantage, with respect to BT, over the number of IRKA iterations[1].

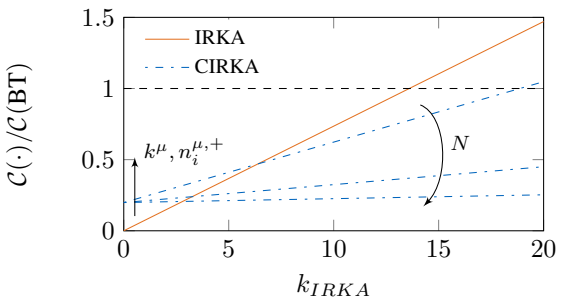


Fig. 2. Comparison between IRKA and CIRKA for different full orders.

[1]The plot was generated assuming dense direct algorithms for both LSE and LyapChol, as well as following parameters: $n = 10$, $n_0^\mu = 12$, $n_i^{\mu,+} = 5$, $k^\mu = 4$. For CIRKA, the full orders chosen are $N = 40, 60, 100$.

It becomes evident that the influence of a high number of steps during the optimization becomes negligible especially for higher full orders. On the other hand, the number of steps CIRKA requires until convergence and the number of model function updates become the main cost drivers. As a general rule of thumb, the proposed method is effective as long as

$$\sum_{i=1}^{k^\mu} n_i^{\mu,+} < n \cdot k_{IRKA}. \tag{19}$$

VI. Numerical examples

As a proof of concept, this section shows some reduction results obtained with CIRKA in comparison to IRKA on different benchmark models [17], [19]. The results, which are summarized in Table II[2], show the effective speedup of the model function framework.

TABLE II
Numerical results with IRKA and CIRKA ($n = 10$).

beam ($N=348$)	$\sum_{i=1}^{k^\mu} k_{IRKA}$	k^μ	$n_{k^\mu}^\mu$	t_{red}/s
IRKA	157	-	-	19.5
CIRKA	143	7	30	6.2
speedup				**3.2**
eady ($N=598$)	$\sum_{i=1}^{k^\mu} k_{IRKA}$	k^μ	$n_{k^\mu}^\mu$	t_{red}/s
IRKA	21	-	-	11.1
CIRKA	34	4	32	2.4
speedup				**4.7**
rail ($N=79841$)	$\sum_{i=1}^{k^\mu} k_{IRKA}$	k^μ	$n_{k^\mu}^\mu$	t_{red}/s
IRKA	15	-	-	104.4
CIRKA	10	1	12	3.9
speedup				**26.9**

VII. Conclusions

In this contribution, the arithmetic complexity of $\mathcal{H}_2$-optimal rational Krylov algorithms has been analyzed. It has been discussed that the computational expense can become higher than the one of direct balancing methods, depending on the amount of optimization steps required until convergence. In order to overcome this bottleneck, we have proposed a general framework that uses *model functions*, i.e. local approximations of the FOM, to speedup the optimization. Through this framework, the cost of *reduction* (obtaining information from the original model) is separated from the cost of *optimization* (finding suitable reduction frequencies), which becomes negligible especially for high full orders. It has been shown that if the model function is appropriately updated throughout the procedure, optimality with respect to the original model is preserved. Finally, a faster version of IRKA, namely *confined IRKA* was proposed. First numerical examples have demonstrated the effective speedup.

[2]All computations were conducted using MATLAB® R2015b on an Intel® Core™ i7-2640M CPU @2.80GHz with 8.00 GB RAM.

Appendix I
MATLAB code for CIRKA

The MATLAB functions `cirka` and a more general `modelFctMor`, which applies the model function framework to *any* $\mathcal{H}_2$-optimal method, will be included in the release v1.05 of *sssMOR*, a free and open-source MATLAB toolbox for the analysis and reduction of large-scale dynamical systems, available for download at `www.rt.mw.tum.de/?sssMOR` [20].

Acknowledgements

The authors would like to thank the reviewers for the constructive comments on how to improve section III.

References

[1] Bruce C. Moore. Principal component analysis in linear systems: controllability, observability and model reduction. *IEEE Transactions on Automatic Control*, AC-26:17–32, 1981.

[2] A. C. Antoulas. *Approximation of Large-Scale Dynamical Systems*. SIAM, 2005.

[3] T. Penzl. Numerical solution of generalized lyapunov equations. *Advances in Computational Mathematics*, 8(1–2):33–48, 1998.

[4] Eric J. Grimme. *Krylov Projection Methods for Model Reduction*. PhD thesis, Dep. of Electrical Eng., Uni. Illinois at Urbana Champaign, 1997.

[5] Serkan Gugercin, A. C. Antoulas, and C. A. Beattie. $\mathcal{H}_2$ model reduction for large-scale linear dynamical systems. *SIAM Journal on Matrix Analysis and Applications*, 30(2):609–638, 2008.

[6] Kyle A. Gallivan, Antoine Vandendorpe, and Paul Van Dooren. Sylvester equations and projection-based model reduction. *Journal of Computational and Applied Mathematics*, 162(1):213–229, 2004.

[7] C. A. Beattie and Serkan Gugercin. A trust region method for optimal $\mathcal{H}_2$ model reduction. In *IEEE Conference on Decision and Control*, 2009.

[8] Heiko K. F. Panzer, Stefan Jaensch, Thomas Wolf, and Boris Lohmann. A greedy rational krylov method for $\mathcal{H}_2$-pseudooptimal model order reduction with preservation of stability. In *American Control Conference*, pages 5532–5537, 2013.

[9] Heiko K. F. Panzer. *Model Order Reduction by Krylov Subspace Methods with Global Error Bounds and Automatic Choice of Parameters*. PhD thesis, Technische Universität München, 2014.

[10] David Harel and Yishai A Feldman. *Algorithmics: the spirit of computing*. Pearson Education, 2004.

[11] Michel Daydé, Osni Marques, and Kengo Nakajima. *High Performance Computing for Computational Science-Vecpar 2014*. Springer, 2015.

[12] Valeria Simoncini. A new iterative method for solving large-scale lyapunov matrix equations. *SIAM Journal on Scientific Computing*, 29(3):1268–1288, 2007.

[13] Gene H Golub and Charles F Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, 1996.

[14] Danny C Sorensen and Yunkai Zhou. Direct methods for matrix sylvester and lyapunov equations. *Journal of Applied Mathematics*, 2003(6):277–303, 2003.

[15] Yousef Saad. *Iterative methods for sparse linear systems*. Society for Industrial and Applied Mathematics, 2003.

[16] Iain S Duff, Albert Maurice Erisman, and John Ker Reid. *Direct methods for sparse matrices*. Clarendon press Oxford, 1986.

[17] Younes Chahlaoui and Paul Van Dooren. A collection of benchmark examples for model reduction of linear time invariant dynamical systems. 2002.

[18] Michael Lehner and Peter Eberhard. A two-step approach for model reduction in flexible multibody dynamics. *Multibody System Dynamics*, 17(2-3):157–176, 2007.

[19] Peter Benner and Jens Saak. A semi-discretized heat transfer model for optimal cooling of steel profiles. In *Dimension Reduction of Large-Scale Systems*, pages 353–356. Springer, 2005.

[20] Alessandro Castagnotto, Maria Cruz Varona, Lisa Jeschek, and Boris Lohmann. sss & sssMOR: Analysis & reduction of large-scale dynamic systems with MATLAB. In Preparation.

A.2 A New Framework for $\mathcal{H}_2$-Optimal Model Reduction

Summary: This contribution follows up on the preliminary work in Appendix A.1 and presents in more general terms a framework by the name of "Model Function" which can be applied to different system classes for which interpolatory $\mathcal{H}_2$-optimal (or -inspired) reduction algorithms are available. After revising fundamentals of interpolatory $\mathcal{H}_2$-model reduction for linear time-invariant systems, the contribution starts with an analysis of the numerical cost attached to performing $\mathcal{H}_2$-optimal reduction with conventional methods. The numerical algorithms involved (sparse LU decomposition, QR decomposition, Eigenvalue decomposition, projection) are compared in terms of execution times for benchmark models of different orders ($10^1 - 10^5$) and structure. From this analysis, it becomes evident that the sparse LU decompositions are by far the bottleneck in interpolatory reduction, with execution times up to nearly four orders of magnitude higher than the other steps. This motivates the estimation of algorithmic complexity counting the number of LU decompositions performed. Applied to IRKA, this investigation makes evident that the cost of optimization is weighted with the full cost of reduction of the full-order model. For this reason, the Model Function framework is introduced, together with an illustrative example and information about wise choices for initialization and update. Derivations and optimality proofs are presented for the class of multiple-input, multiple-output linear time-invariant systems. The new framework is then reflected, given different interpretations, and a theoretical discussion on when it is expected to be numerically advantageous with respect to conventional $\mathcal{H}_2$-optimal reduction. In addition, first ideas on how to further exploit the Model Function, which is computed "for free" during $\mathcal{H}_2$-optimal reduction, are presented, such as the estimation of the approximation error. Numerical results show both effectiveness and limitations of the framework. The manuscript ends with first indication on how to apply the framework also to other system classes, e.g. systems of differential-algebraic equations and nonlinear systems.

Contribution(s): Derivations, analysis, software development, numerical examples and writing have been conducted predominantly by the first author.

Reference: Alessandro Castagnotto & Boris Lohmann (2018) A new framework for $\mathcal{H}_2$-optimal model reduction, Mathematical and Computer Modelling of Dynamical Systems, 24:3, 236-257, DOI: 10.1080/13873954.2018.1464030

A.3 An Approach for Globalized $\mathcal{H}_2$-Optimal Model Reduction

Summary: This contribution presents a numerically efficient approach to tackle the problem of finding the *globally* $\mathcal{H}_2$-optimal reduced-order model of prescribed order n. After revising the fundamentals of conventional $\mathcal{H}_2$-optimal model reduction, the paper starts with a numerical example motivating the need of global approaches. In fact, $\mathcal{H}_2$-optimal reduction by IRKA is started from 200 initial points in the complex plane, converging to a total of four different local minima with very different (up to three orders of magnitude) approximation qualities . For this reason, a globalized approach inspired by the general literature on global optimization is proposed. This approach requires in a first step a stochastic *global sampling* of the search space and in a second step localized reduction initiated at all samples, hence the name *globalized* local search. A brief comparison of two different approaches, namely multi-start and clustering (or path relinking) is discussed, making the former seem more appropriate to the problem at hand. In addition, as opposed to the latter, it allows parallel execution of local optimization. To increase the numerical efficiency of the proposed globalized version IRKA, some additional considerations are introduced. Most predominantly, the globalized optimization is combined with the Model Function framework of Appendix A.1 and Appendix A.2 to decouple the cost of repeated optimization from the cost of reducing the full-order model. Further, the Hilbert projection theorem is used to avoid the expensive computation of large-scale $\mathcal{H}_2$ norms and efficiently select the "best" local optimum amongst all optima found. While this approach does not guarantee that the global optimum will be found (in fact, the probability tends to 1 as the number or initial samples increases), it does make $\mathcal{H}_2$-optimal reduction less sensitive to initializations and returns the best local minimum amongst all minima found. Numerical examples demonstrate the effectiveness and efficiency of the approach, comparing it to simply running IRKA from many initial samples.

Contribution(s): Derivations and software development have been conducted by the first author with preliminary support by the second author. Analysis, numerical examples and writing have been conducted predominantly by the first author.

Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnline CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 51-2 (2018) 196–201

An Approach for Globalized $\mathcal{H}_2$-Optimal Model Reduction ⋆

Alessandro Castagnotto, Siyang Hu, Boris Lohmann

Chair of Automatic Control, Technical University of Munich, Boltzmannstr. 15, D-85748 Garching. (e-mail: {a.castagnotto, siyang.hu, lohmann}@tum.de)

Abstract In this contribution, we present a model order reduction algorithm for linear systems with multiple inputs and multiple outputs that aims at finding the *global* optimal reduced model of prescribed order n, with respect to the $\mathcal{H}_2$ norm. Our approach is based on *globalized local optimization*, which requires a global sampling of the search space and subsequent local $\mathcal{H}_2$ optimization. The increased cost resulting from repeated $\mathcal{H}_2$ optimization will be mitigated by exploiting the *Model Function* framework for $\mathcal{H}_2$-optimal model reduction, making the optimization cost negligible compared to the cost of reduction. Numerical investigations motivate the need for globalized approaches in $\mathcal{H}_2$-optimal reduction and demonstrate how our method is capable of finding global optima, at a far lower cost than running conventional $\mathcal{H}_2$-optimal reduction for different initial samples.

Keywords: Minisymposium on Model Reduction; model reduction; model approximation; large-scale systems; MIMO; global optimization;

1. INTRODUCTION

The innovation and development of technical systems have profited greatly from the advent of numerical simulations, allowing system design and analysis based on *virtual prototypes*. As the complexity of the systems and the requirements on the accuracy of the numerical models grow, so does the complexity of the mathematical models used to describe their dynamical behavior. This poses high demands on the computational resources required to evaluate the models. This is even more true if the computations are run repeatedly, e.g. during optimizations, or need to be run in real-time, e.g. in *embedded controllers* or *digital twins*. In these scenarios, it is of particular interest to find the *best* possible approximation of the original model for a given admissible complexity.

Modern model order reduction algorithms seek to automatically capture the relevant dynamics of a given model into approximate models of much smaller order, while preserving fundamental characteristics and being numerically efficient. Amongst all methods, $\mathcal{H}_2$-optimal model reduction has gained a wide interest in the past years, as it is numerically tractable for large-scale models *and* satisfies optimality conditions with respect to the $\mathcal{H}_2$ norm of the error (Gugercin et al., 2008; Van Dooren et al., 2008; Beattie and Gugercin, 2009; Panzer et al., 2013). Nonetheless, as the optimization problem is non-convex, all methods only aim at finding a *local* optimum.

In this contribution, we extend existing *local* $\mathcal{H}_2$ reduction algorithms by introducing approaches from the field of *global* optimization. The increased cost resulting from repeated local optimization will be mitigated by exploiting the *Model Function* framework for fast $\mathcal{H}_2$optimization (Castagnotto et al., 2016; Castagnotto and Lohmann, 2018). Even though, in general, there is no guarantee the *global* optimum will be found, our algorithm is able to efficiently find several local minima and determine the best one amongst them without additional large-scale operations.

The remainder of the paper is structured as follows: Section 2 briefly revises the fundamentals of $\mathcal{H}_2$-optimal model reduction by tangential interpolation. Section 3 will motivate the development of globalized approaches by analyzing the convexity of the optimization problem. Section 4 will introduce a globalized approach for $\mathcal{H}_2$-optimal reduction. Finally, Section 5 will demonstrate the effectiveness of the proposed procedure in numerical simulations, while Section 6 will present concluding remarks.

2. PRELIMINARIES

2.1 Model Reduction by Tangential Interpolation

We consider stable linear dynamical systems of the form

$$\left.\begin{aligned} E\,\dot{x}(t) &= A\,x(t) + B\,u(t) \\ y(t) &= C\,x(t) \end{aligned}\right\}\,\Sigma \tag{1}$$

where $E \in \mathbb{R}^{N\times N}$ is the regular *descriptor matrix*, $A \in \mathbb{R}^{N\times N}$ is the system matrix and $x \in \mathbb{R}^N$, $u \in \mathbb{R}^m$, $y \in \mathbb{R}^p$ $(p, m \ll N)$ represent the state, input and output vectors of the system, respectively. Σ denotes the system (1) by its state-space representation. The input-output behavior of (1) can be characterized in the frequency domain by $y(s) = G(s)u(s)$, with the rational transfer function matrix

$$G(s) := C\,(sE - A)^{-1}\,B \quad \in \mathbb{C}^{p\times m}, \tag{2}$$

obtained through Laplace transform of (1) under the assumption $x(t = 0) = 0$. The construction of a reduced-

⋆ The work related to this contribution is supported by the German Research Foundation (DFG), Grant LO408/19-1.

10.1016/j.ifacol.2018.03.034

order model from the full-order model (1) can be obtained by means of *Petrov-Galerkin* projection

$$\left.\begin{aligned} \overbrace{W^\top E V}^{E_r}\,\dot{x}_r(t) &= \overbrace{W^\top A V}^{A_r}\,x_r(t) + \overbrace{W^\top B}^{B_r}\,u(t) \\ y_r(t) &= \underbrace{C V}_{C_r}\,x_r(t) \end{aligned}\right\}\Sigma_r \tag{3}$$

where $x_r \in \mathbb{R}^n$ ($n \ll N$) represents the reduced state vector and the matrices $V, W \in \mathbb{R}^{N\times n}$ are called projection matrices. Σ_r denotes the realization of the reduced model (3) and $\Sigma_r = W^\top \Sigma V$ is short-hand notation to specify the projection matrices used.

Given a model (1), our primary goal is the approximation of the transfer behavior, represented by the transfer function: $G(s) \approx G_r(s)$. To achieve this goal, a judicious choice of projection matrices V, W needs to be made. In this contribution, we will choose V, W such that the reduced model Σ_r is a *bi-tangential Hermite interpolant* of the full model with respect to given frequencies $\sigma_i \in \mathbb{C}$ and input, resp. output, *tangential directions* $r_i \in \mathbb{C}^m$, $l_i \in \mathbb{C}^p$. The following result is based on (Gallivan et al., 2004; Beattie and Gugercin, 2014). For brevity, we introduce $A_{\sigma_i} := A - \sigma_i E$.

Theorem 1. (Bi-tangential Hermite Interpolation). Consider a full-order model Σ as in (1) with transfer function $G(s)$ and let scalar frequencies $\sigma_i \in \mathbb{C}$ and vectors $r_i \in \mathbb{C}^m$, $l_j \in \mathbb{C}^p$ be given such that A_{σ_i} is nonsingular for $i = 1, \dots, n$. Consider a reduced-order model Σ_r as in (3) with transfer function $G_r(s)$, obtained through projection $\Sigma_r = W^\top \Sigma V$.

(1) If
$$A_{\sigma_i}^{-1} B\, r_i \in \mathcal{R}(V), \qquad i = 1, \dots, n \tag{4}$$
then $G(\sigma_i)\, r_i = G_r(\sigma_i)\, r_i$.

(2) If
$$A_{\sigma_i}^{-\top} C^\top l_i \in \mathcal{R}(W), \qquad i = 1, \dots, n \tag{5}$$
then $l_i^\top \cdot G(\sigma_i) = l_i^\top \cdot G_r(\sigma_i)$.

(3) If both (4) and (5) hold, then, in addition,
$$l_i^\top G'(\sigma_i)\, r_i = l_i^\top G_r'(\sigma_i)\, r_i, \qquad i = 1, \dots, n, \tag{6}$$
where $G'(s)$ denotes the first derivative with respect to s.

Following this result, an appropriate choice for the interpolation frequencies $\{\sigma_i\}_{i=1}^n$ and tangential directions $\{r_i\}_{i=1}^n$ and $\{l_i\}_{i=1}^n$ needs to be made. As this is a non-trivial task, an automatic selection of reduction parameters, minimizing the approximation error $\|G - G_r\|$ for some chosen norm, is highly desirable.

2.2 $\mathcal{H}_2$-Optimal Model Reduction

In this contribution, we address the problem of finding an optimal reduced model of prescribed order n solving the optimization problem

$$G_r(s) = \arg \min_{\deg \widehat{G}_r = n} \|G - \widehat{G}_r\|_{\mathcal{H}_2}. \tag{7}$$

The $\mathcal{H}_2$ norm in (7) is induced by the $\mathcal{H}_2$ inner product according to (Antoulas, 2005)

$$\|G\|_{\mathcal{H}_2}^2 = \langle G,\, G\rangle_{\mathcal{H}_2} := \frac{1}{2\pi}\int_{-\infty}^{\infty} \operatorname{tr}\left(G^H(-j\omega)G(j\omega)\right) \mathrm{d}\omega. \tag{8}$$

The optimal solution of (7) has been characterized in different ways (cf. Gugercin et al. (2008) for an overview). For our discussion, necessary conditions for local $\mathcal{H}_2$ optimality in terms of bi-tangential Hermite interpolation are of particular interest (Meier and Luenberger, 1967; Gugercin et al., 2008; Van Dooren et al., 2008).

Theorem 2. (First-order optimality cond.). Consider a full order model (1) with transfer function $G(s)$. Consider a reduced-order model with transfer function $G_r(s) = \sum_{i=1}^{n} \frac{\hat{c}_i \hat{b}_i}{s - \lambda_{r,i}}$ where $\lambda_{r,i} \in \mathbb{C}$ are the reduced poles and $\hat{b}_i^\top \in \mathbb{C}^m$, $\hat{c}_i \in \mathbb{C}^p$ the input resp. output residual directions.

If $G_r(s)$ satisfies (7) locally, then

$$G(-\bar{\lambda}_{r,i})\hat{b}_i^\top = G_r(-\bar{\lambda}_{r,i})\hat{b}_i^\top \tag{9a}$$
$$\hat{c}_i^\top G(-\bar{\lambda}_{r,i}) = \hat{c}_i^\top G_r(-\bar{\lambda}_{r,i}) \tag{9b}$$
$$\hat{c}_i^\top G'(-\bar{\lambda}_{r,i})\hat{b}_i^\top = \hat{c}_i^\top G_r'(-\bar{\lambda}_{r,i})\hat{b}_i^\top \tag{9c}$$
for $i = 1, \dots, n$.

Extensions to poles with higher multiplicities can be found in (Van Dooren et al., 2010). In addition, note that by the Hilbert projection theorem, for any reduced model $G_r(s)$ satisfying (7), the expression of the $\mathcal{H}_2$ error norm can be simplified as follows.

Lemma 3. (Gugercin et al. (2008)). Consider a full-order model Σ with transfer function $G(s)$. Let $G_r(s)$ be a local minimizer of (7) and have simple poles. Then
$$\langle G - G_r,\, G_r\rangle_{\mathcal{H}_2} = 0, \tag{10}$$
and, as a consequence,
$$\|G - G_r\|_{\mathcal{H}_2} = \langle G,\, G\rangle_{\mathcal{H}_2} - \langle G_r,\, G_r\rangle_{\mathcal{H}_2}. \tag{11}$$

The question remains open on how to construct a reduced model satisfying (9). Theorem 1 specifies how to construct bi-tangential Hermite interpolants. However, in general, the eigenvalues and residual directions of the reduced model are not known a-priori. For this reason, an iterative scheme known as Iterative Rational Krylov Algorithm (IRKA) has been developed (Gugercin et al., 2008; Van Dooren et al., 2008) to adapt the interpolation data until the conditions (9) are satisfied. A sketch is given in Algorithm 1.

