Structure-Preserving Model Reduction for Mechanical Systems

Dissertation

zur Erlangung des akademischen Grades

doctor rerum naturalium
(Dr. rer. nat.)

von **Steffen W. R. Werner, M.Sc.**

geb. am **September 6, 1992** in Stendal, Germany

genehmigt durch die Fakultät für Mathematik

der Otto-von-Guericke-Universität Magdeburg

Gutachter: **Prof. Dr. Peter Benner**

Prof. Dr. Serkan Gugercin

Prof. Dr. Tobias Damm

eingereicht am: **27.04.2021**

Verteidigung am: **19.08.2021**

PUBLICATIONS

Several parts of this thesis have been published in papers or are available in preprint form as summarized below.

Chapter 4 is a rearranged, partly revised and extended version of the results in:

[26]: R. S. Beddig, P. Benner, I. Dorschky, T. Reis, P. Schwerdtner, M. Voigt, and S. W. R. Werner, *Model reduction for second-order dynamical systems revisited*, Proc. Appl. Math. Mech., 19 (2019), p. e201900224, https://doi.org/10.1002/pamm.201900224

[27]: R. S. Beddig, P. Benner, I. Dorschky, T. Reis, P. Schwerdtner, M. Voigt, and S. W. R. Werner, *Structure-preserving model reduction for dissipative mechanical systems*, e-print 2010.06331, arXiv, 2020, https://arxiv.org/abs/2010.06331

[54]: P. Benner and S. W. R. Werner, *Frequenz- und zeitbeschränktes balanciertes Abschneiden für Systeme zweiter Ordnung*, in Tagungsband GMA-FA 1.30 'Modellbildung, Identifikation und Simulation in der Automatisierungstechnik' und GMA-FA 1.40 'Systemtheorie und Regelungstechnik', Workshops in Anif, Salzburg, 23.-27.09.2019, T. Meurer and F. Woittennek, eds., 2019, pp. 460–474.

[57]: P. Benner and S. W. R. Werner, *Frequency- and time-limited balanced truncation for large-scale second-order systems*, Linear Algebra Appl., 623 (2021), pp. 68–103, https://doi.org/10.1016/j.laa.2020.06.024, Special issue in honor of P. Van Dooren, Edited by F. Dopico, D. Kressner, N. Mastronardi, V. Mehrmann, and R. Vandebril.

[168]: J. Saak, D. Siebelts, and S. W. R. Werner, *A comparison of second-order model order reduction methods for an artificial fishtail*, at-Automatisierungstechnik, 67 (2019), pp. 648–667, https://doi.org/10.1515/auto-2019-0027

However, various parts of Section 4.1, as well as most parts of the presented numerical results are not published in any form yet. Most of the methods described in Chapter 4 have been implemented and published in the following three MATLAB toolboxes:

> [55]: P. Benner and S. W. R. Werner, *MORLAB – Model Order Reduction LABoratory (version 5.0)*, 2019, https://doi.org/10.5281/zenodo.3332716, see also: https://www.mpi-magdeburg.mpg.de/projects/morlab

> [58]: P. Benner and S. W. R. Werner, *SOLBT – Limited balanced truncation for large-scale sparse second-order systems (version 3.0)*, 2021, https://doi.org/10.5281/zenodo.4600763

> [59]: P. Benner and S. W. R. Werner, *SOMDDPA – Second-Order Modally-Damped Dominant Pole Algorithm (version 2.0)*, 2021, https://doi.org/10.5281/zenodo.3997649

Chapter 5 contains partly modified and revised versions of

> [42]: P. Benner, S. Gugercin, and S. W. R. Werner, *Structure-preserving interpolation for model reduction of parametric bilinear systems*, Automatica J. IFAC, 132 (2021), p. 109799, https://doi.org/10.1016/j.automatica.2021.109799

> [43]: P. Benner, S. Gugercin, and S. W. R. Werner, *Structure-preserving interpolation of bilinear control systems*, Adv. Comput. Math., 47 (2021), p. 43, https://doi.org/10.1007/s10444-021-09863-w

In particular, the theoretical and numerical results in Section 5.6 are not published in any form yet.