Algorithm 1 Iterative Rational Krylov Algorithm

Input: Σ, $\{\sigma_i\}_{i=1}^n$, $\{r_i\}_{i=1}^n$, $\{l_i\}_{i=1}^n$
Output: Σ_r satisfying (9)
1: **while** not converged **do**
2: $\quad \mathcal{R}(V) \leftarrow \mathcal{R}\left(\left[A_{\sigma_1}^{-1} B\, r_1 \;\dots\; A_{\sigma_n}^{-1} B\, r_n\right]\right)$
3: $\quad \mathcal{R}(W) \leftarrow \mathcal{R}\left(\left[A_{\sigma_1}^{-\top} C^\top l_1 \;\dots\; A_{\sigma_n}^{-\top} C^\top l_n\right]\right)$
4: $\quad \Sigma_r \leftarrow W^\top \Sigma V$
5: $\quad [X, D, Y] = \mathrm{eig}(\Sigma_r)$ *// eigendec. (cf. MATLAB)*
6: $\quad \sigma_i \leftarrow -e_i^\top \overline{D} e_i; r_i \leftarrow B_r^\top Y e_i; l_i \leftarrow C_r X e_i$
7: **end while**

2.3 Model Function Framework

Recently, a new framework to speedup $\mathcal{H}_2$-optimal reduction has been presented, based on the local nature of

model reduction by tangential interpolation. It achieves—in some sense—a decoupling of the cost of optimization (i.e. finding optimal reduction parameters) from the cost of reduction (i.e. evaluating the full-order model) (Panzer, 2014; Castagnotto et al., 2016; Castagnotto and Lohmann, 2018). This involves the definition of a surrogate model, the *Model Function* Σ_μ, a bi-tangential Hermite interpolation of the full-order model with respect to iteratively updated interpolation data. This framework can be applied to general interpolatory $\mathcal{H}_2$ reduction algorithms; In combination with IRKA, the resulting algorithm is called Confined IRKA (CIRKA), briefly sketched in Algorithm 2.

Algorithm 2 Confined IRKA

Input: Σ, $\{\sigma_i\}_{i=1}^n$, $\{r_i\}_{i=1}^n$, $\{l_i\}_{i=1}^n$
Output: Σ_r satisfying (9), Σ_μ, $\tilde{\epsilon}_{\mathcal{H}_2}$

1: **while** not converged **do**
2: $\quad \Sigma_\mu \leftarrow \text{update}\Sigma_\mu\left(\Sigma, \{\sigma_i\}_{i=1}^n, \{r_i\}_{i=1}^n, \{l_i\}_{i=1}^n\right)$
3: $\quad \left[\Sigma_r, \{\sigma_{*,i}\}_{i=1}^n, \{r_{*,i}\}_{i=1}^n, \{l_{*,i}\}_{i=1}^n\right] \leftarrow \text{IRKA}\left(\Sigma_\mu, \{\sigma_i\}_{i=1}^n, \{r_i\}_{i=1}^n, \{l_i\}_{i=1}^n\right)$
4: $\quad \{\sigma_i\}_{i=1}^n \leftarrow \{\sigma_{*,i}\}_{i=1}^n$; $\{r_i\}_{i=1}^n \leftarrow \{r_{*,i}\}_{i=1}^n$; $\{l_i\}_{i=1}^n \leftarrow \{l_{*,i}\}_{i=1}^n$
5: **end while**
6: $\tilde{\epsilon}_{\mathcal{H}_2} \leftarrow \frac{\|\Sigma_\mu - \Sigma_r\|_{\mathcal{H}_2}}{\|\Sigma_\mu\|_{\mathcal{H}_2}}$

Note how CIRKA naturally—i.e. at not additional cost—returns a middle-sized surrogate Σ_μ that can be used, e.g. for error estimation $\tilde{\epsilon}_{\mathcal{H}_2}$. Finally, note that the decoupling of the cost of reduction from the cost of optimization is what motivates us to introduce globalized approaches in the reduction of large-scale systems. In fact, as the optimization is run with respect to a middle-sized surrogate, its cost is negligible compared to the cost of evaluating the full-order model (Castagnotto et al., 2016; Castagnotto and Lohmann, 2018).

3. CONVEXITY ANALYSIS

To motivate the need of globalized approaches in $\mathcal{H}_2$-optimal reduction, we perform a numerical analysis [1], initializing IRKA at different sets of frequencies $\{\sigma_i\}_{i=1}^n$ and comparing the fixed points $\{\sigma_{*,i}\}_{i=1}^n$ found. We use the IRKA implementation of the `sssMOR` toolbox (Castagnotto et al., 2017), using the options `stopCrit = 's0'`, `tol = '1e-6'` and `maxiter = 5e2`. We investigate the benchmark model of the International Space Station (`iss`) taken from the collection (Chahlaoui and Van Dooren, 2002). The model has a full order of $N = 270$ as well as $m = p = 3$ inputs and outputs. To simplify the graphical representation of the results, we inspect the subsystem from the first input to the first output (SISO) and restrict the discussion to a reduced order $n=2$. Therefore only pairs of complex conjugated frequencies are considered, for which a representation on the first quadrant in the complex plane suffices. To find a meaningful initialization for IRKA, we inspect the spectrum of the original model, as optimal frequencies often lie close to the mirrored eigenvalues. The eigenvalues with positive imaginary part, mirrored on the imaginary axis, are shown in Figure 1, as well as the region chosen for initialization.

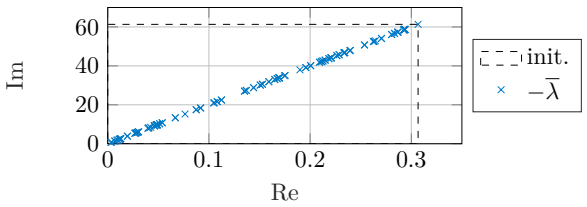


Figure 1. Mirrored spectrum and initialization region.

Figure 2 depicts all initial frequencies, as well as the respective IRKA fixed points. Small markers indicate initial points, while large markers of the same type denote the fixed points IRKA converged to.

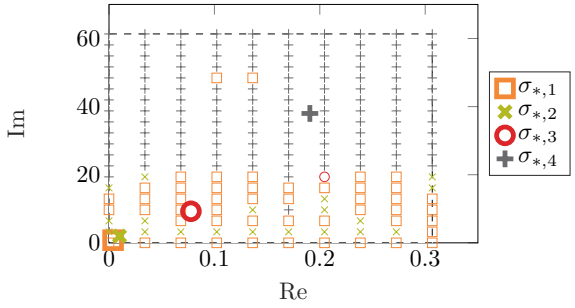


Figure 2. IRKA initialization and fixed points.

Over all, four local minima are found, confirming that the optimization problem (7) is non-convex. Compare also Table 1, where the optimal frequencies are listed, together with the relative $\mathcal{H}_2$ error $\epsilon_{\mathcal{H}_2} := \frac{\|G-G_r\|_{\mathcal{H}_2}}{\|G\|_{\mathcal{H}_2}}$ of the respective reduced model.

Table 1. Local optima and relative $\mathcal{H}_2$ errors for the SISO `iss` model.

	$\sigma_{*,1}$	$\sigma_{*,2}$	$\sigma_{*,3}$	$\sigma_{*,4}$
σ_*	$0.004 \pm 0.775i$	$0.01 \pm 1.99i$	$0.08 \pm 9.27i$	$0.19 \pm 37.99i$
$\epsilon_{\mathcal{H}_2}$	$5.8e-3$	$9.3e-1$	1.0	$8.6e-1$

The global optimal frequency is $\sigma_{*,1}$, with a relative error two orders of magnitude lower than all other minima. Out of all 200 initial points chosen, 49 converged to the global optimum (ca. 25%). Note, in particular, how the basins (regions of attraction) of the local minima are disconnected and that even optimizations started close to $\sigma_{*,1}$ may converge to one of the neighboring local minima. Finally, a comparison of the local optimal reduced models is given in terms of Bode plots in Figure 3.

Similar analyses were conducted with other benchmark models from the collection (Chahlaoui and Van Dooren, 2002). E.g. in the case of the model `building`, a similar initialization with respect to the mirrored spectrum yields the fixed points listed in Table 2. Out of the 200 initial points, 63 (ca. 32%) converged to the global optimum $\sigma_{*,1}$.

[1] All numerical simulations were run using MATLAB® R2016b on an Intel® Core™ i7-2640 CPU @ 2.80 GHz computer with 8 GB RAM.

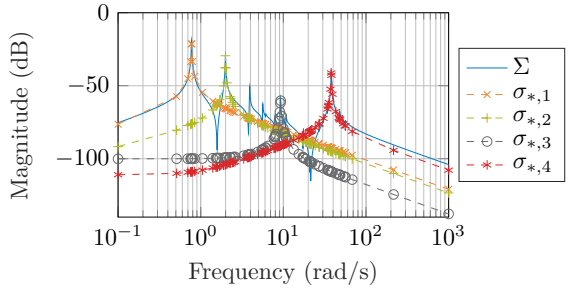


Figure 3. Bode plots of the `iss` model and local optima.

Table 2. Fixed points and relative $\mathcal{H}_2$ errors for the `building` model.

	$\sigma_{*,1}$	$\sigma_{*,2}$	$\sigma_{*,3}$
σ_*	$0.49 \pm 5.22i$	$0.68 \pm 13.74i$	$-33.64 \pm 18.89i$
$\epsilon_{\mathcal{H}_2}$	3.2e-2	7.8e-1	n.a.

However, 21 cases (ca. 10%) converged to the fixed point $\sigma_{*,3}$. As indicated by its real part (cf. Table 2), the respective reduced model is unstable. As the $\mathcal{H}_2$ norm is undefined in this case, the fixed point clearly is not a local optimum. Indeed, an unstable reduced model is not an acceptable approximation of a stable model. In all other cases (ca. 58 %), IRKA converged to the local optimum $\sigma_{*,2}$. Finally, note that for some models (cf. e.g. `beam`) IRKA shows a great robustness in finding the global optimum, converging only in a few cases to different fixed points.

From these examples, it is easy to see how the optimization problem (7) is non-convex and that, in certain cases, IRKA may not converge to the global optimum. In addition, in few cases, it may even return an unstable reduced model. Clearly, if we had knowledge of all fixed points within the search space, one could choose the best one and—by inspection of the real part—rule out instability.

4. A GLOBALIZED APPROACH

In this section, we present a globalized approach for $\mathcal{H}_2$-optimal reduction, which aims at finding the *global* optimal reduced order model of prescribed order n. In general, it is not possible to guarantee global optimality, except for some special cases. Nonetheless, there are approaches that aim at finding the global optimum by means of *globalized local optimization* (Pintér, 1991). They are based on generating a set of starting points, globally sampling the search space (*initial global search*), and subsequently performing local optimization. Amongst all local minima found, the one with best optimal value is (hopefully) the global optimum. Following this approach, our globalized $\mathcal{H}_2$-optimal reduction scheme can be divided in following steps.

4.1 Initial Global Search (Initialization of σ_i, r_i, l_i)

The first step consists in globally sampling the search space by defining k_0 sets of initial samples

$$\{\{\sigma_i\}_{i=1}^{n}, \{r_i\}_{i=1}^{n}, \{l_i\}_{i=1}^{n}\}_{j=1}^{k_0}.$$

Common approaches in literature include *grid* and *random searches*, cf. e.g. Pintér (1991). While the former guarantee global convergence as the grid size diminishes, they are computationally demanding, generally resulting in high values for k_0. For this reason, random searches are generally more efficient. This sampling can be improved by introducing information about the full-order model available at a low computational cost. For instance, computing a few eigenvalues and eigenvectors can be performed very efficiently for sparse large-scale models using power methods (Demmel, 1997). In the following, we propose two strategies:

- I.1 Compute the eigenvalues λ_{min} and λ_{max} of smallest and largest magnitude. Define a region delimited by semi-circles of radii $|\lambda_{min}|$ and $|\lambda_{max}|$ in the right half-plane and take k_0 random samples, e.g. using *normal*, *uniform* or *lognormal* distributions. Assuming no dominance of inputs and outputs is known a-priori, set all r_i, l_i to vectors $[1\ 1\ \dots\ 1]^\top$ of appropriate dimensions.
- I.2 Compute the $k_0 \cdot n$ eigenvalues of smallest magnitude $\lambda_{\mathrm{sm,i}}$ and their respective eigenvenctors $v_{\mathrm{sm,i}}$. Choose σ_i from the mirrored eigenvalues $-\overline{\lambda}_{\mathrm{sm,i}}$ and tangential directions according to $r_i = \left(v_{\mathrm{sm,i}}^\top B\right)^\top$ and $l_i = C v_{\mathrm{sm,i}}$.

In both cases, we recommend adding the origin $\sigma_i = 0$ as additional sample.

4.2 Generation of a Model Function Σ_μ

The next step in globalized optimization usually represents the local optimization for all k_0 initial samples. In our case, this requires the repeated reduction of the full model Σ through IRKA, resulting in a significant increase of reduction cost as k_0 grows. For this reason, we propose to exploit the advantages of the Model Function framework of Section 2.3 to reduce the cost of optimization by using CIRKA. Hence, before running the local optimization, we generate a middle-sized Model Function Σ_μ. As there are different possibilities for this step, we present a few most relevant.

- M.1 Generate Σ_μ as a bi-tangential Hermite interpolant of Σ with respect to *all* k_0 initial samples. This guarantees the Model Function to be a good local approximation around all initial frequencies. However, the computational cost tied to this increases as k_0 grows.
- M.2 Generate Σ_μ as a bi-tangential Hermite interpolant of Σ with respect to $k_c \in \mathbb{N}$ *centroids* of respective *k-means clusters* of all initial frequencies (Lloyd, 1982). This allows to control the cost tied to initializing Σ_μ through choice of k_c and still generates a valid local approximation in the vicinity of most starting frequencies. As tangential directions, we choose $[1\ 1\ \dots\ 1]^\top$ of appropriate dimensions.

4.3 Globalized Local Optimization

Given the Model Function Σ_μ, localized optimization can be run starting IRKA from all initial samples. As the dimension of Σ_μ is small, this step is fast and can be

performed in parallel without occurring in memory limitations. The set of all fixed points found will be denoted by $\left\{\{\sigma_{*,i}\}_{i=1}^{n}, \{r_{*,i}\}_{i=1}^{n}, \{l_{*,i}\}_{i=1}^{n}\right\}_{j=1}^{k_*}$, where $k_* \in [1, \dots, k_0]$. If a local optimization does not converge [2] or results in an unstable reduced model, the solution is disregarded.

Note: Following the discussions in literature, it is also possible to run the local optimization only for initial samples that are most promising. This approach is generally known as *scatter search* or *path relinking* [3] (Pintér, 1991). It is based on estimating the basins and running the optimization only for starting samples that are likely to lie outside. This reduces the number of local optimization being performed, cannot however be implemented in parallel. In addition, a valid estimate of the basins for $\mathcal{H}_2$ optimization—at least for IRKA—is non-trivial, as apparent from Figure 2.

4.4 Selection of the Global Optimum

Given all minima found, the question arises on how to efficiently determine which local minimum is the best. Clearly, an evaluation of (7) is too expensive, as it requires the solution of a generalized Lyapunov equation of dimension $N+n$. Fortunately, a computationally efficient method to compare the quality of different minima can be derived by using Lemma 3: The best locally optimal reduced model $G_{r,*}^{\text{gl}}(s)$, amongst all local optima $G_{r,*,1}(s), \dots, G_{r,*,k_*}(s)$, is the one with largest $\mathcal{H}_2$ norm, i.e.

$$G_{r,*}^{\text{gl}} := \arg\min_i \|G - G_{r,*,i}\|_{\mathcal{H}_2} = \arg\max_i \|G_{r,*,i}\|_{\mathcal{H}_2}. \quad (12)$$

This is easy to compute, as it only requires the solution of n-dimensional Lyapunov equations.

4.5 Update of Σ_μ and Fixed-Point Iteration

A local optimum in the reduction of Σ_μ may not be a local optimum with respect to Σ. As discussed in (Castagnotto et al., 2016; Castagnotto and Lohmann, 2018), optimality with respect to Σ can only be claimed if, after an update of Σ_μ, the optimizer does not change. For this reason, the Model Function Σ_μ needs to be updated with information of the local optima (or at least of the global optimum) and the globalized optimization needs to be repeated—starting from all local optima of the previous iteration—until convergence. Note that one may add new samples in order to increase the chances of finding new local minima. Further note that, in general, only a few full-dimensional LU decompositions are required for the update of Σ_μ.

A summary of the proposed algorithm *global CIRKA* is given in Algorithm 3. For details on implementation, we refer to the function `cirka` provided with this contribution, to be used within the `sssMOR` toolbox (Castagnotto et al., 2017). To run global CIRKA, it suffices to set the option `Opts.global = true`. In addition, a function `girka` is provided, used to run IRKA for several initial samples in parallel.

[2] Depending on the initialization and reduced order chosen, IRKA may not converge—up to a given tolerance—to a fixed point within a prescribed maximum number of iterations.

[3] Cf. the MATLAB function `GlobalSearch` distributed with the Global Optimization Toolbox™.

Algorithm 3 Global CIRKA

Input: Σ, $\left\{\{\sigma_i\}_{i=1}^{n}, \{r_i\}_{i=1}^{n}, \{l_i\}_{i=1}^{n}\right\}_{j=1}^{k_0}$
Output: (best) Σ_r satisfying (9), Σ_μ, $\tilde{\epsilon}_{\mathcal{H}_2}$

1: **while** not converged **do**
2: $\quad \Sigma_\mu \leftarrow \text{update}\Sigma_\mu\left(\Sigma, \left\{\{\sigma_i\}_{i=1}^{n}, \{r_i\}_{i=1}^{n}, \{l_i\}_{i=1}^{n}\right\}_{j=1}^{k_0}\right)$
3: $\quad$ **for** $j = 1$ to k_0 **do**
4: $\quad\quad \left[\Sigma_{r,j}, \left\{\{\sigma_{*,i}\}_{i=1}^{n}, \{r_{*,i}\}_{i=1}^{n}, \{l_{*,i}\}_{i=1}^{n}\right\}_j\right] \leftarrow \text{IRKA}\left(\Sigma_\mu, \left\{\{\sigma_i\}_{i=1}^{n}, \{r_i\}_{i=1}^{n}, \{l_i\}_{i=1}^{n}\right\}_j\right)$
5: $\quad$ **end for**
6: $\quad \Sigma_r \leftarrow \arg\max_j \|\Sigma_{r,j}\|_{\mathcal{H}_2}$
7: $\quad \left\{\{\sigma_i\}_{i=1}^{n}, \{r_i\}_{i=1}^{n}, \{l_i\}_{i=1}^{n}\right\}_{j=1}^{k_0} \leftarrow \left\{\{\sigma_{*,i}\}_{i=1}^{n}, \{r_{*,i}\}_{i=1}^{n}, \{l_{*,i}\}_{i=1}^{n}\right\}_{j=1}^{k_*}$
8: **end while**
9: $\tilde{\epsilon}_{\mathcal{H}_2} \leftarrow \frac{\|\Sigma_\mu - \Sigma_r\|_{\mathcal{H}_2}}{\|\Sigma_\mu\|_{\mathcal{H}_2}}$

5. NUMERICAL RESULTS

In this section we show the reduction results obtained with global CIRKA on the `iss` model already introduced in Section 3, taking into considerations all inputs and outputs (MIMO). In order to facilitate the depiction of results, our goal is to obtain the global optimal reduced model of order $n = 2$. We initialize global CIRKA with $k_0 = 40$ samples of frequencies and tangential directions using the initialization strategy I.2.

Figure 4 shows the initial frequencies and all minima found both for global CIRKA (gCIRKA) and IRKA run from all the inital sets (gIRKA).

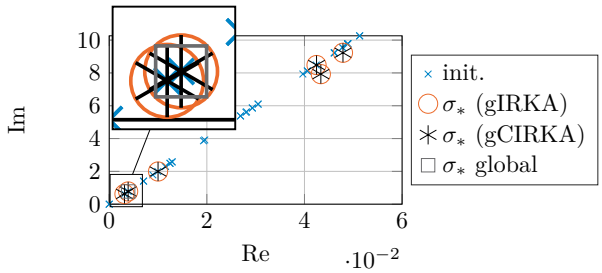


Figure 4. Optimization results with gCIRKA and gIRKA.

Table 3 lists all local optima, including the relative $\mathcal{H}_2$ errors and their estimations using the Model Function. Note that the significant quality difference between $\sigma_{*,1}$ and $\sigma_{*,6}$ results from different tangential directions to which gCIRKA converged.

Our approach gCIRKA is able to effectively find the global optimum $\sigma_{*,1}$. As it can be seen from Figure 4, gIRKA and gCIRKA find the same local minima, showing that surrogate optimization using the Model Function does not influence this results. Further, we note that in gIRKA, IRKA converged to the global optimum for only 20% of the initial points considered.

Finally, we conclude with some remarks on the computational efficiency of gCIRKA, whose cost is dominated

Table 3. Local optima, relative $\mathcal{H}_2$ errors and their estimates for the MIMO `iss` model.

	$\sigma_{*,1}$	$\sigma_{*,2}$	$\sigma_{*,3}$	$\sigma_{*,4}$	$\sigma_{*,5}$	$\sigma_{*,6}$
σ_*	$0.004\pm0.775i$	$0.01\pm1.99i$	$0.04\pm8.48i$	$0.05\pm9.23i$	$0.04\pm7.93i$	$0.003\pm0.623i$
$\epsilon_{\mathcal{H}_2}$	7.0e-01	9.4e-01	9.7e-01	9.8e-01	1.0e+00	1.0e+00
$\tilde{\epsilon}_{\mathcal{H}_2}$	7.4e-01	9.8e-01	1.0e+00	1.0e+00	1.0e+00	1.0e+00

Table 4. Cost of gCIRKA and gIRKA.

	gCIRKA	gIRKA	gCIRKA (k-means)
n_{LU}	40	376	18

by the number of large-scale LU decompositions[4] (n_{LU}) required in computing and updating the Model Function Σ_μ. In this numerical example, the Model Function was initialized with respect to all initial frequencies (M.1), leading to a total of $n_{LU}=40$. Note that bluntly running IRKA for all initial points (gIRKA) requires $n_{LU}=376$ (cf. Table 4), indicating the significant advantage of using the Model Function approach. In addition, this cost can be further reduced by k-means clustering (M.2). Figure 5 shows the result using $\sigma=0$ and $k_c=9$ centroids.

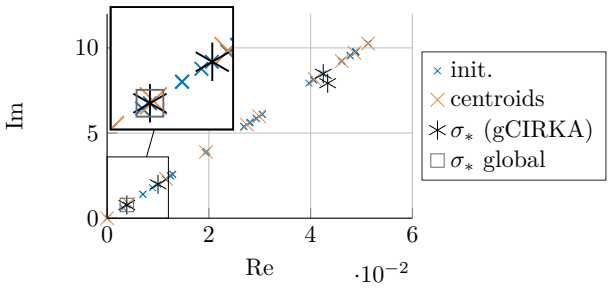


Figure 5. Results with gCIRKA using k-means clustering.

Note how in this case, not all local minima of Figure 4 are found. Nonetheless, the global optimum is successfully found.

6. CONCLUSIONS AND OUTLOOK

In this contribution, we have shown that the reduction quality of conventional $\mathcal{H}_2$-optimal reduction algorithms, such as IRKA, may heavily depend on the chosen initialization. What is more, in some cases they may even converge to *unstable* reduced models. For this reason, we have introduced an approach that, by means of globalized local optimization, aims at finding the (hopefully) global $\mathcal{H}_2$-optimal reduced model for linear systems with multiple-inputs and multiple-outputs. By exploiting the Model Function framework, localized optimization can be conducted from many initial samples in the search space at a negligible cost, compared to the cost of reducing the full-order model, thus making globalized approaches attractive and efficient also in the case of very large-scale models. The functions introduced in this contribution are available to use in combination with the `sssMOR` toolbox.

[4] Similar considerations can be conducted if iterative solvers are used instead of sparse LU decomposition.

REFERENCES

Antoulas, A. (2005). *Approximation of Large-Scale Dynamical Systems*, volume 6 of *Advances in Design and Control*. SIAM Publications, Philadelphia, PA.

Beattie, C.A. and Gugercin, S. (2009). A trust region method for optimal $\mathcal{H}_2$ model reduction. In *IEEE Conference on Decision and Control*, 5370–5375. IEEE.

Beattie, C.A. and Gugercin, S. (2014). Model reduction by rational interpolation. arXiv e-prints 1409.2140v1, Cornell University. Math.NA.

Castagnotto, A., Cruz Varona, M., Jeschek, L., and Lohmann, B. (2017). sss & sssMOR: Analysis and reduction of large-scale dynamic systems in MATLAB. *at-Automatisierungstechnik*, 65(2), 134–150.

Castagnotto, A. and Lohmann, B. (2018). A new framework for $\mathcal{H}_2$-optimal model reduction. URL `https://arxiv.org/abs/1709.07270`. Submitted.

Castagnotto, A., Panzer, H.K.F., and Lohmann, B. (2016). Fast $\mathcal{H}_2$-optimal model order reduction exploiting the local nature of Krylov-subspace methods. In *European Control Conference 2016*, 1958–1963. Aalborg, Denmark.

Chahlaoui, Y. and Van Dooren, P. (2002). A collection of benchmark examples for model reduction of linear time invariant dynamical systems. Technical Report 2002–2, SLICOT Working Note.

Demmel (1997). *Applied numerical linear algebra*. SIAM.

Gallivan, K., Vandendorpe, A., and Van Dooren, P. (2004). Model reduction of MIMO systems via tangential interpolation. *SIAM J. Matrix Anal. Appl.*, 26(2), 328–349.

Gugercin, S., Antoulas, A.C., and Beattie, C.A. (2008). $\mathcal{H}_2$ model reduction for large-scale dynamical systems. *SIAM J. Matrix Anal. Appl.*, 30(2), 609–638.

Lloyd, S. (1982). Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2), 129–137.

Meier, L. and Luenberger, D.G. (1967). Approximation of linear constant systems. *IEEE Transactions on Automatic Control*, 12(5), 585–588.

Panzer, H., Jaensch, S., Wolf, T., and Lohmann, B. (2013). A greedy rational Krylov method for H_2-pseudooptimal model order reduction with preservation of stability. In *Proceedings of the American Control Conference*, 5512–5517.

Panzer, H.K.F. (2014). *Model Order Reduction by Krylov Subspace Methods with Global Error Bounds and Automatic Choice of Parameters*. Ph.D. thesis, Technical University of Munich.

Pintér, J.D. (1991). Global optimization in action. *Scientific American*, 264, 54–63.

Van Dooren, P., Gallivan, K., and Absil, P.A. (2008). $\mathcal{H}_2$-optimal model reduction of MIMO systems. *Appl. Math. Lett.*, 21, 1267–1273.

Van Dooren, P., Gallivan, K.A., and Absil, P.A. (2010). $\mathcal{H}_2$-optimal model reduction with higher-order poles. *SIAM J. Matrix Anal. Appl.*, 31(5), 2738–2753.

A.4 Interpolatory Methods for $\mathcal{H}_\infty$ Model Reduction of Multi-Input/Multi-Output Systems

Summary: This contribution extends the "Interpolatory $\mathcal{H}_\infty$ Approximation" algorithm by Flagg, Beattie and Gugercin to linear time-invariant models with multiple inputs and multiple outputs, yielding a numerically efficient approach to obtain reduced-order models with low $\mathcal{H}_\infty$ approximation error. After revising the fundamentals of $\mathcal{H}_\infty$-optimal reduction and existing approaches for generating reduced-order models with low $\mathcal{H}_\infty$-errors, the contribution starts by motivating why rational interpolation and, in particular, $\mathcal{H}_2$-optimal reduction, is a valid approach to address also the $\mathcal{H}_\infty$ reduction problem. Conceptual discussions are accompanied by a numerical example comparing the $\mathcal{H}_\infty$-approximation error of IRKA and balanced truncation. Once a set of favorable interpolation conditions is found, the only remaining degrees of freedom in the reduced-order model are given by the entries of the feed-through matrix, which in the $\mathcal{H}_2$-optimal context is always chosen to satisfy $D = D_r$. A new proof is given on how to introduce an additional feed-through term D_r into the reduced-order model while preserving the original interpolatory conditions. A new approach called "MIMO Interpolatory $\mathcal{H}_\infty$ Approximation" (MIHA) is presented, which starts from $\mathcal{H}_2$-optimal reduction and subsequently minimizes the $\mathcal{H}_\infty$ error with respect to D_r. To increase the numerical efficiency, the Sherman-Morrison-Woodbury formula is used to separate the reduced-order model resulting from IRKA from the additional term resulting from D_r optimization. This allows the approximation of the error system (after IRKA) by means of data-driven surrogate modeling. The data is taken "for free" during IRKA iterations, while the methods chosen to generate the surrogate is vector fitting. In this way, large-scale $\mathcal{H}_\infty$-norm computations can be avoided. When using the surrogate, optimality with respect to the original error system cannot be guaranteed per se. Nonetheless, numerical examples show the effectiveness of the proposed procedure, being very close to (and sometimes even better than) optimal Hankel Norm approximations.

Contribution(s): Derivations, analysis and writing have been conducted by the first author with the support of the second and third author. Software development and numerical examples have been conducted predominantly by the first author.

Chapter 22
Interpolatory Methods for $\mathscr{H}_\infty$ Model Reduction of Multi-Input/Multi-Output Systems

Alessandro Castagnotto, Christopher Beattie, and Serkan Gugercin

Abstract We develop here a computationally effective approach for producing high-quality $\mathscr{H}_\infty$-approximations to large scale linear dynamical systems having multiple inputs and multiple outputs (MIMO). We extend an approach for $\mathscr{H}_\infty$ model reduction introduced by Flagg et al. (Syst Control Lett 62(7):567–574, 2013) for the single-input/single-output (SISO) setting, which combined ideas originating in interpolatory $\mathscr{H}_2$-optimal model reduction with complex Chebyshev approximation. Retaining this framework, our approach to the MIMO problem has its principal computational cost dominated by (sparse) linear solves, and so it can remain an effective strategy in many large-scale settings. We are able to avoid computationally demanding $\mathscr{H}_\infty$ norm calculations that are normally required to monitor progress within each optimization cycle through the use of "data-driven" rational approximations that are built upon previously computed function samples. Numerical examples are included that illustrate our approach. We produce high fidelity reduced models having consistently better $\mathscr{H}_\infty$ performance than models produced via balanced truncation; these models often are as good as (and occasionally better than) models produced using optimal Hankel norm approximation as well. In all cases considered, the method described here produces reduced models at far lower cost than is possible with either balanced truncation or optimal Hankel norm approximation.

22.1 Introduction

The accurate modeling of dynamical systems often requires that a large number of differential equations describing the evolution of a large number of state variables be integrated over time to predict system behavior. The number of state variables and

A. Castagnotto (✉)
Technical University of Munich, Garching bei München, Germany
e-mail: a.castagnotto@tum.de

C. Beattie • S. Gugercin
Virginia Tech, Blacksburg, VA 24061, USA
e-mail: beattie@vt.edu; gugercin@vt.edu

P. Benner et al. (eds.), *Model Reduction of Parametrized Systems*,
MS&A 17, DOI 10.1007/978-3-319-58786-8_22

differential equations involved can be especially large and forbidding when these models arise, say, from a modified nodal analysis of integrated electronic circuits, or more broadly, from a spatial discretization of partial differential equations over a fine grid. Most dynamical systems arising in practice can be represented at least locally around an operating point, with a state-space representation having the form

$$\begin{aligned} E\,\dot{x} &= A\,x + B\,u, \\ y &= C\,x + D\,u, \end{aligned} \tag{22.1}$$

where $E \in \mathbb{R}^{N\times N}$ is the *descriptor matrix*, $A \in \mathbb{R}^{N\times N}$ is the system matrix and $x \in \mathbb{R}^N$, $u \in \mathbb{R}^m$, and $y \in \mathbb{R}^p$ ($p, m \ll N$) represent the state, input, and output of the system, respectively. A static feed-through relation from the control input u to the control output y is modeled through the matrix $D \in \mathbb{R}^{p\times m}$. Most practical systems involve several actuators (input variables) and several quantities of interest (output variables), motivating our focus here on systems having multiple inputs and multiple outputs (MIMO).

In many application settings, the state dimension N (which typically matches the order of the model) can grow quite large as greater model fidelity is pursued, and in some cases it can reach magnitudes of 10^6 and more. Simulation, optimization, and control design based on such large-scale models becomes computationally very expensive, at times even intractable. This motivates consideration of *reduced order models* (ROMs), which are comparatively low-order models that in spite of having significantly smaller order, $n \ll N$, are designed so as to reproduce the input-output response of the full-order model (FOM) accurately while preserving certain fundamental structural properties, that may include stability and passivity. For state space models such as (22.1), reduced models are obtained generally through Petrov-Galerkin projections having the form:

$$\begin{aligned} \overbrace{W^\top E\,V}^{E_r}\,\dot{x}_r &= \overbrace{W^\top A\,V}^{A_r}\,x_r + \overbrace{W^\top B}^{B_r}\,u, \\ y_r &= \underbrace{C\,V}_{C_r}\,x_r + D_r\,u. \end{aligned} \tag{22.2}$$

The projection matrices $V, W \in \mathbb{R}^{N\times n}$ become the primary objects of scrutiny in the model reduction enterprise, since how they are chosen has a great impact on the quality of the ROM. For truly large-scale systems, *interpolatory model reduction*, which includes approaches known variously as *moment matching* methods and *Krylov subspace* methods, has drawn significant interest due to its flexibility and comparatively low computational cost [1–3]. Indeed, these methods typically require only the solution of large (generally sparse) linear systems of equations, for which several optimized methods are available. Through the appropriate selection of V and W, it is possible to match the action of the transfer function

$$G(s) = C\,(sE - A)^{-1}\,B + D \tag{22.3}$$

along arbitrarily selected *input* and *output* tangent directions at arbitrarily selected (driving) frequencies. The capacity to do this is central to our approach and is stated briefly here as:

Theorem 1 ([4, 5]) *Let $G(s)$ be the transfer function matrix (22.3) of the FOM (22.1) and let $G_r(s)$ be the transfer function matrix of an associated ROM obtained through Petrov-Galerkin projection as in (22.2). Suppose $\sigma, \mu \in \mathbb{C}$ are complex scalars ("shifts") that do not coincide with any eigenvalues of the matrix pencil (E, A) but otherwise are arbitrary. Let also $r \in \mathbb{C}^m$ and $l \in \mathbb{C}^p$ be arbitrary nontrivial tangent directions. Then*

$$G(\sigma) \cdot r = G_r(\sigma) \cdot r \quad \text{if } (A - \sigma E)^{-1} Br \in \mathsf{Ran}(V), \tag{22.4a}$$

$$l^\top \cdot G(\mu) = l^\top \cdot G_r(\mu) \quad \text{if } (A - \mu E)^{-\top} C^\top l \in \mathsf{Ran}(W), \tag{22.4b}$$

$$l^\top \cdot G'(\sigma) \cdot r = l^\top \cdot G_r'(\sigma) \cdot r \quad \text{if, additionally, } \sigma = \mu. \tag{22.4c}$$

A set of complex shifts, $\{\sigma_i\}_{i=1}^n$, $\{\mu_i\}_{i=1}^n$, with corresponding tangent directions, $\{r_i\}_{i=1}^n$, $\{l_i\}_{i=1}^n$, will be collectively referred to as *interpolation data* in our present context. We define *primitive projection matrices* as

$$\widetilde{V} := \left[(A - \sigma_1 E)^{-1} Br_1, \ldots, (A - \sigma_n E)^{-1} Br_n\right] \tag{22.5a}$$

$$\widetilde{W} := \left[(A - \mu_1 E)^{-\top} C^\top l_1, \ldots, (A - \mu_n E)^{-\top} C^\top l_n\right] \tag{22.5b}$$

Note that $\widetilde{V}$ and $\widetilde{W}$ satisfy Sylvester equations having the form:

$$A\widetilde{V} - E\widetilde{V}S_\sigma = B\widetilde{R} \quad \text{and} \quad A^\top \widetilde{W} - E^\top \widetilde{W} S_\mu^\top = C^\top \widetilde{L}, \tag{22.6}$$

where $S_\sigma = \mathrm{diag}\,(\sigma_1, .., \sigma_n) \in \mathbb{C}^{n\times n}$, $S_\mu = \mathrm{diag}\,(\mu_1, .., \mu_n) \in \mathbb{C}^{n\times n}$, $\widetilde{R} = [r_1, .., r_n] \in \mathbb{C}^{m\times n}$ and $\widetilde{L} = [l_1, \ldots, l_n] \in \mathbb{C}^{p\times n}$ [6]. In this way, the Petrov-Galerkin projection of (22.2) is parameterized by interpolation data and the principal task in defining interpolatory models then becomes the judicious choice of shifts and tangent directions.

Procedures have been developed over the past decade for choosing interpolation data that yield reduced models, $G_r(s)$, that minimize, at least locally the approximation error, $G(s) - G_r(s)$, as measured with respect to the $\mathscr{H}_2$-norm:

$$\|G - G_r\|_{\mathscr{H}_2} := \sqrt{\frac{1}{2\pi}\int_{-\infty}^{\infty} \|G(j\omega) - G_r(j\omega)\|_F^2 \, d\omega} \tag{22.7}$$

(see [1]). Minimizing the $\mathscr{H}_2$-error, $\|G - G_r\|_{\mathscr{H}_2}$, is of interest through the immediate relationship this quantity bears with the induced system response error:

$$\|y - y_r\|_{\mathscr{L}_\infty} \le \|G - G_r\|_{\mathscr{H}_2} \, \|u(t)\|_{\mathscr{L}_2}, \tag{22.8}$$

A well-known approach to accomplish this that has become popular at least in part due to its simplicity and effectiveness is the *Iterative Rational Krylov Algorithm* (IRKA) [7], which, in effect, runs a simple fixed point iteration aimed at producing interpolation data that satisfy first-order $\mathscr{H}_2$-optimality conditions, i.e.,

$$G(-\lambda_i)\cdot\hat{b}_i = G_r(-\lambda_i)\cdot\hat{b}_i, \quad \hat{c}_i^\top\cdot G(-\lambda_i) = \hat{c}_i^\top\cdot G_r(-\lambda_i), \tag{22.9a}$$

$$\text{and} \quad \hat{c}_i^\top\cdot G'(-\lambda_i)\cdot\hat{b}_i = \hat{c}_i^\top\cdot G_r'(-\lambda_i)\cdot\hat{b}_i. \tag{22.9b}$$

for $i = 1,\ldots,n$. The data λ_i, $\hat{b}_i$ and $\hat{c}_i$ are reduced poles and right/left vector residues corresponding to the pole-residue expansion of the ROM:

$$G_r(s) = \sum_{i=1}^{n}\frac{\hat{c}_i\,\hat{b}_i^\top}{s-\lambda_i}. \tag{22.10}$$

Despite the relative ease with which $\mathscr{H}_2$-optimal reduced models can be obtained, there are several circumstances in which it might be preferable to obtain a ROM which produces a small error as measured in the $\mathscr{H}_\infty$-norm:

$$\|G-G_r\|_{\mathscr{H}_\infty} := \max_{\omega}\varsigma_{max}(G(j\omega)-G_r(j\omega)), \tag{22.11}$$

where $\varsigma_{max}(M)$ denotes the largest singular value of a matrix M (see [1]). ROMs having small $\mathscr{H}_\infty$-error produce an output response with a uniformly bounded "energy" error:

$$\|y-y_r\|_{\mathscr{L}_2} \le \|G-G_r\|_{\mathscr{H}_\infty}\,\|u\|_{\mathscr{L}_2}. \tag{22.12}$$

The $\mathscr{H}_\infty$ norm is also used as a robustness measure for closed-loop control systems and is therefore of central importance in robust control. It finds frequent use in aerospace applications, among others, where the $\mathscr{L}_2$ energy of the system response is of critical interest in design and optimization.

Strategies for producing reduced models that give good $\mathscr{H}_\infty$ performance has long been an active area of research [8]. Analogous to the $\mathscr{H}_\infty$-control design problem, the optimal $\mathscr{H}_\infty$ reduction problem can be formulated in terms of *linear matrix inequalities*, although advantageous features such as linearity and convexity are lost in this case [9, 10]. Due to the high cost related to solving these matrix inequalities, this approach is generally not feasible in large-scale settings.

Another family of methods for the $\mathscr{H}_\infty$ reduction problem relates it to the problem of finding an *optimal Hankel norm approximation* (OHNA) [11–13]. Along these lines the *balanced truncation* (BT) algorithm yields rigorous upper bounds on the $\mathscr{H}_\infty$ error and often produces small approximation error, especially for higher reduced order approximants [1, 14]. Each of these procedures is generally feasible only for mid-size problems since either an all-pass dilation requiring large-scale eigenvalue decomposition (for OHNA) or the solution of generalized Lyapunov

equations (for BT) is required. Extensions to large-scale models are available, however—e.g., in [15–20].

A wholly different approach to the $\mathscr{H}_\infty$ model reduction problem for SISO models was proposed by Flagg, Beattie, and Gugercin in [21]. A locally $\mathscr{H}_2$-optimal reduced model is taken as a starting point and adjusted through the variation of rank-one modifications parameterized by the scalar feed-through term, D. Minimization of the $\mathscr{H}_\infty$-error with respect to this parameterization available through D produces ROMs that are observed to have generally very good $\mathscr{H}_\infty$-performance, often exceeding what could be attained with OHNA.

In this work, we extend these earlier interpolatory methods to MIMO systems. We introduce a strategy that reduces the computational expense of the intermediate optimization steps by means of data-driven MOR methods (we use *vector fitting* [22, 23]). Stability of the reduced model is guaranteed through appropriate constraints in the resulting multivariate optimization problem. Numerical examples show effective reduction of approximation error, often outperforming both OHNA and BT.

22.2 MIMO Interpolatory $\mathscr{H}_\infty$-Approximation (MIHA)

In this section we first characterize the $\mathscr{H}_\infty$-optimal reduced order models from the perspective of rational interpolation. This motivates the usage of $\mathscr{H}_2$-optimal reduction as a starting point for the model reduction algorithm we propose for the $\mathscr{H}_\infty$ approximation problem.

22.2.1 Characterization of $\mathscr{H}_\infty$-Approximants via Rational Interpolation

In the SISO case, Trefethen [13] has characterized best $\mathscr{H}_\infty$ approximations within a broader context of rational interpolation:

Theorem 2 (Trefethen [13]) *Suppose $G(s)$ is a (scalar-valued) transfer function associated with a SISO dynamical system as in (22.3). Let $\widehat{G}_r(s)$ be an optimal $\mathscr{H}_\infty$ approximation to $G(s)$ and let G_r be any nth order stable approximation to $G(s)$ that interpolates $G(s)$ at $2n+1$ points in the open right half-plane. Then*

$$\min_{\omega\in\mathbb{R}} |G(j\omega) - G_r(j\omega)| \le \|G - \widehat{G}_r\|_{\mathscr{H}_\infty} \le \|G - G_r\|_{\mathscr{H}_\infty}$$

In particular, if $|G(j\omega) - G_r(j\omega)| = \mathsf{const}$ for all $\omega \in \mathbb{R}$ then G_r is itself an optimal $\mathscr{H}_\infty$-approximation to $G(s)$.