ABSTRACT

In this thesis, structure-preserving model order reduction for dynamical systems is studied. The particular focus lies on mechanical systems described by differential equations with second-order time derivatives. Different system classes are considered such as linear, bilinear and general nonlinear systems. Starting with the linear system case, existing theory from modal truncation and dominant poles is used to derive a new structure-preserving dominant pole algorithm for the special case of modally damped mechanical systems. Error bounds are proposed for this new method and an extension is suggested for further improvement of the approximation quality. In the sense of model order reduction with localized approximation behavior, structure-preserving extensions of the frequency- and time-limited balanced truncation methods for linear second-order systems are developed. Further approaches are discussed to counter the arising problem of stability preservation, and numerical methods are outlined to apply the model reduction methods to systems with large-scale sparse matrices. Moreover, the class of bilinear systems involving the multiplication of state and control variables is considered. Mainly motivated by the mechanical system case, a representation of structured bilinear systems in the frequency domain is developed. Considering the structured subsystem transfer functions as main object of interest, an interpolation framework is proven for structure-preserving model order reduction of these special nonlinear systems. Thereafter, this framework is extended to the case of structured parametric bilinear systems. Tangential interpolation can be used in case of linear multi-input/multi-output systems to carefully steer the resulting dimensions of constructed reduced-order models in contrast to the approach of matrix interpolation, which depends on the input and output dimensions of the original system. Based on different motivations, a similar theory for tangential interpolation is developed for structured bilinear systems. Structured systems with more general nonlinearities are considered last, where the process of quadratic-bilinearization is used to rewrite the systems into a form with easier manageable nonlinearities. Similar to the bilinear system case, a particular nonlinear mechanical system example is used to derive structured representations of quadratic-bilinear systems in the frequency domain. Based on that, a variety of transfer function interpolation results are developed for structure-preserving model reduction of quadratic-bilinear systems. Numerical experiments are used for all

introduced model reduction approaches to validate the developed theoretical results and compare them to known model reduction methods from the literature.

ZUSAMMENFASSUNG

Die vorliegende Arbeit befasst sich mit strukturerhaltender Modellordnungsreduktion für dynamische Systeme. Dabei liegt der besondere Schwerpunkt auf mechanischen Systemen mit Zeitableitungen zweiter Ordnung. Es werden verschiedene Systemklassen wie z.B. lineare, bilineare und Systeme mit allgemeineren Nichtlinearitäten betrachtet. Beginnend mit dem linearen Systemfall wird ein neuer strukturerhaltender Dominant-Pole-Algorithmus für modal gedämpfte, mechanische Systeme entwickelt. Dieser basiert auf bekannter Theorie über modales Abschneiden und dominante Pole. Es werden Fehlerschranken für diese Methode bewiesen und eine Erweiterung vorgeschlagen, um das Approximationsverhalten weiter zu verbessern. Im Sinne von Modellordnungsreduktion mit lokalisierter Approximation werden frequenz- und zeitbeschränktes balanciertes Abschneiden zu strukturerhaltenden Methoden für lineare Systeme zweiter Ordnung erweitert. Um dem Verlust der Stabilitätserhaltung entgegenzuwirken und um die Modellreduktionsmethoden auch im Fall von großen, dünnbesetzten Systemen zweiter Ordnung anwenden zu können werden weitere Ansätze diskutiert und numerische Verfahren skizziert. Des Weiteren wird die Klasse der bilinearen Systeme, welche das Produkt aus Zustands- und Steuerungsvariablen enthalten, betrachtet. Hauptsächlich motiviert durch den mechanischen Fall wird eine Darstellung von strukturierten, bilinearen Systemen im Frequenzbereich entwickelt. Zur strukturerhaltenden Modellreduktion dieser speziellen nichtlinearen Systeme wird ein Interpolationsansatz hergeleitet, bei welchem die strukturierten Übertragungsfunktionen als zu interpolierende Objekte betrachtet werden. Darauffolgend wird dieser Ansatz auf den Fall von strukturierten, parametrischen, bilinearen Systemen erweitert. Tangentiale Interpolation bietet im Fall von linearen Mehrgrößensystemen die Möglichkeit, die Dimensionen des konstruierten, reduzierten Modells besser zu kontrollieren, welche beim Ansatz der Matrixinterpolation an die Anzahl der Ein- und Ausgänge gebunden sind. Basierend auf verschiedenen Motivationsbeispielen wird eine ähnliche Theorie für strukturierte, bilineare Systeme entwickelt. Den Abschluss bildet die Betrachtung von Systemen mit allgemeineren Nichtlinearitäten. Es wird die Methode der quadratischen Bilinearisierung benutzt, um diese Systeme in eine Form umzuschreiben, welche einfachere Nichtlinearitäten beinhaltet. Ein spezielles nichtlineares, mechanisches Beispiel wird verwendet um ähnlich zum bilinearen Fall strukturierte Darstellungen im Frequenzbereich herzuleiten.