For the SISO case, a good $\mathscr{H}_\infty$ approximation will be obtained when the modulus of the error, $|G(s) - G_r(s)|$, is nearly constant as $s = j\omega$ runs along the imaginary axis. In the MIMO case, the analogous argument becomes more technically involved

as the maximum singular value of matrix-valued function $G(s) - G_r(s)$ will not generally be analytic in the neighborhood of the imaginary axis (e.g., where multiple singular values occur). Nonetheless, the intuition of the SISO case carries over to the MIMO case, as the following *Gedankenexperiment* might suggest: Suppose that $\widehat{G}_r$ is an $\mathscr{H}_\infty$-optimal interpolatory approximation to G but $\varsigma_{max}(G(j\omega) - G_r(j\omega))$ is not constant with respect to $\omega \in \mathbb{R}$. Then there exist frequencies $\hat{\omega}$ and $\tilde{\omega} \in \mathbb{R}$ and $\epsilon > 0$ such that

$$\begin{aligned}\|G - \widehat{G}_r\|_{\mathscr{H}_\infty} = \varsigma_{max}(G(j\hat{\omega}) - \widehat{G}_r(j\hat{\omega})) &\geq \epsilon + \min_{\omega} \varsigma_{max}(G(j\omega) - \widehat{G}_r(j\omega)) \\ &= \epsilon + \varsigma_{max}(G(j\tilde{\omega}) - \widehat{G}_r(j\tilde{\omega})).\end{aligned}$$

By nudging interpolation data away from the vicinity of $\tilde{\omega}$ and toward $\hat{\omega}$ while simultaneously nudging the poles of $\widehat{G}_r$ away from the vicinity of $\hat{\omega}$ and toward $\tilde{\omega}$, one may decrease the value of $\varsigma_{max}(G(j\hat{\omega}) - \widehat{G}_r(j\hat{\omega}))$ while increasing the value of $\varsigma_{max}(G(j\tilde{\omega}) - \widehat{G}_r(j\tilde{\omega}))$. This will (incrementally) decrease the $\mathscr{H}_\infty$ norm and bring the values of $\varsigma_{max}(G(j\hat{\omega}) - \widehat{G}_r(j\hat{\omega}))$ and $\varsigma_{max}(G(j\tilde{\omega}) - \widehat{G}_r(j\tilde{\omega}))$ closer together toward a common value.

Of course, the nudging process described above contains insufficient detail to suggest an algorithm, and indeed, our approach to this problem follows a somewhat different path, a path that nonetheless uses the guiding heuristic for (near) $\mathscr{H}_\infty$-optimality:

$$\varsigma_{max}(G(j\omega) - \widetilde{G}_r(j\omega)) \approx \mathsf{const} \quad \text{for all } \omega \in \mathbb{R}. \tag{22.13}$$

Approximations with good $\mathscr{H}_\infty$ performance should have an advantageous configuration of poles and interpolation data that locates them symmetrically about the imaginary axis, thus balancing regions where $\varsigma_{max}(G(s) - \widetilde{G}_r(s))$ is big (e.g., pole locations) symmetrically against regions reflected across the imaginary axis where $\varsigma_{max}(G(s) - \widetilde{G}_r(s))$ is small (e.g., interpolation locations). This configuration of poles and interpolation data, we note, is precisely the outcome of optimal $\mathscr{H}_2$ approximation as well, and this will provide us with an easily computable approximation that is likely to have good $\mathscr{H}_\infty$ performance.

22.2.2 $\mathscr{H}_\infty$ Approximation with Interpolatory $\mathscr{H}_2$-Optimal Initialization

Local $\mathscr{H}_2$-optimal ROMs are often observed to give good $\mathscr{H}_\infty$ performance—this is in addition to the expected good $\mathscr{H}_2$ performance. This $\mathscr{H}_\infty$ behaviour is illustrated in Fig. 22.1, where the $\mathscr{H}_\infty$ approximation errors of local $\mathscr{H}_2$-optimal ROMs produced by IRKA are compared to ROMs of the same order obtained through BT for the CD player MIMO benchmark model [24].

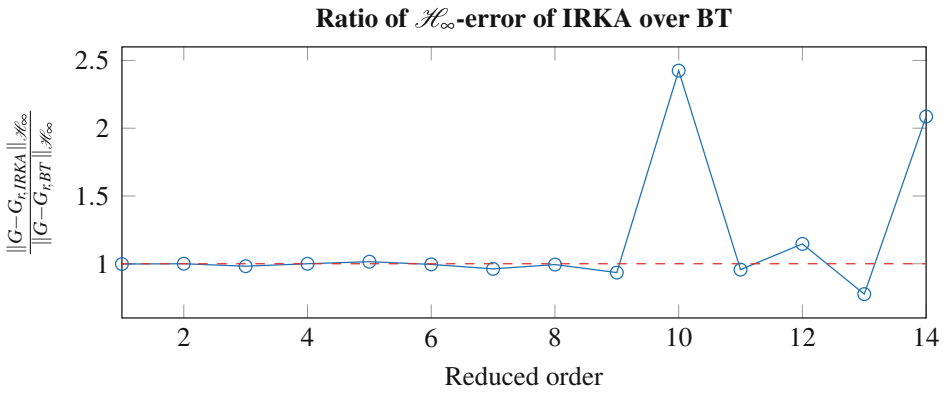


Fig. 22.1 Numerical investigations indicate that IRKA models are often good also in terms of the $\mathscr{H}_\infty$-error

The frequently favourable $\mathscr{H}_\infty$ behaviour of IRKA models has particular significance in this context, since they are computationally cheap to obtain even in large-scale settings, indeed often they are much cheaper than comparable BT computations. The resulting locally $\mathscr{H}_2$-optimal ROMs can be further improved (with respect to $\mathscr{H}_\infty$ error) by relaxing the (implicit) interpolation constraint at ∞ while preserving the $\mathscr{H}_2$-optimal interpolation conditions (which is the most important link the $\mathscr{H}_2$-optimal ROM has with the original model).

Consider the partial fraction expansion

$$G_r(s) = \sum_{i=1}^{N} \frac{\hat{c}_i \hat{b}_i^\top}{s - \lambda_i} + D_r. \tag{22.14}$$

For ease of exposition, we assume the poles, λ_i, to be simple, although the results we develop here can be extended to the case of higher multiplicity. The input/output behavior is determined by n scalar parameters λ_i, n pairs of input/output residuals $\hat{b}_i, \hat{c}_i$ and the $p \times m$-dimensional feed-through D_r. Considering that a constant scaling factor can be arbitrarily defined in the product of the residuals, this leaves us a total of $n\,(p+m) + p \cdot m$ parameters, $n\,(p+m)$ of which can be described in terms of two-sided tangential interpolation conditions (22.4). This interpolation data is established for the original $\mathscr{H}_2$-optimal ROM and we wish it to remain invariant over subsequent adjustments, so the only remaining degrees-of-freedom are the $p \cdot m$ entries in the feed-through matrix D_r.

In the typical context of $\mathscr{H}_2$-optimal model reduction, D_r is chosen to match the feed-through term D of the original model, thus guaranteeing that the error $G - \widetilde{G}_r$ remains in $\mathscr{H}_2$. Note that D remains untouched by the state-space projections in (22.2), moreover since typically $p, m \ll N$, the feed-through term need not be involved in the reduction process and may be retained from the FOM. Indeed, retaining the original feed-through term is a necessary condition for $\mathscr{H}_2$ optimality, forcing interpolation at $s = \infty$ and as a consequence, small error at higher frequencies. Contrasting significantly with $\mathscr{H}_2$-based model reduction, good $\mathscr{H}_\infty$ performance does not require $D_r = D$, and in this work we exploit this flexibility

in a crucial way. A key observation playing a significant role in what follows was made in [25, 26] that the feed-through term D_r induces a parametrization of all reduced order models satisfying the two-sided tangential interpolation conditions. This result is summarized by following theorem taken from [25, Theorem 4.1] and [26, Theorem 3]

Theorem 3 *Let $\widetilde{R}$, $\widetilde{L}$ be defined through the Sylvester equations in (22.6). Assume, without loss of generality, that the full order model satisfies $D = 0$ and let the nominal reduced model $G_r^0(s) = C_r\left(sE_r - A_r\right)^{-1} B_r$ be obtained through Petrov-Galerkin projection using the primitive projection matrices (22.5). Then, for any $D_r \in \mathbb{C}^{p\times m}$, the perturbed reduced order model*

$$\widetilde{G}_r^D(s, D_r) = \left(\widetilde{C}_r + D_r\widetilde{R}\right)\left[s\widetilde{E}_r - \left(\widetilde{A}_r + \widetilde{L}^\top D_r\widetilde{R}\right)\right]^{-1}\left(\widetilde{B}_r + \widetilde{L}^\top D_r\right) + D_r \tag{22.15}$$

also satisfies the tangential interpolation conditions (22.4).

Note that for $D \neq 0$, the results of Theorem 3 can be trivially extended by adding D to the right-hand side in (22.15). Even though for theoretical consideration the use of primitive Krylov bases $\widetilde{V}$, $\widetilde{W}$ introduced in (22.5) is often convenient, from a numerical standpoint there are several reason why one may choose a different basis for the projection matrices. This next result shows that the interpolation conditions are preserved also for arbitrary bases—in particular also real and orthonormal bases—provided that the shifting matrices R and L are appropriately chosen.

Corollary 1 *Let $T_v, T_w \in \mathbb{C}^{n\times n}$ be invertible matrices used to transform the primitive bases $\widetilde{V}$, $\widetilde{W}$ of the Krylov subspace to new bases $V = \widetilde{V}T_v$ and $W = \widetilde{W}T_w$. Let the same transformation be applied to the matrices of tangential directions, resulting in $R = \widetilde{R}T_v$ and $L = \widetilde{L}T_w$. Then, for any D_r, the ROM G_r^D is given by*

$$G_r^D(s, D_r) = \underbrace{\left(C_r + D_rR\right)}_{C_r^D}\left[sE_r - \underbrace{\left(A_r + L^\top D_rR\right)}_{A_r^D}\right]^{-1}\underbrace{\left(B_r + L^\top D_r\right)}_{B_r^D} + D_r \tag{22.16}$$

Proof The proof amounts to showing that the transfer function matrix G_r^D of the ROM is invariant to a change of basis from $\widetilde{V}$ and $\widetilde{W}$ as long as $\widetilde{R}$ and $\widetilde{L}$ are adapted accordingly.

$$\begin{aligned}
&G_r^D - D_r = C_r^D\left(sE - A_r^D\right)^{-1} B_r^D \\
&= \left(CV + D_rR\right)\left[sW^\top EV - W^\top AV - L^\top D_rR^\top\right]^{-1}\left(W^\top B + L^\top D_r\right) \\
&= \left(C\widetilde{V} + D_r\widetilde{R}\right)T_v\left[T_w^\top\left(s\widetilde{W}^\top E\widetilde{V} - \widetilde{W}^\top A\widetilde{V} - \widetilde{L}^\top D_r\widetilde{R}^\top\right)T_v\right]^{-1} T_w^\top\left(\widetilde{W}^\top B + \widetilde{L}^\top D_r\right) \\
&= \left(C\widetilde{V} + D_r\widetilde{R}\right)\left[s\widetilde{W}^\top E\widetilde{V} - \widetilde{W}^\top A\widetilde{V} - \widetilde{L}^\top D_r\widetilde{R}^\top\right]^{-1}\left(\widetilde{W}^\top B + \widetilde{L}^\top D_r\right) \\
&= \widetilde{G}_r^D - D_r\,.
\end{aligned}$$

The results of Theorem 3 generalize to the case of arbitrary bases. Following the notation from [25, Definition 2.1], the state-space models resulting from Petrov-Galerkin projections with V, W and $\widetilde{V}, \widetilde{W}$ respectively are *restricted system equivalent*. As a consequence, they share the same transfer function matrix.

Using the Sherman-Morrison-Woodbury formula [27] for the inverse of rank k perturbations of a matrix, we are able to decompose the transfer function of the shifted reduced model into the original reduced model and an additional term.

Corollary 2 *Define the auxiliary variable $\mathscr{K}_r := sE_r - A_r$. The transfer function of the shifted reduced model G_r^D can be given as*

$$G_r^D(s) = G_r^0(s) + \Delta G_r^D(s, D_r), \tag{22.17}$$

where G_r^0 is the transfer function of the unperturbed model and ΔG_r^D is defined as

$$\begin{aligned} \Delta G_r^D &= \Delta_1 + \Delta_2 + \Delta_3 \cdot (\Delta_4)^{-1} \cdot \Delta_2 + D_r \\ &\quad \textit{given} \\ \Delta_1 &:= C_r \mathscr{K}_r^{-1} L^\top D_r & \Delta_2 &:= D_r R \mathscr{K}_r^{-1} \left(B_r + L^\top D_r\right) \\ \Delta_3 &:= \left(C_r + D_r R\right) \mathscr{K}_r^{-1} L^\top & \Delta_4 &:= I - D_r R \mathscr{K}_r^{-1} L^\top \end{aligned} \tag{22.18}$$

Proof Note that by the Sherman-Morrison-Woodbury formula, following equality holds:

$$\left(\mathscr{K}_r - L^\top D_r R\right)^{-1} = \mathscr{K}_r^{-1} + \mathscr{K}_r^{-1} L^\top \left(I - D_r R \mathscr{K}_r^{-1} L^\top\right)^{-1} D_r R \mathscr{K}_r^{-1}. \tag{22.19}$$

Using this relation in the definition of G_r^D, the proof is completed by straightforward algebraic manipulations.

We proceed by attempting to exploit the additional degrees-of-freedom available in D_r to trade off excessive accuracy at high frequencies for improved approximation in lower frequency ranges, as measured with the $\mathscr{H}_\infty$-norm. We first obtain an $\mathscr{H}_2$-optimal ROM by means of IRKA and subsequently minimize the $\mathscr{H}_\infty$-error norm with respect to the constant feed-through matrix D_r while preserving tangential interpolation and guaranteeing stability. The resulting ROM G_r^* will represent a local optimum out of the set of all stable ROMs satisfying the tangential interpolation conditions. The outline of our proposed reduction procedure, called *MIMO interpolatory $\mathscr{H}_\infty$-approximation* (MIHA), is given in Algorithm 1.

Numerical results in Sect. 22.3 will show the effectiveness of this procedure in further reducing the $\mathscr{H}_\infty$-error for a given IRKA model. However, at this stage the optimization in Step 2 appears problematic, for it requires both the computation of the $\mathscr{H}_\infty$-norm of a large-scale model and a constrained multivariate optimization of a non-convex, non-smooth function. It turns out that both of these issues can be resolved effectively, as it will be discussed in the following sections.

Algorithm 1 MIMO interpolatory $\mathscr{H}_\infty$-approximation (MIHA)

Input: $G(s)$, n
Output: Stable, locally optimal reduced order model G_r^*, approximation error $e^*_{\mathscr{H}_\infty}$
1: $G_r^0 \leftarrow \text{IRKA}(G(s), n)$
2: $D_r^* \leftarrow \arg\min_{D_r} \left\| G(s) - G_r^D(s, D_r) \right\|_{\mathscr{H}_\infty}$ s.t. $G_r^D(s, D_r^*)$ is stable
3: $G_r^* \leftarrow G_r^D(s, D_r^*)$
4: $e^*_{\mathscr{H}_\infty} \leftarrow \left\| G(s) - G_r^*(s) \right\|_{\mathscr{H}_\infty}$

22.2.3 Efficient Implementation

As we have noted, the main computational burden of the algorithm described above resides mainly in Step 2. We are able to lighten this burden somewhat through judicious use of (22.17) and by taking advantage of previously computed transfer function evaluations.

22.2.3.1 A "Free" Surrogate Model for the Approximation Error $G - G_r^0$

Step 1 of Algorithm 1 requires performing $\mathscr{H}_2$-optimal reduction using IRKA. This is a fixed point iteration involving a number of steps k before convergence is achieved. At every step j, Hermite tangential interpolation about some complex frequencies $\{\sigma_i\}_{i=1}^n$ and tangential directions $\{r_i\}_{i=1}^n$, $\{l_i\}_{i=1}^n$ is performed. For this purpose, the projection matrices in (22.5) are computed, and it is easy to see that for all $i = 1, \ldots, n$ it holds

$$C \cdot \widetilde{V} e_i = C \left(A - \sigma_i E\right)^{-1} B r_i = G(\sigma_i) r_i \tag{22.20a}$$

$$e_i^\top \widetilde{W}^\top \cdot B = l_i^\top C \left(A - \sigma_i E\right)^{-1} B = l_i^\top G(\sigma_i) \tag{22.20b}$$

$$e_i^\top \widetilde{W}^\top E \widetilde{V} e_i = l_i^\top \left(A - \sigma_i E\right)^{-1} E \left(A - \sigma_i E\right)^{-1} r_i = l_i^\top G'(\sigma_i) r_i \tag{22.20c}$$

Observe that, at basically no additional cost, we can gather information about the FOM while performing IRKA. Figure 22.2a illustrates this point by showing the development of the shifts during the IRKA iterations reducing the CDplayer benchmark model to a reduced order $n = 10$. For all complex frequencies indicated by a marker, tangent data for the full order model is collected.

To use this "free" data, there are various choices for "data-driven" procedures that produce useful rational approximations. *Loewner methods* [25, 28–30] are effective and are already integrated into IRKA iteration strategies [31]. We adopt here a *vector fitting* strategy [22, 23, 32–34] instead. This allows us to produce stable low-order approximations of the reduction error after IRKA

$$\widetilde{G_e^0} \approx G_e^0 := G - G_r^0. \tag{22.21}$$

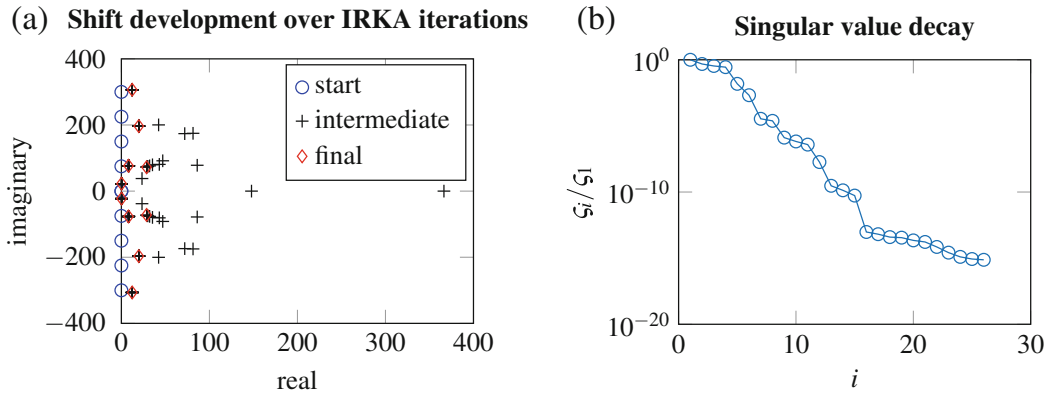


Fig. 22.2 Data collecting during IRKA can be used to generate data-driven surrogates. (**a**) Points at which data of the FOM is collected during IRKA. (**b**) Decay of singular values of the matrix $[\mathbb{L}, \sigma\mathbb{L}]$ for the data collected during IRKA

An appropriate choice of order for the surrogate model can be obtained by forming the Loewner $\mathbb{L}$ and shifted Loewner $\sigma\mathbb{L}$ matrices from G and G' evaluations that were generated in the course of the IRKA iteration and then observing the singular value decay of the matrix $[\mathbb{L}, \sigma\mathbb{L}]$, as indicated in Fig. 22.2b.

Using the decomposition in (22.17), the $\mathscr{H}_\infty$-norm evaluations required during the optimization will be feasible even for large-scale full order models. In addition, it will allow us to obtain a cheap estimate $\tilde{e}_{\mathscr{H}_\infty}$ for the approximation error

$$e_{\mathscr{H}_\infty} := \left\| G - G_r^D \right\|_{\mathscr{H}_\infty} \approx \left\| \widetilde{G_e^0} - \Delta G_r^D \right\|_{\mathscr{H}_\infty} = \tilde{e}_{\mathscr{H}_\infty} \tag{22.22}$$

22.2.3.2 Constrained Multivariate Optimization with Respect to D_r

The focus of this work lies in the development of new model reduction strategies. Our intent is not directed toward making a contribution to either the theory or practice of numerical optimization and we are content in this work to use standard optimization approaches. In the results of Sect. 22.3, we rely on state-of-the-art algorithms that are widespread and available, e.g., in MATLAB. With that caveat understood, we do note that the constrained multivariate optimization over the reduced feed-through, D_r, is a challenging optimization problem, so we will explain briefly the setting that seems to work best in our case. The computation and optimization of $\mathscr{H}_\infty$-norms for large-scale models remains an active area of research, as demonstrated by Mitchell and Overton [35, 36] and Aliyev et al. [37].

The problem we need to solve in step 2 of Algorithm 1 is

$$\begin{aligned} &\min_{D_r \in \mathbb{R}^{p\times m}} \max_{\omega} \varsigma_{max}\left(G(j\omega) - G_r^D(j\omega, D_r)\right) \\ &s.t. \quad G_r^D(s, D_r) \text{ is stable} \end{aligned} \tag{22.23}$$

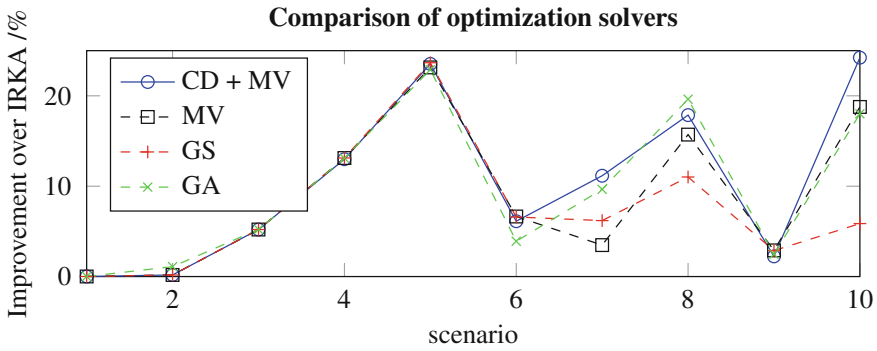


Fig. 22.3 Comparison of different solvers shows the effectiveness of coordinate descent followed by multivariate optimization

which represents a non-smooth, non-convex multivariate optimization problem in a $p \times m$-dimensional search space. In our experience, the best strategy considering both optimization time and optimal solution is given by a combination of *coordinate descent* (CD) [38] and subsequent *multivariate optimization* (MV). We refer to this combined strategy as CD+MV. The coordinate descent strategy is used in this setting somewhat like an initialization procedure to find a better starting point than $D_r^0 = 0$. This initialization is based on reducing the search space from $p \cdot m$ dimensions to just one, hence performing a much simpler univariate optimization in each step. Once one cycle has been conducted for all elements in the feed-through matrix, the resulting feed-through is used to initialize a nonlinear constrained optimization solver that minimizes the error with respect to the whole D_r matrix. We have used a sequential quadratic programming (SQP) method as implemented in MATLAB's `fmincon`, although acceptable options for this final step abound. Further information about optimization strategies can be found in [39].

The suitability of CD+MV is motivated by extensive simulations conducted comparing different strategies, such as direct multivariate optimization, *global search* (GS) [40], and *genetic algorithms* (GA) (cp. Fig. 22.3). Ultimately, we rely on the results of Sect. 22.3 to show that this procedure is effective.

22.3 Numerical Results

In the following we demonstrate the effectiveness of the proposed procedure by showing reduction results with different MIMO models. The reduction code is based on the sssMOR toolbox[1] [41]. For generation of vector fitting surrogates, we use the `vectfit3` function[2] [22, 32, 33]. Note that more recent implementation of MIMO

[1] Available at www.rt.mw.tum.de/?sssMOR.

[2] Available at www.sintef.no/projectweb/vectfit/downloads/vfut3/.

vector fitting introduced in [23] could be used instead, especially for improved robustness.

22.3.1 Heat Model

Our proposed procedure is demonstrated through numerical examples conducted on a MIMO benchmark model representing a discretized heat equation of order $N = 197$ with $p = 2$ outputs and $m = 2$ inputs [42].

Model reduction for this model was conducted for a range of reduced orders; the results are summarized in Table 22.1. The table shows the reduced order n, the order n_m of the error surrogate G_e^0, and the relative $\mathscr{H}_\infty$ error of the proposed ROM G_r^D, as well as the percentage improvement over the initial IRKA model. Our proposed method improves significantly on the $\mathscr{H}_\infty$ performance of IRKA, in some cases by more than 50%.

Figure 22.4 gives a graphical representation of the reduction results. The plots compare the approximation error achieved after applying MIHA, with a vector fitting surrogate as described in Sect. 22.2.3.1, to other reduction strategies. These include the direct reduction with IRKA, balanced truncation (BT), Optimal Hankel Norm Approximation (OHNA) as well as the optimization with respect to the actual error G_e^0 (MIHA without surrogate). For a better graphical comparison throughout the reduced orders studied, the errors are related to the theoretical lower bound given by

$$\underline{e}_{\mathscr{H}_\infty} := \varsigma_{n+1}^H, \tag{22.24}$$

Table 22.1 Results for the heat model problem

n	1	2	3	4	5	6	7	8	9	10
n_m	14	24	20	22	24	30	32	36	36	36
$\frac{\|G-G_r^D\|}{\|G\|}$	8.7e-2	7.6e-3	1.2e-2	1.2e-3	6.5e-4	5.7e-4	4.1e-4	1.6e-4	4.4e-5	8.6e-6
$1-\frac{\|G-G_r^D\|}{\|G-G_r^0\|}$	50.8%	39.0%	27.0%	36.7%	36.0%	44.8%	52.0%	44.6%	49.5%	42.6%

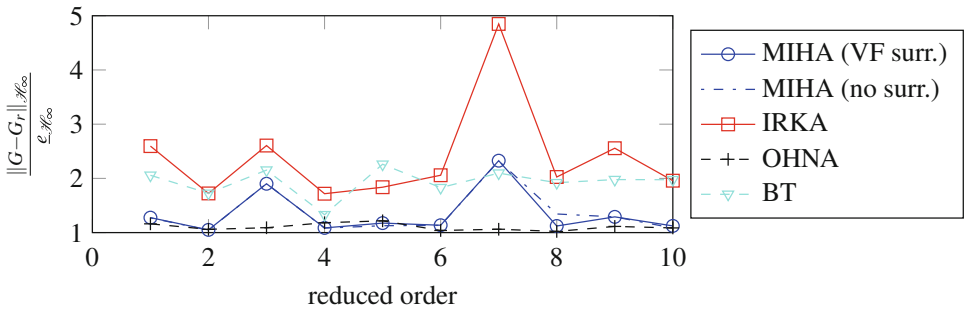


Fig. 22.4 Plot of the approximation error relative to the theoretical error bound

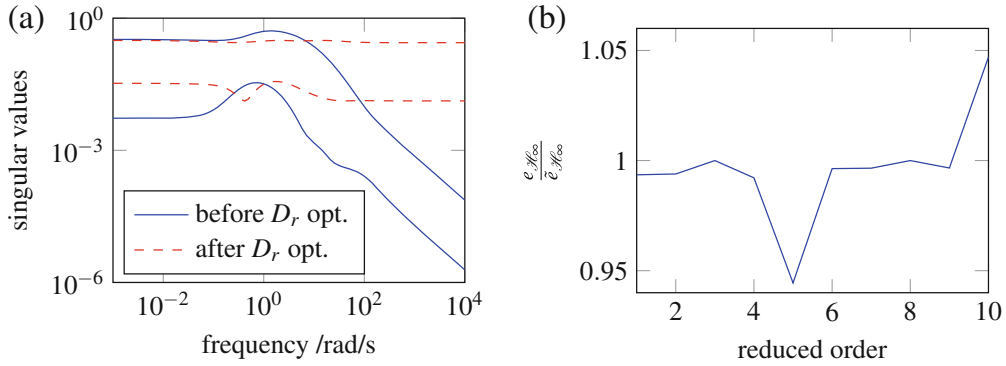


Fig. 22.5 Optimization with the surrogate effectively reduces and provides an accurate estimate of the true error. (**a**) Singular value plot of the error before and after optimization. (**b**) Comparison of error estimate $\tilde{e}_{\mathscr{H}_\infty}$ versus true error $e_{\mathscr{H}_\infty}$

with which we denote the Hankel singular value of order $n + 1$.

Notice how effectively the ROMs resulting from the D_r-optimization reduce the $\mathscr{H}_\infty$-error beyond what is produced by the IRKA ROMs and that they often, (here, in 9 out of 10 cases) yield better results than BT and sometimes (here, in 3 out of 10 cases) yield better results even than OHNA. Note also that the optimization with respect to the vector-fitting surrogate produces as good a result as optimization with respect to the true error. For reduced order $n = 8$, optimization with respect to the surrogate yields even a better result. This is not expected and may be due to the different cost functions involved, causing optimization of the true error to converge to a worse solution.

The plot also confirms our initial motivation in using IRKA models as starting points, since their approximation in terms of the $\mathscr{H}_\infty$ norm is often not far from BT. Finally, note how in several cases the resulting ROM is very close to the theoretical lower bound, which implies that the respective ROMs are not far from being the *global* optimum.

Figure 22.5a shows the approximation error before and after the feed-through optimization for a selected reduced order of 2. The largest singular value is drastically reduced (ca. 40%) by lifting up the value at high frequencies. This confirms our intuition that the $\mathscr{H}_\infty$-optimal reduced order model should have a nearly constant error modulus over all frequencies. Finally, Fig. 22.5b demonstrates the validity of the error estimate $\tilde{e}_{\mathscr{H}_\infty}$ obtained using the surrogate model.

22.3.2 ISS Model

Similar simulations were conducted on a MIMO model with $m = 3$ inputs and $p = 3$ outputs of order $N = 270$, representing the 1r component of the International Space Station (ISS) [24]. The results are summarized in Table 22.2 and Fig. 22.6. Note that the $\mathscr{H}_\infty$-error after IRKA is comparable to that of BT and the proposed

Table 22.2 Results for the ISS problem

n	2	4	6	8	10	12	14	16	18	20
n_m	12	18	12	18	18	15	42	48	30	30
$\frac{\|G-G_r^D\|}{\|G\|}$	2.7e-1	9.4e-2	8.4e-2	7.9e-2	3.6e-2	3.4e-2	2.2e-2	2.2e-2	1.0e-2	7.7e-3
$1-\frac{\|G-G_r^D\|}{\|G-G_r^0\|}$	7.5%	9.9%	8.8%	4.9%	9.5%	13.8%	23.3%	15.7%	3.5%	25.8%

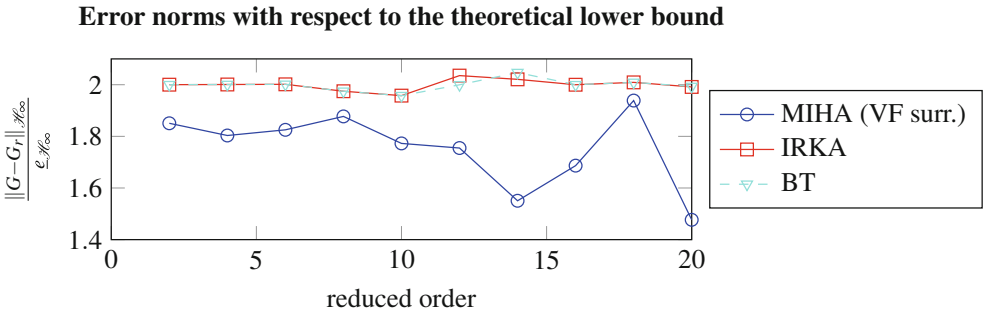


Fig. 22.6 Plot of the approximation error relative to the theoretical error bound (ISS)

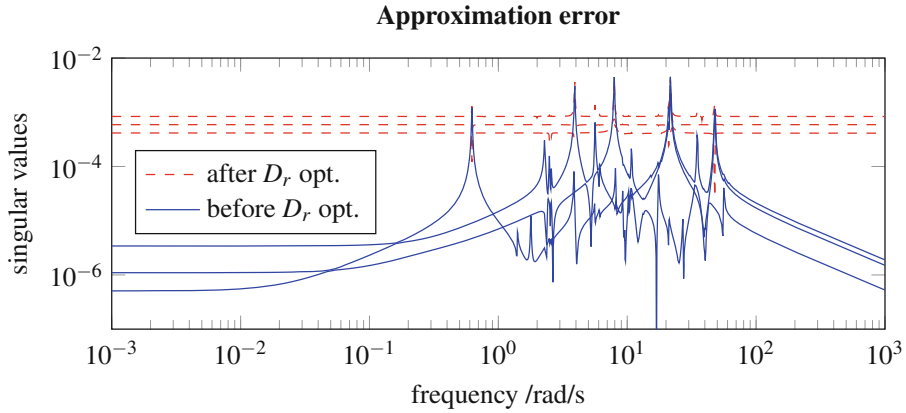


Fig. 22.7 Singular value plot of the error before and after optimization (ISS)

procedure is effective in further reducing the error, outperforming BT in all cases investigated.

Finally, note also in this case that the modulus of the error due to this $\mathscr{H}_\infty$-approximation procedure is nearly constant, as anticipated. This is demonstrated in Fig. 22.7, where the error plots for the reduction order $n = 10$ are compared.

Acknowledgements The work of the first author was supported by the German Research Foundation (DFG), Grant LO408/19-1, as well as the TUM Fakultäts-Graduiertenzentrum Maschinenwesen; the work of the second author was supported by the Einstein Foundation, Berlin; the work of the third author was supported by the Alexander von Humboldt Foundation.

References

1. Antoulas, A.C.: Approximation of Large-Scale Dynamical Systems. Society for Industrial and Applied Mathematics, Philadelphia, PA (2005)
2. Gallivan, K.A., Vandendorpe, A., Van Dooren, P.: On the generality of multipoint padé approximations. In: 15th IFAC World Congress on Automatic Control (2002)
3. Beattie, C.A., Gugercin, S.: Model reduction by rational interpolation. In: Benner, P., Cohen, A., Ohlberger, M., Willcox, K. (eds.) Model Reduction and Approximation: Theory and Algorithms. SIAM, Philadelphia (2017)
4. Grimme, E.J.: Krylov Projection Methods for Model Reduction. Ph.D. thesis, Department of Electrical Engineering, University of Illinois at Urbana Champaign, 1997
5. Gallivan, K.A., Vandendorpe, A., Van Dooren, P.: Model reduction of MIMO systems via tangential interpolation. SIAM J. Matrix Anal. Appl. **26**(2), 328–349 (2004)
6. Gallivan, K.A., Vandendorpe, A., Van Dooren, P.: Sylvester equations and projection-based model reduction. J. Comput. Appl. Math. **162**(1), 213–229 (2004)
7. Gugercin, S., Antoulas, A.C., Beattie, C.A.: $\mathscr{H}_2$ model reduction for large-scale linear dynamical systems. SIAM J. Matrix Anal. Appl. **30**(2), 609–638 (2008)
8. Antoulas, A.C., Astolfi, A.: $\mathscr{H}_\infty$-norm approximation. In: V. Blondel, A. Megretski (eds.) Unsolved Problems in Mathematical Systems and Control Theory, pp. 267–270. Princeton University Press, Princeton (2002)
9. Helmersson, A.: Model reduction using LMIs. In: Proceedings of 1994 33rd IEEE Conference on Decision and Control, vol. 4, pp. 3217–3222. Lake Buena Vista, IEEE (1994). doi:10.1109/CDC.1994.411635
10. Varga, A., Parrilo, P.: Fast algorithms for solving $\mathscr{H}_\infty$-norm minimization problems. In: Proceedings of the 40th IEEE Conference on Decision and Control, 2001, vol. 1, pp. 261–266. IEEE, Piscataway, NJ (2001)
11. Kavranoglu, D., Bettayeb, M.: Characterization of the solution to the optimal $\mathscr{H}_\infty$ model reduction problem. Syst. Control Lett. **20**(2), 99–107 (1993)
12. Glover, K.: All optimal hankel-norm approximations of linear multivariable systems and their linf-error bounds. Int. J. Control **39**(6), 1115–1193 (1984)
13. L.N. Trefethen, Rational Chebyshev approximation on the unit disk. Numer. Math. **37**(2), 297–320 (1981)
14. Gugercin, S., Antoulas, A.C.: A survey of model reduction by balanced truncation and some new results. Int. J. Control **77**(8), 748–766 (2004)
15. Benner, P., Quintana-Ortí, E.S., Quintana-Ortí, G.: Computing optimal Hankel norm approximations of large-scale systems. In: 43rd IEEE Conference on Decision and Control, 2004. CDC, vol. 3, pp. 3078–3083. IEEE, Piscataway, NJ (2004)
16. Li, J.-R.: Model reduction of large linear systems via low rank system gramians, Ph.D. thesis, Massachusetts Institute of Technology, 2000
17. Benner, P., Kürschner, P., Saak, J.: Self-generating and efficient shift parameters in ADI methods for large Lyapunov and Sylvester equations. Electron. Trans. Numer. Anal. **43**, 142–162 (2014)
18. Kürschner, P.: Efficient low-rank solution of large-scale matrix equations, Ph.D. thesis, Otto-von-Guericke Universität Magdeburg (2016)
19. Sabino, J.: Solution of large-scale Lyapunov equations via the block modified Smith method. Ph.D. thesis, Citeseer (2006)
20. Gugercin, S., Sorensen, D.C., Antoulas, A.C.: A modified low-rank smith method for large-scale Lyapunov equations. Numerical Algoritm. **32**(1), 27–55 (2003)
21. Flagg, G.M., Beattie, C.A., Gugercin, S.: Interpolatory $\mathscr{H}_\infty$ model reduction. Syst. Control Lett. **62**(7), 567–574 (2013)
22. Gustavsen, B., Semlyen, A.: Rational approximation of frequency domain responses by vector fitting. IEEE Trans. Power Delivery **14**(3), 1052–1061 (1999)

23. Drmač, Z., Gugercin, S., Beattie, C.: Vector fitting for matrix-valued rational approximation. SIAM J. Sci. Comput. **37**(5), A2346–A2379 (2015)
24. Chahlaoui, Y., Van Dooren, P.: A collection of benchmark examples for model reduction of linear time invariant dynamical systems. Working Note 2002-2, 2002
25. Mayo, A.J., Antoulas, A.C.: A framework for the solution of the generalized realization problem. Linear Algebra Appl. **425**(2–3), 634–662 (2007)
26. Beattie, C.A., Gugercin, S.: Interpolatory projection methods for structure-preserving model reduction. Syst. Control Lett. **58**(3), 225–232 (2009)
27. Golub, G.H., Van Loan, C.F.: Matrix Computations, vol. 3. JHU Press, Baltimore (2012)
28. Antoulas, A.C., Anderson, B.D.Q.: On the scalar rational interpolation problem. IMA J. Math. Control Inf. **3**(2-3), 61–88 (1986)
29. Anderson, B.D.O., Antoulas, A.C.: Rational interpolation and state-variable realizations. Linear Algebra Appl. **137**, 479–509 (1990)
30. Lefteriu, S., Antoulas, A.C.: A new approach to modeling multiport systems from frequency-domain data. IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **29**(1), 14–27 (2010)
31. Beattie, C.A., Gugercin, S.: Realization-independent $\mathscr{H}_2$-approximation. In: 51st IEEE Conference on Decision and Control, pp. 4953–4958. IEEE, Piscataway, NJ (2012)
32. Gustavsen, B.: Improving the pole relocating properties of vector fitting. IEEE Trans. Power Delivery **21**(3), 1587–1592 (2006)
33. Deschrijver, D., Mrozowski, M., Dhaene, T., De Zutter, D.: Macromodeling of multiport systems using a fast implementation of the vector fitting method. IEEE Microwave Wireless Compon. Lett. **18**(6), 383–385 (2008)
34. Drmac, Z., Gugercin, S., Beattie, C.: Quadrature-based vector fitting for discretized h_2 approximation. SIAM J. Sci. Comput. **37**(2), A625–A652 (2015)
35. Mitchell, T., Overton, M.L.: Fixed low-order controller design and $\mathscr{H}_\infty$ optimization for large-scale dynamical systems. IFAC-PapersOnLine **48**(14), 25–30 (2015)
36. Mitchell, T., Overton, M.L.: Hybrid expansion–contraction: a robust scaleable method for approximating the $\mathscr{H}_\infty$ norm. IMA J. Numer. Anal. **36**(3), 985–1014 (2016)
37. Aliyev, N., Benner, P., Mengi, E., Voigt, M.: Large-scale computation of $\mathscr{H}_\infty$ norms by a greedy subspace method. SIAM J. Matrix Anal. Appl. (2016, submitted to). http://portal.ku.edu.tr/~emengi/papers/large_hinfinity.pdf
38. Wright, S.J.: Coordinate descent algorithms. Math. Program. **151**(1), 3–34 (2015)
39. Nocedal, J., Wright, S.: Numerical Optimization. Springer Science and Business Media, New York (2006)
40. Ugray, Z., Lasdon, L., Plummer, J., Glover, F., Kelly, J., Martí, R.: Scatter search and local NLP solvers: a multistart framework for global optimization. INFORMS J. Comput. **19**(3), 328–340 (2007)
41. Castagnotto, A., Varona, M.C., Jeschek, L., Lohmann, B.: sss and sssMOR: Analysis and reduction of large-scale dynamic systems with MATLAB. at-Automatisierungstechnik **65**(2), 134–150(2017). doi:10.1515/auto-2016-0137
42. Antoulas, A.C., Sorensen, D.C., Gugercin, S.: A survey of model reduction methods for large-scale systems. In: Structured Matrices in Opera Structured, Numerical Analysis, Control, Signal and Image Processing. Contemporary Mathematics, vol. 280, pp. 193–219. AMS Publications, Providence, RI (2001)

A.5 sss & sssMOR: Analysis and reduction of large-scale dynamic systems in MATLAB

Summary: This contribution gives an introduction and overview to the MATLAB toolboxes developed and released during this thesis, namely `sss` and `sssMOR`. At the beginning of the paper, a broad introduction to the *curse of dimensionality* and the need for sparse matrix representations is given, which explains the limitations of MATLAB's built-in Control System Toolbox. Subsequently, an introduction to state-space models and projective model reduction is given. The second part is dedicated to introducing the `sss` toolbox, i.e. the class of `sss` objects and related analysis and manipulation functions. This section ends with a description of the tailoring concept, using `Opts` structure objects to define optional execution parameters, that would be otherwise chosen automatically from the functions. In addition, the main computational functions `solveLse` and `lyapchol` are introduced, These functions are the core of `sss` in that they define the algorithms to solve linear systems of equations and matrix equations, the two key computations in analysis and reduction. The definition of these functions allows easy extension to other algorithms that might be preferred by the users, as they only need to be added in these two functions. The third section is dedicated to `sssMOR`, introducing `ssRed` objects, i.e. reduced-order dynamical model objects containing information about the full-order model and the reduction. Further, a selection of model reduction functions is presented, ranging from classic to more state-of-the-art methods. The end of this section is dedicated to the `sssMOR` app, i.e. a graphical user interface that provides the main functionality of `sss` and `sssMOR`. Finally, the fourth section presents some numerical examples and comparisons to the built-in functions.
This contribution received the 2018 "Best Paper Award" in the section "Tools" of the journal "at-Automatisierungstechnik".

Contribution(s): Writing and numerical examples have been conducted predominantly by the first author with contributions of the second and third author. Software development has been initiated, coordinated and conducted predominantly by the first and second author with support of the third author and contributions from Jorge Luiz Moreira Silva, Rodrigo Mancilla, Siyang Hu, Michael Ott, Max Gille, Jonathan Seiti Miura, Jonas Ferber, Maximilian Loderer, Niklas Kochdumper. Stefan Jaensch, Thomas Emmert, Philip Holzwarth and Nico-Philipp Walz.

Tools

Alessandro Castagnotto*, Maria Cruz Varona, Lisa Jeschek, and Boris Lohmann

sss & sssMOR: Analysis and reduction of large-scale dynamic systems in MATLAB

sss & sssMOR: Analyse und Reduktion dynamischer Systeme hoher Ordnung in MATLAB

DOI 10.1515/auto-2016-0137
Received November 30, 2016; accepted December 22, 2016

Abstract: We present two MATLAB toolboxes, provided as open-source code, that expand the capabilities of the Control System Toolbox to large-scale models. **sss** allows the definition and analysis of *sparse state-space* (`sss`) objects with functions (such as `bode`, `step`, `norm`,...) revisited to exploit the *sparsity* of the system matrices. **sssMOR** entails model reduction algorithms that capture the relevant dynamics of high order systems in models of significantly lower dimensions. The `sssMOR_App` provides a graphical user interface for easy interaction with the tools. With **sss** and **sssMOR** it is possible to analyze dynamical systems with state-space dimensions higher than $\mathcal{O}(10^4)$, which is typically the limit for built-in `ss` objects. In this contribution, we give a first introduction to the toolboxes and the main functionality. Numerical examples show the advantages of using the tools.

Keywords: Large-scale systems, control systems, model reduction, MATLAB, control system toolbox.