Es wird eine Vielzahl von Ergebnissen zur Übertragungsfunktionsinterpolation entwickelt, welche der strukturerhaltenden Modellordnungsreduktion von quadratisch bilinearen Systemen dienen. Numerische Experimente werden für alle entwickelten Modellreduktionsmethoden benutzt um sowohl die theoretischen Resultate zu validieren, als auch diese neuen Methoden mit anderen bekannten Modellreduktionsmethoden aus der Literatur zu vergleichen.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

LIST OF ACRONYMS

DAE	differential-algebraic equation
DEIM	discrete empirical interpolation method
EIM	empirical interpolation method
FOM	full-order model
IRKA	iterative rational Krylov algorithm
LR-ADI	low-rank alternating direction implicit
LTI	linear time-invariant
MIMO	multiple-input/multiple-output
ODE	ordinary differential equation
PDE	partial differential equation
POD	proper orthogonal decomposition
QBDAE	quadratic-bilinear differential-algebraic equation
QBODE	quadratic-bilinear ordinary differential equation
SISO	single-input/single-output
SOFLBT	second-order frequency-limited balanced truncation
SOMDDA	second-order modally damped dominant pole algorithm
SOTLBT	second-order time-limited balanced truncation
SVD	singular value decomposition
TF	transfer function
TF-IRKA	transfer function iterative rational Krylov algorithm

LIST OF SYMBOLS

$\mathbb{R}$, $\mathbb{C}$	fields of real and complex numbers
$\mathbb{R}_{>0}$, $\mathbb{R}_{\geq 0}$	positive and non-negative real numbers
$\mathbb{N}$, $\mathbb{N}_0$	set of natural numbers, $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$, respectively
$\mathbb{R}^n$, $\mathbb{C}^n$	vector spaces of real or complex valued tuples of length n
$\mathbb{R}^{n\times m}$, $\mathbb{C}^{n\times m}$	vector spaces of real or complex valued matrices with n rows and m columns
$\lvert\mathcal{I}\rvert$	cardinality of the set $\mathcal{I}$
$\mathfrak{i}$	imaginary unit ($\mathfrak{i}^2 = -1$)
$\operatorname{Re}(z)$, $\operatorname{Im}(z)$	real and imaginary parts of the complex number $z = \operatorname{Re}(z) + \mathfrak{i}\operatorname{Im}(z) \in \mathbb{C}$
$\overline{z}$	$:= \operatorname{Re}(z) - \mathfrak{i}\operatorname{Im}(z)$, conjugate of the complex number z
$\lvert z\rvert$	absolute value of real or complex number z
$\mathbb{1}_n$	vector with all entries 1 of length n
e_k	k-th column of the identity matrix of suitable size
I_n	identity matrix of size n
$\operatorname{diag}(d_1, \ldots, d_n)$	diagonal matrix of size n with entries $d_1, \ldots, d_n$
A^{T}	transposed of the matrix A
A^{H}	$:= \overline{A}^{\mathsf{T}}$, conjugate transposed of the matrix A
$\operatorname{span}(A)$	linear subspace spanned by the columns of the matrix A
A^{-1}	inverse of the regular matrix A
$A^{-\mathsf{T}}$, $A^{-\mathsf{H}}$	inverses of A^{T} or A^{H}, respectively
$\det(A)$	determinant of the matrix A
$\Lambda(A)$	spectrum of the matrix A
$\Lambda(E, A)$	spectrum of the matrix pencil $\lambda E - A$