Zusammenfassung: Wir stellen zwei MATLAB Tooboxen vor, welche wir open-source zur Verfügung stellen und die Funktionalität der Control System Toolbox auf hochdimensionale Modelle erweitern. **sss** ermöglicht die Definition und Analyse dünnbesetzter (Engl.: sparse) Zustandsraummodelle mit Funktionen (z.B. `bode`, `step`, `norm`,...), welche zur Ausnutzung der Dünnbesetzheit angepasst wurden. **sssMOR** enthält Modellreduktionsalgorithmen, womit die relevante Dynamik in Modellen deutlich niedriger Ordnung beschrieben werden kann. Die `sssMOR_App` bietet eine graphische Benutzeroberfläche für eine einfachere Interaktion mit den Tools an. Mit **sss** und **sssMOR** ist es möglich, Zustandraummodelle der Ordnung höher als $\mathcal{O}(10^4)$, die typische Grenze bei built-in Funktionen, zu analysieren. In diesem Betrag wird eine erste Einführung in die Toolboxen gegeben. Numerische Beispiele motivieren die Vorteile deren Nutzung.

Schlüsselwörter: Hochdimensionale Modelle, dynamische Syteme, Modellordnungsreduktion, MATLAB, Control System Toolbox.

1 Introduction

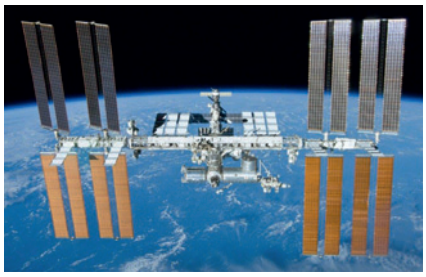
1.1 The curse of dimensionality

The accurate modeling of dynamical systems often results in a large number of differential equations and state variables describing the system behavior in time. This is the case, for instance, when discretizing partial differential equations over a fine spatial grid or modeling systems with a large number of components like integrated circuits. Applications in which large-scale models arise are numerous and cover different domains. Figure 1 shows a few examples[1].

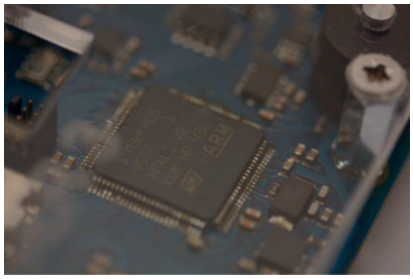
Especially in a control theoretic setting, such models are often formulated in a state-space representation, where the state vector $x(t) \in \mathbb{R}^n$ describes the state of the system at each point in time. With ever rising demands on modeling fidelity, the model *order n* can become prohibitively large, easily exceeding hundreds of thousands. This poses a great challenge on the numerical treatment

***Corresponding author: Alessandro Castagnotto,** Technical University of Munich, Department of Mechanical Engineering, Chair of Automatic Control, Boltzmannstr. 15, 85748 Garching, Germany, e-mail: a.castagnotto@tum.de
Maria Cruz Varona, Lisa Jeschek, Boris Lohmann: Technical University of Munich, Department of Mechanical Engineering, Chair of Automatic Control, Boltzmannstr. 15, 85748 Garching, Germany

1 The gyroscope photograph "Badger 3" by Adam Greig is licensed under CC BY-SA 2.0.



(a) International Space Station (component 1R)



(b) Gyroscope

(c) Cooling of a steel profile

Figure 1: Examples of large-scale dynamical systems (described in Section 3).

of such models, first and foremost due to storage limitations. In fact, even in the simpler yet common case of linear systems, storing a matrix of such dimensions can become a hard task, as it is demonstrated by the following MATLAB[2] example that defines an identity matrix of size $n = 10^5$:

```
» A = eye(1e5);
Error using eye
Requested 100000x100000 (74.5GB) array exceeds
maximum array size preference.
Creation of arrays greater than this limit may
take a long time and cause MATLAB to become
unresponsive.
```

The problem is that storing all $n^2 = 10^{10}$ entries of the matrix requires roughly $8 \cdot 10^{10}$ Bytes = 80 GB of memory, which is not feasible on a standard computer. The question at this point is natural: *is there any way of storing large matrices in MATLAB at all?*

Luckily, the answer lies in the structure of the matrix `A=eye(n)` we wish to store. In this case, the number of nonzero entries, denoted by `nnz(A)`, is small compared to a full matrix, since they only lie on the diagonal. Such a matrix is called *sparse* and can be defined in MATLAB using `A = speye(n)`,

```
» A = speye(1e5); whos A
Name      Size          Bytes  Class    Attributes

A     100000x100000    2400008  double   sparse
```

which requires only 2.4 MB of storage! By exploiting sparsity, it is possible to store diagonal matrices up until a size of $\mathcal{O}(10^8)$. *Does this mean that we can store also large-scale dynamical systems in MATLAB?*

Unfortunately, this is not the case, at least not with the built-in functionality of MATLAB's Control System Toolbox[2], as demonstrated by the following example:

```
» A = speye(1e5); b = rand(1e5,1); c = b';
» sys = ss(A,b,c,[]);
Error using ss (line 259)
Requested 10000000000x1 (9.3GB) array exceeds
maximum array size.
```

In fact, the dynamic system objects for state-space models such as `ss` or `dss` do not support sparse matrices and convert all system matrices to *full* arrays. This restricts the maximum model order that can be stored and analyzed in MATLAB drastically.

The **sss** toolbox presented in this contribution overcomes this drawback by defining *sparse state-space* objects, i.e. dynamic systems defined by sparse matrices (`sss`). This allows both to store large-scale models with *hundreds of millions of state variables*, as well as to exploit the sparsity of the system matrices to reduce the computational burden when manipulating and analyzing them.

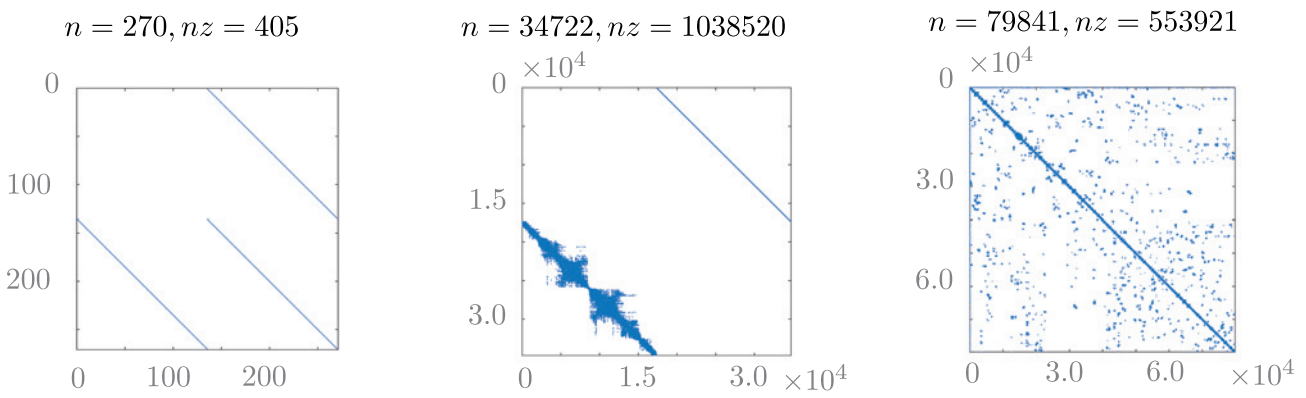
```
» A = speye(1e7); b = rand(1e7,1); c = b';
» sys = sss(A,b,c)

sys =

(SSS)(SISO)
10000000 states, 1 inputs, 1 outputs
Continuous-time state-space model.
```

That said, even when using the **sss** toolbox, computations such as simulations, optimization and control design algorithms based on large-scale models will require a substantial amount of computational resources, provided they can be carried through. For this reason, in the

2 MATLAB and Control System Toolbox (Release 2015b) are registered trademarks of The MathWorks, Inc., Natick, Massachussets, United States.



(a) International Space Station (component 1R)

(b) Gyroscope

(c) Cooling of a steel profile

Figure 2: Sparsity patterns of the systems shown in Figure 1.

large-scale setting we often seek reduced order models (ROM) of much smaller order $q \ll n$ that capture the relevant dynamics and possibly preserve fundamental properties. The **sssMOR** toolbox provides a set of model order reduction (MOR) routines that range from classic methods like *modal truncation*, *balanced truncation* and *rational Krylov* methods, as well as state-of-the-art algorithms that allow for an adaptive choice of reduced order and optimal reduction parameters.

In this contribution, we give a brief overview of the structure, functionality and advantages of the **sss** and **sssMOR** toolboxes, after revising a few basic concepts on large-scale models and model reduction.

1.2 Preliminaries

1.2.1 Linear time-invariant state-space models

In this contribution, we consider linear time-invariant systems that can be represented in state-space by models of the form

$$\begin{aligned} E\,\dot{x} &= A\,x + B\,u \\ y &= C\,x + D\,u \end{aligned} \tag{1}$$

where $E \in \mathbb{R}^{n\times n}$ is the *descriptor matrix*, $A \in \mathbb{R}^{n\times n}$ is the system matrix and $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $y \in \mathbb{R}^p$ $(p, m \ll n)$ represent the state, input and output of the system respectively. These types of models are often used to describe the dynamic behavior of the system under consideration, at least locally around some operating point. Due to the amount of system theoretic results and control design schemes available, they are widely used both in literature and in real-life applications.

By taking the Laplace transforms $\mathcal{L}\{u(t)\} = U(s)$, $\mathcal{L}\{x(t)\} = X(s)$ and $\mathcal{L}\{y(t)\} = Y(s)$ and assuming zero initial conditions, we obtain a representation of (1) in the frequency domain known as the transfer function matrix $G(s)$ satisfying

$$\begin{aligned} G(s) &:= C\,(sE - A)^{-1}\,B + D, \\ Y(s) &= G(s) \cdot U(s) \end{aligned} \tag{2}$$

1.2.2 Sparse matrices

The system matrices E and A in (1) are generally *sparse*, meaning that the number of nonzero elements `nnz(E)`, `nnz(A)` is significantly less than the number of entries of the full matrix. On the other hand, a matrix that has only a few—if any—zeros is called *dense*. Broadly speaking, the sparse structure of the system matrices results from the modeling itself and is due to the fact that each element in the model has a direct influence only to its neighboring elements. Note also at this point that—provided E is regular—the *implicit*[3] state space representation in (1) is equivalent to a respective *explicit* representation with $E = I$. However, the representation in implicit form is crucial to preserve the sparsity of the system matrices.

Figure 2 shows, as an example, the sparsity patterns of the different benchmark models corresponding to

3 State space representations as in (1) are sometimes referred to as *descriptor systems*. However, note that this term is used inconsistently in the literature both for systems with regular *and* singular E. The latter case, better known as systems of *Differential Algebraic Equations* is however much more involved.

the systems in Figure 1, where each blue dot represents a nonzero entry.

Sparse matrices bear a great potential especially in terms of the *storage requirements*. In fact, while storing a dense matrix of size N in IEEE double-precision floating-point format takes $N^2 \cdot 8$ Bytes, a sparse matrix is equivalently represented by its nonzero entries. For this reason, data structures have been developed to efficiently store sparse matrices (e.g. *compressed-column* matrix format used in MATLAB [1]).

Moreover, the sparsity of the matrices can be exploited also to reduce the *computational effort*, since operations on zero elements can be explicitly avoided. Matrix-vector multiplications are a basic example of operations that can be significantly speed-up and, in addition, a large number of algorithms have been adapted. For example, the LU-decomposition of a sparse matrix generally runs much faster than the LU-decomposition conducted on the full matrix. Furthermore, by *reordering* the entries of a matrix and using graph theoretical considerations, it is possible to minimize the *fill-in*, i.e. the number of new nonzero entries in the LU factors generated during the decomposition. A more detailed discussion on *direct* and *iterative* sparse algorithms lies outside the scope of this treatise and can be found e.g. in [1–4]. For the purposes of this contribution, it is however important to understand at this point that the sparsity of the matrices describing dynamical systems can and *should* be exploited by the numerical algorithms employed.

1.2.3 Model order reduction by projection

With only a few exceptions (cp. e.g. [5, 6]), model reduction for dynamical systems in the linear and nonlinear case is generally performed by means of *projections* with appropriately chosen subspaces. For this, it is assumed that the trajectory of a given LTI model (1) dwells predominantly in a q-dimensional subspace $\mathrm{Ran}(V)$. Following this assumption, the full state $x \in \mathbb{R}^n$ can be approximated well in reduced coordinates $x_r \in \mathbb{R}^q$ through following relationship

$$x = V x_r + e, \tag{3}$$

where e denotes the approximation error. Using this ansatz in the original dynamics (1) results in an overdetermined system of differential equations

$$E V \dot{x}_r = A V x_r + B u + \underbrace{A e - E \dot{e}}_{\varepsilon}, \tag{4}$$

which generally cannot be solved uniquely for x_r. To overcome this burden, the dynamics (4) are projected onto a low dimensional subspace by means of the projector $\Pi = EV(W^\top EV)^{-1}W^\top$. Enforcing the Petrov–Galerkin condition, the projected differential equation in reduced coordinates (4) is solved for the residual ε that vanishes after projection, hence satisfying $W^\top \varepsilon = 0$. The reduced order dynamics are finally obtained by considering the low-dimensional set of differential equations

$$\begin{aligned} \overbrace{W^\top E V}^{E_r} \dot{x}_r &= \overbrace{W^\top A V}^{A_r} x_r + \overbrace{W^\top B}^{B_r} u \\ y_r &= \underbrace{C V}_{C_r} x_r + \underbrace{D}_{D_r} u \end{aligned} \tag{5}$$

Following this setting, the main task of any model reduction technique can be boiled down to appropriately designing the projection matrices V, W.

1.3 Overview of the toolbox

The functions presented in this contribution are distributed within two toolboxes: **sss** and **sssMOR**.

sss expands the capabilities of the Control System Toolbox by allowing the definition of sparse state space (*sss*) objects as well as analysis functions (such as `bode`, `step`, `isstable` etc.) tailored to exploit the sparsity of the system matrices whenever possible. As such, its usage is not limited to the field of model order reduction and may be introduced in several applications where large-scale dynamical systems are involved. This is the case e.g. in the taX[4] toolbox for modeling and analysis of thermoacoustic networks [7, 8].

sssMOR entails both classic and state-of-the-art model reduction algorithms that capture the relevant dynamics of the high order system in models of significantly lower dimension. The reduction functions take `sss` full order models and return reduced `ssRed` models with the desired characteristics, depending on the chosen method. Figure 3 gives a schematic overview of the toolboxes and their dependencies. As indicated by the graphic, **sssMOR** comes with the graphical user interface `sssMOR_App`, which will be briefly introduced in Section 3.4.

The article is structured as follows: In Section 2 we give an introduction to the main functionality of the **sss** toolbox. Analogously Section 3 gives an overview of the functionality of the **sssMOR** toolbox. In Section 4 we demonstrate the functionality by means of simple numerical examples demonstrating the validity of using **sss** and

4 https://tax.wiki.tum.de/

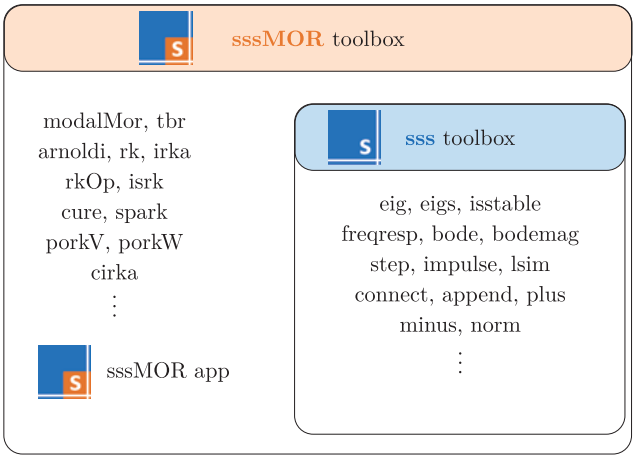


Figure 3: Overview of the toolboxes presented in this contribution.

sssMOR when dealing with mid- to large-scale models. Finally in Section 5 we summarize the main strengths of the toolboxes, acknowledge third party and older contributors and give an outlook on new functionalities being developed.

2 sss – Sparse State-Space Toolbox

In this section, we introduce the **sss** toolbox and its main functionality. Due to space limitations we can only give a brief introduction. For more details on the functionality, algorithms and optional parameters refer to the detailed documentation of the toolbox directly in MATLAB using the `help` and `doc` commands.

2.1 sss – sparse state-space objects

To overcome the storage limitations of `ss` objects and exploit sparsity, the `sss` class, i.e. the *sparse state-space* class, was created. This class allows the definition of `sss` objects, i.e. state-space objects defined by sparse matrices.

2.1.1 Definition of sss-objects

`sss` models of dynamical systems can be defined based on given system matrices by the call `sys=sss(A,B,C,...)` both for continuous time and discrete-time models.

`sss`	definition of sparse state-space objects
In	system matrices A,B,C,D,E; sampling time
Out	sss-object
Opt(s)	-

The `sss` class not only stores the sparse system, but also provides useful information about the model:

```
» load('iss.mat'); sys = sss(A,B,C)
sys =

(SSS)(MIMO)
270 states, 3 inputs, 3 outputs
Continuous-time state-space model.

» sys.n
ans =

270

» sys.isDescriptor
ans =

0

» sys.isMimo
ans =

1
```

Other related sss functions
`c2d`, `disp`, `loadSss`, `second2first`

2.1.2 Manipulation of `sss`-objects

Often times a given model is only a subsystem, or the desired model results as the sum or difference of individual models. Also, if the given model is MIMO, we may want to select a subsystem corresponding to certain input/output channels. Following functions allow these operations with `sss`-objects:

Other related sss functions
`plus`, `minus`, `truncate`

`connect`	block diagram interconnection of individual `sss`-models
In	`sss`-objects, input and output names
Out	`sss`-object of the interconnection
Opt(s)	-

`append`	group `sss`-models by appending their inputs and outputs
In	`sss`-objects
Out	`sss`-object with all input/outputs of the individual models
Opt(s)	-

2.2 System analysis

The definition of a dynamic system model (`sss`, `ss`) is only the first step in analyzing the characteristics of the underlying system. Typically, one may want to compute the eigenvalues of a model, check for stability, analyze the behavior in frequency or time domain and conduct simulations for certain input signals.

Many of the system analysis functions known from the Control System Toolbox can be found in **sss**, allowing similar analysis with large-scale models. Following this idea, **sss** has been developed keeping the compatibility with existing MATLAB scripts and its familiar syntax in mind. As such, most functions are named exactly like to the corresponding MATLAB built-in counterpart and can be called by simply passing a sparse `sss`-object instead of a dense `ss`-object.

2.2.1 Eigenvalues and stability

Several characteristic properties of linear dynamical systems are tied to their eigenvalues. For this reason, it is often desirable to get an overview of the spectrum of a model. To compute all eigenvalues of a given `sss`-object, one may use `eig(sys)`:

`eig`	compute all eigenvalues of an `sss`-model
In	`sss`-object
Out	eigenvalues, eigenvectors
Opt(s)	-

Note however that this computation requires dense operations (Schur decompositions) and might be very demanding in terms of computational time and storage [9]. On the other hand, the computation of a few eigenvalues can be conducted efficiently for large-scale models and can be obtained by calling `eigs(sys)`:

`eigs`	compute some eigenvalues of an `sss`-model
In	`sss`-object
Out	eigenvalues, eigenvectors
Opt(s)	which eigenvalues to compute (`'lm'`, `'sm'`, ...)

Both functions are the basis for determining the stability of a given model. The function `isstable(sys)` first attempts at computing the eigenvalue with largest real part using `eigs` and falls back to `eig` otherwise.

`isstable`	determine stability of an `sss`-model
In	`sss`-object
Out	boolean value, spectral abscissa
Opt(s)	-

Other related sss functions

`pzmap`, `zeros`, `poles`, `zpk`

2.2.2 Frequency response and Bode plot

To analyze a given model in the frequency domain, we often need to evaluate its transfer function matrix $G(s)$ (2) at several frequencies in the complex plain, typically on the imaginary axis. This can be done by the function `freqresp(sys)`.

`freqresp`	frequency response of an `sss`-model
In	`sss`-object(s), frequencies/frequency range
Out	values of transfer function matrix, frequencies
Opt(s)	return an `frd`-object

To get an overview over the dynamic behavior for several frequencies, the amplitude and phase of $G(s)$ are typically plotted over the imaginary axis in a *Bode plot*.

`bode`	generate amplitude and phase plot of `sss`-model
In	`sss`-object(s), frequencies/frequency range
Out	bode plot; magnitude, phase and frequencies
Opt(s)	line styles

Other related sss functions

`bodeplot`, `bodemag`, `sigma`

2.2.3 System norms

A fundamental characterization of dynamical systems is given by system norms defined in Lebesgue $\mathcal{L}$ and Hardy $\mathcal{H}$ spaces. This is the case, for instance, when measuring the distance between two models. Just like in the built-in case, the `norm` function computes the $\mathcal{H}_2$ and $\mathcal{L}_\infty$ norms of models passed as arguments. Unlike the built-in case, the function uses sparse computations and can therefore deal with large-scale models. The *Lyapunov* equations required for the $\mathcal{H}_2$-norm computations are solved using sparse iterative solvers and the $\mathcal{L}_\infty$ norm is estimated by evaluating the frequency response and its derivatives at different frequencies, which requires only the solution of large sparse linear systems of equations (LSE). More details on the implementation will be given in Section 2.3.2.

`norm`	compute system norm of an `sss`-model
In	`sss`-object
Out	system norm; peak gain frequency
Opt(s)	2, inf

2.2.4 Time domain analysis – response to impulse, step and arbitrary inputs

Sometimes the behavior of a dynamical system is best interpreted and analyzed in the time domain, for example by analyzing the response from an initial state $x(t = 0)$ under the influence of a driving signal $u(t)$. Typical inputs used in a system theoretic setting are *impulse* $u(t) = \delta(t)$ and *step* signals $u(t) = \int_{-\infty}^{t} \delta(\tau)d\tau$.

`impulse`	generate impulse response plot of an `sss`-model
In	`sss`-object(s), time vector/final time
Out	impulse plot; impulse response and time vectors
Opt(s)	line styles

`step`	generate step response plot of an `sss`-model
In	`sss`-object(s), time vector/final time
Out	step plot; step response and time vectors
Opt(s)	line styles

To analyze the response to arbitrary input signals $u(t)$ there are several functions implemented, e.g. `lsim`:

`lsim`	simulate the response of an `sss`-model to an arbitrary input
In	`sss`-object(s), input sequence, input sampling time
Out	response plot; response, state and time vectors
Opt(s)	time integration method

Other related sss functions

`simBackwardEuler`, `simDiscrete`, `simForwardEuler`, `simRK4`

2.3 General properties and advantages of `sss`

The following sections give a brief overview of some general aspects that characterize the functionality of **sss**, such as customizability, the handling of *core large-scale* computations and compatibility with built-in MATLAB functions.

2.3.1 `Opts` structures – from highly automatized to highly tailored

Most **sss** (and **sssMOR**) functions allow a variable `Opts` of type `structure` as an optional input. This variable is used to influence the function execution, e.g. by defining parameters or choosing amongst different methods implemented. The specification of these parameters is optional: all of them have default values, allowing inexpert users to get started quickly, without preventing expert users from making their own choices.

2.3.2 `solveLse` and `lyapchol` – the core of sss

The main goal of the **sss** toolbox is to preserve the sparsity of large-scale models and *exploit* it for computations, that would otherwise be expensive or even unfeasible. For this reason, many operations that are usually involved in the analysis of linear systems, such as *eigenvalue*, *singular value*, *Hessenberg* and *Schur* decompositions, have to be avoided at all times. In general, this requires the development of alternative methods, which sometimes can compute only approximations to the quantities of interest (compare e.g. `eigs` and `eig`).

Therefore, in general the largest cost allowed when dealing with large-scale models is given by the solution of *sparse linear systems of equations* (LSE). In **sss** this task is dedicated to the function `solveLse` which is called from all other functions whenever a linear system of the form $Ax = b$ needs to be solved. It is highly customizable to solve large sparse systems (either with sparse LU or iterative solvers) as well as small dense systems (full LU or Hessenberg reduction), depending on the problem at hand. Further, the function automatically recycles information from previous solutions by using `persistent` variables. In this case, the solution of different LSEs with common left-hand side can be computed efficiently.

Another computation frequently encountered in the analysis and reduction of LTI models is given by the solution of *generalized Lyapunov equations*, i.e. equations of the form

$$AXE^\top + EXA^\top + BB^\top = 0, \tag{6}$$

where the solution X is symmetric positive definite and can hence be represented in terms of its *Cholesky factor*

$$X = RR^\top. \tag{7}$$

For small dense problems, equations of type (6) are usually solved for R using direct methods, e.g. the algorithm by Hammarling [10]. For large-scale models, iterative methods are available to find low-rank approximations to $X \approx \tilde{R}\tilde{R}^\top$ with rectangular factors $\tilde{R} \in \mathbb{C}^{n\times q}$. Amongst all, the *Alternating Direction Implicit* (ADI) [11–15] algorithm and *Rational Krylov Subspace Methods* (RKSM) [16, 17] are best known. In **sss**, an own version of the function `lyapchol` is implemented and called whenever a Lyapunov equation of type (6) needs to be solved. Similarly to the case of `solveLse`, this function is customizable so that

both small dense problems (calling built-in `lyapchol`) and large sparse problems (calling the third-party toolbox M-M.E.S.S.[5] [18], which performs ADI) can be solved.

Therefore, all expensive sparse computations are performed from the functions `solveLse` and `lyapchol`. These functions are customizable in their behavior through respective `Opts` structures and their functionalities can be easily extended individually by the users to include their favorite solvers: just add the respective function call and `Opts` entry to either `solveLse` or `lyapchol`.

2.3.3 Compatibility with built-in functions

The goal of the **sss** toolbox is to extend the capability of the Control System Toolbox to `sss` models. As such, all functions that have an equivalent in built-in MATLAB have a compatible calling behavior. This means that the usage of input and outputs of the built-in functions is supported by the **sss** functions, sometimes even extended to improve usability and/or capability.

For example, the function `freqresp(sys,w)` computes the frequency response over the imaginary axis for real and imaginary frequency vectors `w`, but can also compute the frequency response over the whole complex plane if the frequencies in `w` have nonzero real and imaginary parts. In addition, by passing the option `Opts.frd = true`, the output of `freqresp` is an `frd` object, a MATLAB built-in object that can be used to plot the frequency response without repeated computations. As another example, the function `isstable` returns a boolean `true/false` if called with one output, just like in the built-in case. However, it can also return the *spectral abscissa* as a second output argument, i.e. the largest real part in the spectrum.

In developing the `sss` functions, one of our goals is to allow users to take existing scripts based on the Control System Toolbox, change the system definition from `ss` or `dss` to `sss` and run the same script using **sss**.

3 sssMOR – Model Order Reduction Toolbox

Even when using **sss** functions to exploit sparsity, computations such as simulations, optimization and control design algorithms based on the full order models (FOM) will require a substantial amount of time, provided they can be carried through. For this reason, in the large-scale setting we often seek reduced order models (ROM) of much smaller size that capture the relevant dynamics.

As an example, recall the three benchmark models introduced in Figure 1, whose sparsity patterns were given in Figure 2. The first model represents the component 1R of the International Space Station (ISS)[6], which has the state-space dimension $n = 270$ as well as three inputs and outputs $m = p = 3$ [19]. The second model represents a vibrating micro-mechanical butterfly gyroscope (Gyro)[7] of order $n = 34722$ with $m = 1$ input and $p = 12$ outputs [20]. Lastly, the third model represents a thermal problem describing the cooling process of a steel profile (Rail)[8] [21, 22]. There are different model sizes available, ranging from $n = 1357$ to $n = 79841$, with $m = 7$ inputs and $p = 6$ outputs.

In the following figure, three reduced order models are presented to approximate the benchmark models given above. They are obtained with three different MOR techniques, namely *truncated balanced realization* (`tbr`, also known under the name *balanced truncation*), *rational Krylov* subspace method (`rk`) and *Iterative Rational Krylov Algorithm* (`irka`). The magnitude responses of full and reduced order models are plotted in Figure 4 for the SISO transfer function from the respective first input to the first output using `bodemag`.

As it can be seen already from these examples, the relevant dynamics of the large-scale models can be represented by reduced order models of significantly lower order. The main difficulty in the reduction is to find appropriate projection matrices V, W in an efficient manner. This task is taken on by the reduction algorithms in **sssMOR**.

3.1 `ssRed` – a new class for reduced order models

As ROMs are generally described by dense matrices, there is no advantage in defining them as `sss`-objects. Instead, **sssMOR** contains a new subclass of `ss`, namely `ssRed`, which inherits all properties of `ss`-objects and can therefore be used in all built-in functions. In addition, information about the reduction is stored as a property `sysr.reductionParameters`. As one model may result

5 Available at http://dx.doi.org/10.5281/zenodo.49542

6 Available at http://slicot.org/20-site/126-benchmark-examples-for-model-reduction.

7 Available at https://portal.uni-freiburg.de/imteksimulation/downloads/benchmark/The%20Butterfly%20Gyro%20(35889).

8 Available at https://portal.uni-freiburg.de/imteksimulation/downloads/benchmark/Steel%20Profiles%20(38881).

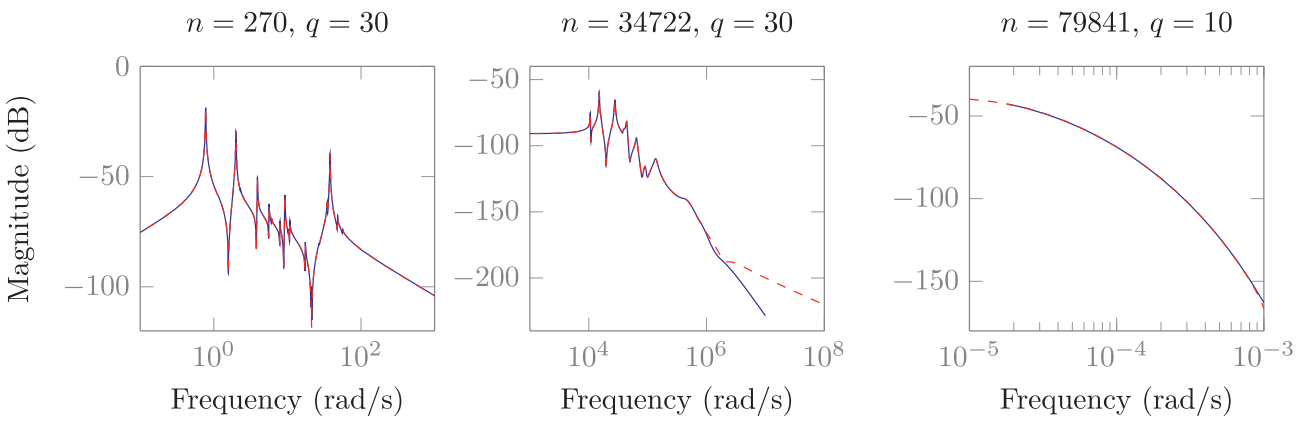


Figure 4: Reduction results for the benchmark models in Figure 1. The blue solid line represents the FOM, the red dashed line the ROM.

from a sequence of k reduction steps (compare e.g. `cure`), the property is a k-dimensional structure containing all parameters involved in the reduction of the FOM. Here is a minimal example showing the reduction parameters of a model obtained through `tbr`:

```
» sys = loadSss('iss'); sysr = tbr(sys,30);
» class(sysr)
ans =

ssRed

» sysr.reductionParameters{1}

ans =

method: 'tbr'
params: [1x1 struct]

» sysr.reductionParameters{1}.params

ans =

originalOrder: 270
type: 'tbr'
redErr: 0
hsvTol: 1.0000e-15
lse: 'gauss'
hsv: [270x1 double]
```

sssMOR contains classic reduction methods such as modal truncation, truncated balanced realizations and rational Krylov subspace methods. At the same time, state-of-the-art algorithms like the *iterative rational Krylov algorithm* (IRKA) and more recent algorithms such as *cumulative reduction framework* (CURE) and the *stability-preserving, rational Krylov algorithm* (SPARK) are implemented. The implementation is strongly coupled with the ongoing research at the Chair of Automatic Control in Munich. In addition, methods developed at other institutions are also frequently updated for benchmarking. In the following, we introduce the main reduction functions available in **sssMOR**.

3.2 Classic methods

The methods presented in this section can be categorized as the established methods that have been studied and developed over the past decades and are widespread in the MOR community. As a detailed explanation of the methods exceeds the scope of this treatise, we point out to comprehensive overviews on model reduction [23–28], whereas method-specific references will be given in the following.

3.2.1 Modal truncation

One very common approach to model reduction is given by preserving the most dominant eigenmodes and truncating the others. The projection matrices V and W span the right and left invariant subspaces corresponding to the eigenvalues that should be preserved. These are typically eigenvalues with smallest magnitude, but a dominance analysis according to Litz [29] can be included to take into account controllability and observability of the eigenmodes. MOR approaches based on eigenmode preservation are quite common in engineering fields such as structural mechanics, as they have a direct physical interpretation.

modalMor	compute a ROM by modal truncation
In	`sss`-object, reduced order
Out	`ssRed`-object, V, W
Opt(s)	Eigenvalues to be preserved (`'sm'`,`'lm'`,...); choice of real and orthogonal projection matrices; dominance analysis

3.2.2 Truncated balanced realization

One of the gold standards of model reduction, this techniques is based on preserving the directions in the state-space that are most relevant in terms of *generalized energy transfer* between inputs and outputs. From a theoretical standpoint, the method is based on finding a balanced realization, where all states are equally controllable and observable. This requires the computation of *controllability* and *observability Gramians* related to the solutions of generalized Lyapunov equations:

$$APE^\top + EPA^\top + BB^\top = 0, \tag{8a}$$
$$A^\top QE + E^\top QA + C^\top C = 0. \tag{8b}$$

The singular values of the product $P(E^\top QE)$ are system invariant (*Hankel singular values*) and their decay can be used to determine a valid reduced order q. An upper bound on the approximation error can also be obtained.

For very-large-scale models, equations (8) cannot be solved directly anymore due to storage limitations and computational expense. Instead, the indirect methods introduced in Section 2.3.2 need to be used [13, 30, 31]. Both direct and indirect methods are implemented in `tbr`, whereas the MATLAB built-in function `balancmr` can only deal with small to mid-sized problems.

Since the efficient solution of large-scale Lyapunov equations of type (8) has been (and still is) focus of intense research, `tbr` could also count as a state-of-the-art method.

tbr	compute a ROM by truncated balanced realization
In	`sss`-object, reduced order
Out	`ssRed`-object, HSV
Opt(s)	Direct or indirect (ADI) solution of (8); desired tolerance on reduction error

3.2.3 Rational Krylov

A different choice of projection matrices V, W that has drawn increased attention over the past decades is given by the *rational input and output Krylov subspaces*, respectively of the form

$$K_q\left((A-\sigma E)^{-1}E, (A-\sigma E)^{-1}B\right) \subseteq \mathrm{Ran}(V), \tag{9a}$$
$$K_q\left((A-\mu E)^{-\top}E^\top, (A-\mu E)^{-\top}C^\top\right) \subseteq \mathrm{Ran}(W). \tag{9b}$$

In fact, it can be shown [32] that the resulting reduced order model interpolates the original transfer function $G(s)$ and its derivatives at the approximation frequencies σ and μ, also called *shifts* or *expansion points*. Bases for rational Krylov subspaces can be computed both efficiently (as they require only the solution of sparse LSEs) and numerically robust by using *Arnoldi* or *Lanczos* processes [33]. In **sssMOR** the bases V and W are computed using the `arnoldi` function. The LSE involved are computed using `solveLse` (cp. Section 2.3.2) to allow a deeper customization.

rk	reduction by rational Krylov subspace methods
In	`sss`-object, input/output shifts, (tangential directions)
Out	`ssRed`-object, V, W
Opt(s)	real, orthogonal, re-orthogonalization, ...

Other related sssMOR functions

`projectiveMor`, `moments`, `isH2opt`

3.3 State-of-the-art methods

In this section, we present model reduction functions developed in recent years, most of which are aimed at improving classic methods and choose the reduction parameters in a judicious way.

3.3.1 IRKA

One of the major questions involved in the reduction by RK-methods is how to choose the reduction parameters, namely *shifts* and, in the MIMO case, *tangential directions*. A very simple yet effective method to adaptively determine these parameters is given by the Iterative RK Algorithm (IRKA) [34]. The fix-point iteration involved yields at convergence a ROM $G_r(s)$ that minimizes—at least locally—the approximation error measured in the $\mathcal{H}_2$-norm, i.e.

$$G_r = \arg\min_{\deg H = q} \|G - H\|_{\mathcal{H}_2}. \tag{10}$$

Expensive computations, like $\mathcal{H}_2$-norms of large-scale models are avoided at all times.

irka	reduction by iterative rational Krylov algorithm
In	sss-object, initial input/output shifts, (tangential directions)
Out	ssRed-object, V, W, optimal shifts (and tangential directions), iterations
Opt(s)	maxiter, tol, stopping criterion, stabilization, ...

3.3.2 CURE

One major challenge in MOR is given by the selection of an appropriate reduced order q. Many methods rely on an a-priori choice of q and require a completely new reduction if the reduced order was not chosen properly (cp. IRKA). CUmulative REduction (CURE) is based on model reduction by Krylov subspace methods and allows the adaptive choice of reduced order by cumulatively updating the ROM [15, 35, 36]. The method is based on a factorization of the approximation error

$$G_e = G - G_r = G_\perp \cdot \widetilde{G}_r. \tag{11}$$

Provided the reduction error is still too large, the reduced model can be updated iteratively following the pattern

$$\begin{aligned} G &= G_r^k + G_\perp^k \cdot \widetilde{G}_r^k \\ &= \underbrace{G_r^k + G_r^{k+}}_{G_r^{k+1}} + G_\perp^{k+1} \cdot \widetilde{G}_r^{k+1} \end{aligned} \tag{12}$$

The function cure can be combined with different Krylov-based reduction methods. In **sssMOR** we implemented the combination with rk, irka and spark, which can be selected through the optional parameters. Further combinations are possible and can be easily integrated individually by the users.

cure	cumulative reduction by rational Krylov methods
In	sss-object
Out	ssRed-object
Opt(s)	Reduction function (spark, irka,...); reduced order at each iteration; initial shifts; stopping criterion; ...

3.3.3 PORK

The task of finding a local $\mathcal{H}_2$-optimal reduced order model, e.g. by using irka, is non-convex and requires an iterative solution. On the other hand, the *global* $\mathcal{H}_2$-*pseudo*-optimal reduced model can be computed explicitly, for example by using the Pseudo-Optimal RK (PORK) algorithm [31, 37, 38]. $\mathcal{H}_2$-pseudo-optimality in this case means optimality within a subspace $\mathcal{G} \subseteq \mathcal{H}_2^{(p,m)}$, defined by the spectrum and residual directions:

$$G_r = \arg\min_{H \in \mathcal{G}} \|G - H\|_{\mathcal{H}_2}. \tag{13}$$

The PORK algorithm requires the basis of either an input or an output Krylov subspace (V or W), and is therefore implemented in the porkV and porkW functions. The functions are based on Sylvester equations for the Krylov subspaces, i.e. equations of the form

$$AV - EVS_v - BR_v = 0, \tag{14}$$

and require therefore in addition the input of the matrices S_v and R_v.

porkV	$\mathcal{H}_2$-pseudo-optimal reduction using input PORK
In	V, S_v, R_v, C
Out	ssRed-object
Opt(s)	-

3.3.4 SPARK

To optimize the subspace $\mathcal{G}$ in $\mathcal{H}_2$-pseudo-optimal reduction to obtain an $\mathcal{H}_2$-*optimal* reduced order model the Stability Preserving, Adaptive RK (SPARK) algorithm was developed [35, 36]. It is based on $\mathcal{H}_2$-pseudo optimal reduction and finds through a trust-region optimization, a pair of $\mathcal{H}_2$-optimal shift parameters. The restriction to two shifts (and hence reduced order q=2) is valid in combination with cure to obtain arbitrary reduced orders.

spark	stability preserving, adaptive $\mathcal{H}_2$-optimal reduction
In	sss-object, initial shifts
Out	ssRed-object
Opt(s)	algorithm type, maximum iterations and function evaluations, tolerances, ...

3.3.5 CIRKA

The cost of each iteration in $\mathcal{H}_2$-optimization, e.g. in irka or spark, is weighted with the full cost of one rk reduction. In order to decouple the cost of optimization from the cost of reduction, the *model function framework* has been recently presented [36, 39]. When applied to irka, this yields a new variant called *Confined IRKA*, which generally leads to substantial speedups, especially for very large-scale models.

cirka	$\mathcal{H}_2$-optimal reduction by Confined IRKA
In	sss-object, initial shifts
Out	ssRed-object
Opt(s)	model function parameters, maximum iterations, tolerance ...

Other related sssMOR functions

`rkIcop, rkOp, isrk, getSylvester, modelFct, modelFctMor`

3.4 Model Reduction App

The basic functionality of the **sss** and **sssMOR** toolboxes can be used also from within the `sssMOR_App`. The graphical user interface allows an easy and intuitive interaction with the toolbox and is meant to facilitate the usage of the system analysis and reduction functions. Note that only the main functionalities of **sss** and **sssMOR** are available in the app. In a nutshell, the user can load and define `sss` models, perform MOR by classic methods, visualize the system responses in time and frequency domain as well as compare models quantitatively in terms of $\mathcal{H}_2$ and $\mathcal{H}_\infty$ norms. Figure 5 shows a screenshot of the model reduction section of the app.

4 Numerical examples

In the following, we present a few numerical examples demonstrating the main functionality of the **sss** and **sssMOR** toolboxes. In addition, we perform a comparison to the Control System Toolbox to demonstrate the relevance of using sparse methods and model reduction when dealing with large-scale models. All computations were conducted using MATLAB® R2015b on an Intel® Core™ i7-2640M CPU @2.80 GHz with 8.00 GB RAM.

4.1 sss and sssMOR – natural extensions of the Control System Toolbox

One of our goals in the development of **sss** and **sssMOR** is to recover (most of) the functionality of the Control System Toolbox when dealing with large-scale models. This fact is shown by following example, in which the same function `yourAnalysisFunction`, written for `ss` objects, can be called with `sss` and `ssRed` objects exploiting the capabilities of **sss** and **sssMOR**.

```
%sssMOR_paper_demo1.m
% Run same analysis for ss, sss and ssRed objects
%% Load benchmark example
clear, clc
sysName = 'iss';
load(sysName)
%% System definition
desClass = 'ss';
switch desClass
    case 'ss'
        sys     = dss(A,B,C,[],[]);
    case 'sss'
        sys     = sss(A,B,C,[],[]);
    case 'ssRed'
        sysFOM  = sss(A,B,C,[],[]);
        sys     = tbr(sysFOM,50);
end
%% Run analysis code
yourAnalysisFunction(sys);
```

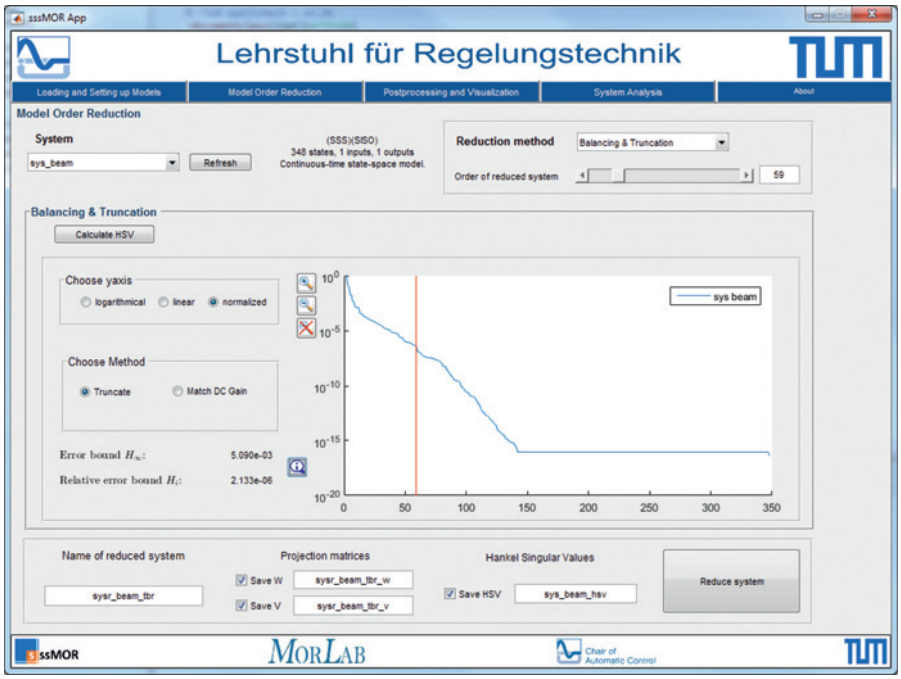


Figure 5: Screenshot of the `sssMOR_App`.

An example of analysis function is given in the following:

```
function yourAnalysisFunction(sys)
% Analyze LTI model
%% Model order
n = size(sys.A,1);
%% Memory requirement to store sys
info    = whos('sys');
reqMem  = info.bytes
%% Check stability of the given model
stabCheck = isstable(sys)
%% Bode magnitude plot
figure; bodemag(sys)
%% H2 and Hinf norms
h2norm = norm(sys)
h8norm = norm(sys,Inf)
```

To exemplify the impact of using **sss** and **sssMOR** for system analysis, Table 1 summarizes the results obtained for the benchmark model `rail_1357.mat` using all three model classes.

As it can be seen, using **sss** and `sss` objects already yields a major improvement in terms of memory and computational requirements with respect to `ss`, even for a model of modest size $n = 1357$. By applying MOR (in this case `tbr`), (basically) the same results can be obtained with significant speedup.

Table 1: Comparison of analysis of the Rail benchmark ($n = 1357$) for different system classes.

		ss	sss	ssRed
order		1357	1357	50
reduction (`tbr`)	time [s]	–	–	2.89
memory	[MB]	29.61	0.32	0.05
`isstable`	time [s]	28.87	0.23	0.01
`bode`	time [s]	31.69	1.06	0.63
`norm(sys)`	value	0.0046	0.0046	0.0046
	time [s]	62.20	54.34	0.01
`norm(sys,inf)`	value	0.3523	0.3523	0.3539
	time [s]	188.19	6.47	0.01

Table 2: Analysis of the Gyro benchmark ($n = 34722$) for different system classes.

		ss	sss	ssRed
order		34 722	34 722	40
reduction (`irka`)	time [s]	–	–	105.20
memory	[MB]	> 9000	39.51	0.07
`bode`	time [s]	–	646.16	0.93
`norm(sys)`	value	–	0.2103	0.2103
	time [s]	–	347.21	0.01
`norm(sys,inf)`	value	–	0.0248	0.0248
	time [s]	–	1534.3	0.04

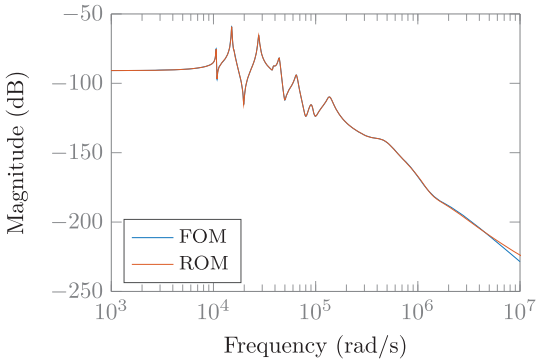


Figure 6: Bode magnitude plot for full and reduced order models for the Gyro benchmark ($n = 34722, q = 40$).

Table 2 shows the results for the `gyro.mat` benchmark model. Due to the relatively high order $n = 34722$, the analysis with MATLAB was not possible, as the respective `ss` object would require more than 9 GB of memory to be stored. The Bode magnitude plot comparing full and reduced order models is given in Figure 6 for the channel between first input and first output (`sys(1,1)`).

4.2 Performance comparison of MOR methods

In this section we compare execution time and reduction quality of different reduction methods available in the toolbox. This can help to get a first idea on what algorithms to choose, even though we stress out that the reduction method of choice highly depends on the desired characteristics of the ROM and the FOM at hand. As benchmark model we chose the Rail example of full order $n = 1357$. To simplify the graphical representation, we restrict our considerations to the SISO subsystem `sys(1,4)` from the 4$^{\text{th}}$ input to the 1$^{\text{st}}$ output.

Table 3 summarizes the results quantitatively, while Figure 7 gives a qualitative comparison by showing the Bode magnitude plots (`bodemag`) for full and reduced order models. For all methods a reduced order $q = 10$ was chosen. The reduction by `modalMor` was conducted preserving eigenmodes corresponding to the first q eigenvalues with smallest magnitude. The reduction by `rk` was conducted matching $2q$ moments at the origin ($\sigma = \mu = 0$). The algorithms `irka`, `rkIcop` and `cure` were also initialized at $\sigma = 0$.

Table 3: Comparison of some reduction methods provided in **sssMOR**.

method	time [s]	$\frac{\|G-G_r\|_{\mathcal{H}_2}}{\|G\|_{\mathcal{H}_2}}$	$\frac{\|G-G_r\|_{\mathcal{H}_\infty}}{\|G\|_{\mathcal{H}_\infty}}$
`modalMor`	0.18	5.628e−02	7.625e−03
`tbr`	0.39	4.966e−02	4.298e−02
`rk`	0.09	4.848e−04	1.951e−05
`irka`	0.28	1.095e−04	3.624e−06
`rkIcop`	0.19	4.492e−04	1.835e−05
`cure` (with `spark`)	4.18	1.811e−02	8.572e−03

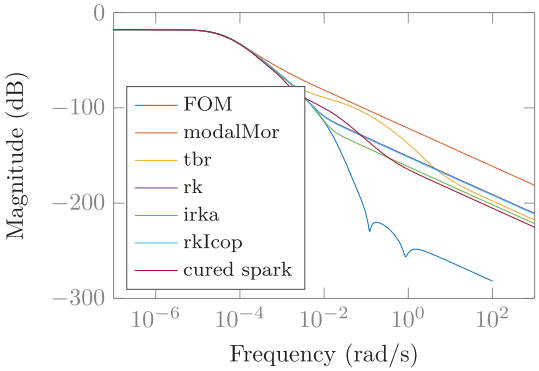


Figure 7: Bode magnitude plot for full and reduced order models for the Rail benchmark ($n = 1357, q = 10$).

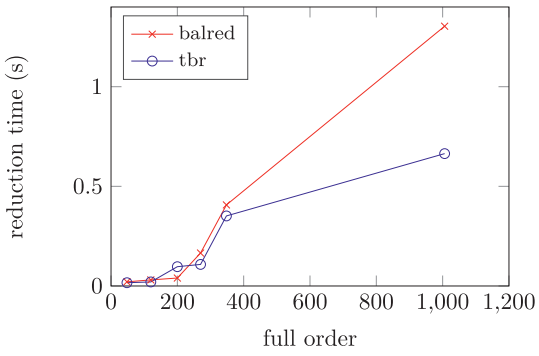


Figure 8: Comparision between `balred` and `tbr`.

As it can be seen from comparing the data, all reduction methods are able to efficiently produce high quality ROMs at a low computational cost. The rational Krylov methods (`rk`, `irka`, `rkIcop`) seem to perform the best for this example both in terms of computational cost and approximation quality. The `cure` algorithm run in combination with `spark` repeatedly performs $\mathcal{H}_2$-optimal reduction to order $q = 2$, hence explaining the relatively high execution time. Nonetheless, note that it is the only Krylov method able to choose the reduced order on the go.

Finally, we add a small comparison between `balred`, the Control System Toolbox built-in function for reduction by balanced truncation, and the `tbr` function in **sssMOR**. The comparison is conducted for models of relatively small size[9] and a reduction to order $q = 10$. The (averaged) execution times in Figure 8 demonstrate that using `tbr` can be advantageous even for small models.

4.3 Reduction of large-scale models – a brief tutorial

Now that you can download and use **sss** and **sssMOR** for free, how can you effectively use the tools to analyze and reduce a given high-order model? To get you started we provide a tutorial, i.e. an example on how to reduce a given large-scale model without prior knowledge of the underlying dynamics. Since an appropriate choice of reduction methods and parameters highly depends on the model at hand, we present a general scheme that might be helpful to get started with model reduction. You can find the tutorial on our website http://www.rt.mw.tum.de/?sssMOR.

5 Conclusions

In this contribution we gave a first presentation of the **sss** and **sssMOR** toolboxes, free and open-source tools for the analysis and reduction of large-scale models in MATLAB. We have demonstrated that the built-in capability of the Control System Toolbox is limited to small and mid-sized problems, making it impossible to even define and store large-scale models. The main reason for this is that the sparsity, which characterizes the system matrices in large-scale applications, is not exploited.

Based on this problem, we have developed the **sss** (sparse state-space) toolbox to define and analyze state-space models of very large size ($n \gg 10^4$). We have presented the main functionality of **sss** and stressed that many of the functions available in the Control System Toolbox have been reprogrammed to exploit sparsity. This makes **sss** a natural extension to the Control System Toolbox for large-scale problems, with the aim of making old routines developed for `ss` objects work efficiently also for `sss` objects.

In addition, we have presented the **sssMOR** toolbox, entailing both classic and state-of-the-art reduction methods such as `modalMor`, `tbr`, `rk`, `irka`, ...to efficiently re-

9 Models: build ($n = 48$), CDplayer ($n = 120$), heat-cont ($n = 200$), iss ($n = 270$), beam ($n = 348$), fom ($n = 1006$) [19].

duce the order of a given model while approximating the dynamics with high fidelity.

The main computational burden in the analysis and reduction of large-scale models is given by the solution of large, sparse linear systems of equations or large-scale matrix equations. Techniques to efficiently handle these computations are implemented centrally in the functions `solveLse` and `lyapchol` respectively. It is therefore easy to select the desired method for the problem at hand, or even individually include specific solvers within the toolbox.

The description of the tools has been complemented by numerical results, showing the significant advantage given by using `sss` and `ssRed` objects when dealing with large-scale models.

5.1 Outlook

Both **sss** and **sssMOR** toolboxes are being actively developed at MORLab, the MOR laboratory of the Chair of Automatic Control, Technical University of Munich. As such, their functionalities are constantly improved and updated along with the theoretic results achieved in research.

Current toolbox developments include the analysis and reduction of *parametric* models [40–42], as well as *data-driven* and $\mathcal{H}_\infty$-optimal model reduction of MIMO models. To keep up to date on new developments, check our website[10] and our page on MATLAB central file exchange[11], or sign up for our newsletter under https://lists.lrz.de/mailman/listinfo/sssmor.

5.2 Third-party software

sss and **sssMOR** make use of openly available third-party software to complement their functionalities. At the moment, the only third-party software involved is M-M.E.S.S.-1.0[12], the Matrix Equation Sparse Solver for MATLAB developed by Jens Saak, Martin Köhler and Peter Benner at the Max Planck Institute for Dynamics of Complex Technical Systems in Magdeburg[13] [18]. It is used for finding approximate (low-rank) solutions to large-scale Lyapunov equations of the form (8) by means of the Alternating Directions Implicit (ADI) algorithm. This step is required both in computing the $\mathcal{H}_2$-norm of an `sss` object and performing model reduction by `tbr`. We thank the authors of M-M.E.S.S. for developing the toolbox and making it publicly available, as well as for the friendly exchange over the past years.

Funding: The work related to this contribution is supported by the German Research Foundation (DFG), Grant LO408/19-1.

Acknowledgement: The first line of code of **sss** dates as far back as 2011 and resulted from the vision and programming knowledge of Heiko Peuscher (né Panzer) [36], Rudy Eid [43], and Sylvia Cremer, all former researchers and/or students at MORLab. To them goes our greatest gratitude for laying the foundations of what **sss** and **sssMOR** (even `sssMOR_App`) are today. Without such a valuable basis, we would probably not have started this undertaking.

In 2014 we decided to complete the toolbox and make it available to students and researches around the world. We expanded the functionality, created new functions and polished everything by adding a comprehensive documentation. To deal with the ever growing size and complexity of the project, we implemented a unit test environment to ensure early bug detection. In these endeavors, several students have been involved, to whom we send our gratitude for their dedication to the project and their hard work: Jorge Luiz Moreira Silva, Rodrigo Mancilla, Siyang Hu, Niklas Kochdumper and Max Gille.

We are also sincerely thankful to Philip Holzwarth and Nico Walz of the Institute of Engineering and Computational Mechanics at University of Stuttgart[14] for sharing with us the functions they use in the MOREMBS[15] toolbox for automatically generating the MATLAB documentation. Without this wonderful piece of code, our documentation would probably be less articulated.

Last but not least, we would like to thank Stefan Jaensch and Thomas Emmert, former researchers at the Professur für Thermofluiddynamik[16], for recognizing the potential of **sss** as a stand-alone toolbox and the fruitful and at all times friendly cooperation to jointly develop it. Today, **sss** is integrated in their taX tool[17] for simulating large-scale thermoacoustic networks [7].

10 www.rt.mw.tum.de/?sssMOR

11 https://de.mathworks.com/matlabcentral/fileexchange/59169-sssmor-toolbox

12 http://dx.doi.org/10.5281/zenodo.49542

13 http://www.mpi-magdeburg.mpg.de/csc

14 http://www.itm.uni-stuttgart.de/index_en.php

15 http://www.itm.uni-stuttgart.de/research/model_reduction/MOREMBS_en.php

16 https://www.tfd.mw.tum.de/index.php?id=5&L=1

17 https://tax.wiki.tum.de/

References

1. Timothy A Davis. *Direct methods for sparse linear systems*, volume 2. Siam, 2006.
2. Iain S. Duff, Albert Maurice Erisman, and John Ker Reid. *Direct methods for sparse matrices*. Clarendon press Oxford, 1986.
3. Yousef Saad. *Iterative methods for sparse linear systems*. Society for Industrial and Applied Mathematics, 2003.
4. John R Gilbert, Cleve Moler, and Robert Schreiber. Sparse matrices in MATLAB: design and implementation. *SIAM Journal on Matrix Analysis and Applications*, 13(1):333–356, 1992.
5. Alessandro Astolfi. A new look at model reduction by moment matching for linear systems. In *46th IEEE Conference on Decision and Control*, pages 4361–4366. IEEE, 2007.
6. Tudor C Ionescu, Alessandro Astolfi, and Patrizio Colaneri. Families of moment matching based, low order approximations for linear systems. *Systems & Control Letters*, 64:47–56, 2014.
7. Thomas Emmert, Stefan Jaensch, Carlo Sovardi, and Wolfgang Polifke. tax – A flexible tool for low-order duct acoustic simulation in time and frequency domain. In *Forum Acusticum, Krakow, Poland, Sept*, pages 7–12, 2014.
8. Thomas Emmert, Maximilian Meindl, Stefan Jaensch, and Wolfgang Polifke. Linear state space interconnect modmodel of acoustic systems. *Acta Acustica United With Acustica*, 98, 2016.
9. Gene H. Golub and Charles F. Van Loan. *Matrix computations*, volume 3. JHU Press, 2012.
10. Sven Hammarling. Numerical solution of the stable, nonnegative definite Lyapunov equation. *IMA Journal of Numerical Analysis*, 2(3):303–323, 1982.
11. Thilo Penzl. Numerical solution of generalized Lyapunov equations. *Advances in Computational Mathematics*, 8(1–2):33–48, 1998.
12. Jing-Rebecca Li and Jacob White. Low rank solution of Lyapunov equations. *SIAM Journal on Matrix Analysis and Applications*, 24(1):260–280, 2002.
13. Jens Saak. *Efficient Numerical Solution of Large Scale Algebraic Matrix Equations in PDE Control and Model Order Reduction*. PhD thesis, Chemnitz University of Technology, 2009.
14. Patrick Kürschner. *Efficient Low-Rank Solution of Large-Scale Matrix Equations*. PhD thesis, Otto-von-Guericke Universität Magdeburg, 2016.
15. Thomas Wolf, Heiko K.F. Panzer, and Boris Lohmann. ADI iteration for Lyapunov equations: A tangential approach and adaptive shift selection. *Applied Numerical Mathematics*, 109:85–95, 2016.
16. Valeria Simoncini. A new iterative method for solving large-scale Lyapunov matrix equations. *SIAM Journal on Scientific Computing*, 29(3):1268–1288, 2007.
17. Vladimir Druskin, Valeria Simoncini, and Mikhail Zaslavsky. Adaptive tangential interpolation in rational Krylov subspaces for MIMO dynamical systems. *SIAM J. Matrix Anal. & Appl.*, 35(2):476–498, 2014.
18. Jens Saak, Martin Köhler, and Peter Benner. M-M.E.S.S.-1.0 – The Matrix Equation Sparse Solver Library, April 2016.
19. Younes Chahlaoui and Paul Van Dooren. A collection of benchmark examples for model reduction of linear time invariant dynamical systems. Working Note 2002-2, 02 2002.
20. Jan Lienemann, Dag Billger, Evgenii B. Rudnyi, Andreas Greiner, and Jan G. Korvink. MEMS compact modeling meets model order reduction: Examples of the application of arnoldi methods to microsystem devices. In *The Technical Proceedings of the 2004 Nanotechnology Conference and Trade Show, Nanotech*, volume 4, 2004.
21. Andreas Unger and Fredi Tröltzsch. Fast solution of optimal control problems in the selective cooling of steel. *Zeitschrift für Angewandte Mathematik und Mechanik ZAMM*, 81(7):447–456, 2001.
22. Jens Saak. Effiziente numerische Lösung eines Optimalsteuerungsproblems für die Abkühlung von Stahlprofilen. Master's thesis, Diplomarbeit, Fachbereich 3/Mathematik und Informatik, Universität Bremen, D-28334 Bremen, 2003.
23. Athanasios C. Antoulas. *Approximation of Large-Scale Dynamical Systems*. SIAM, 2005.
24. Otto Föllinger. *Regelungstechnik*. VDE Verlag, 2013.
25. Athanasios C. Antoulas, Danny Sorensen, and Serkan Gugercin. A survey of model reduction methods for large-scale systems. *Contemporary mathematics*, 280:193–220, 2001.
26. Serkan Gugercin and Athanasios C. Antoulas. A survey of model reduction by balanced truncation and some new results. *International Journal of Control*, 77(8):748–766, 2004.
27. Peter Benner and Tatjana Stykel. Model order reduction for differential-algebraic equations: a survey. Preprint MPIMD/15-19, Max Planck Institute Magdeburg, November 2015. Available from http://www.mpi-magdeburg.mpg.de/preprints/.
28. Ulrike Baur, Peter Benner, and Lihong Feng. Model order reduction for linear and nonlinear systems: a system-theoretic perspective. *Archives of Computational Methods in Engineering*, 2014.
29. L. Litz and H. Roth. State decomposition for singular perturbation order reduction – A modal approach. *International Journal of Control*, 34(5):937–954, 1981.
30. Jens Saak, Peter Benner, and Patrick Kürschner. A goal-oriented dual LRCF-ADI for balanced truncation. In *Vienna Conference on Mathematical Modelling (MATHMOD)*, 2012.
31. Thomas Wolf. $\mathcal{H}_2$ *Pseudo-Optimal Model Order Reduction*. PhD thesis, Technische Universität München, 2014.
32. Christian De Villemagne and Robert E. Skelton. Model reductions using a projection formulation. *International Journal of Control*, 46(6):2141–2169, 1987.
33. Eric J. Grimme. *Krylov Projection Methods for Model Reduction*. PhD thesis, Dep. of Electrical Eng., Uni. Illinois at Urbana Champaign, 1997.
34. Serkan Gugercin, Athanasios C. Antoulas, and Christopher Beattie. $\mathcal{H}_2$ model reduction for large-scale linear dynamical systems. *SIAM Journal on Matrix Analysis and Applications*, 30(2):609–638, 2008.
35. Heiko K. F. Panzer, Stefan Jaensch, Thomas Wolf, and Boris Lohmann. A greedy rational Krylov method for $\mathcal{H}_2$-pseudooptimal model order reduction with preservation of stability. In *American Control Conference*, pages 5532–5537, 2013.
36. Heiko K. F. Panzer. *Model Order Reduction by Krylov Subspace Methods with Global Error Bounds and Automatic Choice of Parameters*. PhD thesis, Technische Universität München, 2014.
37. Christopher Beattie and Serkan Gugercin. Realization-independent $\mathcal{H}_2$-approximation. In *51st IEEE Conference on Decision and Control*, pages 4953–4958. IEEE, 2012.

38. Thomas Wolf, Heiko K. F. Panzer, and Boris Lohmann. $\mathcal{H}_2$ pseudo-optimality in model order reduction by Krylov subspace methods. In *European Control Conference*, 2013.
39. Alessandro Castagnotto, Heiko K. F. Panzer, and Lohmann Boris. Fast $\mathcal{H}_2$-optimal model order reduction exploiting the local nature of Krylov-subspace methods. In *European Control Conference*, pages 1958–1963, Aalborg, 2016.
40. Peter Benner, Serkan Gugercin, and Karen Willcox. A survey of model reduction methods for parametric systems. Preprint MPIMD/13-14, Max Planck Institute Magdeburg, 2013.
41. Heiko Panzer, Jan Mohring, Rudy Eid, and Boris Lohmann. Parametric model order reduction by matrix interpolation. *at–Automatisierungstechnik*, 58(8):475–484, 8 2010.
42. Matthias Geuss. *A Black-Box Method for Parametric Model Order Reduction based on Matrix Interpolation with Application to Simulation and Control*. PhD thesis, Technische Universität München, 2015.
43. Rudy Eid. *Time domain model reduction by moment matching*. PhD thesis, Technische Universität München, 2009.

Bionotes

Alessandro Castagnotto
Technical University of Munich, Department of Mechanical Engineering, Chair of Automatic Control, Boltzmannstr. 15, 85748 Garching, Germany
a.castagnotto@tum.de

Maria Cruz Varona
Technical University of Munich, Department of Mechanical Engineering, Chair of Automatic Control, Boltzmannstr. 15, 85748 Garching, Germany

Lisa Jeschek
Technical University of Munich, Department of Mechanical Engineering, Chair of Automatic Control, Boltzmannstr. 15, 85748 Garching, Germany

Boris Lohmann
Technical University of Munich, Department of Mechanical Engineering, Chair of Automatic Control, Boltzmannstr. 15, 85748 Garching, Germany