$\Lambda(M, E, K)$	spectrum of the quadratic matrix pencil $\lambda^2 M + \lambda E + K$
$\sigma_{\max}(A)$	largest singular value of the matrix A
$\mathrm{vec}(A)$	vectorization of the matrix A ..... Definition 2.1
$A \otimes B$	Kronecker product of the matrices A and B . Definition 2.1
$\|.\|_2$, $\|.\|_\infty$	ℓ^2 or ℓ^∞ vector or induced matrix norm
$\|.\|_{L_2}$, $\|.\|_{L_\infty}$	time domain L_2- or L_∞-norm .. Equations (2.15) and (2.16)
$\|.\|_{\mathcal{H}_2}$, $\|.\|_{\mathcal{H}_\infty}$, $\|.\|_{\mathcal{L}_\infty}$	frequency domain $\mathcal{H}_2$-, $\mathcal{H}_\infty$- or $\mathcal{L}_\infty$-norm .... Definition 2.7
$\partial_s f$	partial derivative of f with respect to s ..... Equation (2.5)
∇f	Jacobi matrix of f ..... Equation (2.6)
$\dot{f}$	$:= \partial_t f$, derivative of f with respect to time
$\ddot{f}$	$:= \partial_{t^2} f$, second-order derivative of f with respect to time

CHAPTER 1

INTRODUCTION

Contents

1.1 Motivation

Almost all real-world phenomena and processes are nowadays described by systems of partial differential equations (PDEs), which relate physical quantities to their partial derivatives with respect to time and space. The most common approach to use these mathematical descriptions of the real world in computer-aided design processes and numerical experiments is a spatial discretization, usually via methods like finite elements or finite differences/volumes, leading to systems of ordinary differential equations (ODEs) or, in the presence of additional physical constraints such as conservation laws, to systems of differential-algebraic equations (DAEs). The resulting systems, which describe the time evolution of processes, are known as dynamical systems. In the presence of external forcing (inputs) and the observation of certain quantities of interests (outputs), dynamical systems can formally be written as

$$\begin{aligned} \mathsf{E}\dot{\mathsf{x}}(t) &= f(t, \mathsf{x}(t), u(t)), \\ y(t) &= g(t, \mathsf{x}(t)), \end{aligned} \tag{1.1}$$

with the solution trajectory $\mathsf{x}\colon \mathbb{R}_{\geq 0} \to \mathbb{R}^{n_1}$ described by a system of differential equations with the state-evolution function $f\colon \mathbb{R}_{\geq 0} \times \mathbb{R}^{n_1} \times \mathbb{R}^m \to \mathbb{R}^{n_1}$ and mass matrix $\mathsf{E} \in \mathbb{R}^{n_1 \times n_1}$. The inputs $u\colon \mathbb{R}_{\geq 0} \to \mathbb{R}^m$ are used to influence the internal behavior of the system from the outside and the outputs $y\colon \mathbb{R}_{\geq 0} \to \mathbb{R}^p$ model observations of the quantities of interest via an algebraic output equation using the function $g\colon \mathbb{R}_{\geq 0} \times \mathbb{R}^n \to \mathbb{R}^p$. Note that the spaces, in which these functions exist, strongly depend on the final definitions of u, f and g, and, therefore, are omitted here.

The complexity of a dynamical system (1.1) reflects the difficulties that come along with the computation of the solution $\mathsf{x}(t)$. This can amount to different meanings, for example, systems that are described by a linear state-evolution function f are less complex than systems with a nonlinear f. However, an important measure for complexity is the number of differential equations n_1 used to describe the system. With a constantly increasing demand for modeling accuracy also the number of differential equations in dynamical systems grows fast, which makes the systems harder to evaluate in numerical computations such as simulations, optimization procedures or the design of controllers. Even with continuously increasing computational capabilities of modern computers, the demand of large-scale dynamical systems ($n_1 \gtrapprox 10^6$) for computational resources, such as time and memory, easily becomes unmanageable for a growing number of differential equations. Observing that in practice the numbers of inputs and outputs in (1.1) are often very small compared to the number of differential equations, $m, p \ll n_1$, motivates the assumption that not the full solution $\mathsf{x}(t)$ of the differential system is needed to describe the system's input-to-output behavior. The process of model order reduction is the construction of a surrogate system for (1.1) that is described by a much smaller number of differential equations $r_1 \ll n_1$. This makes the surrogate model a lot easier to evaluate than the original system in computations. To actually use the reduced-order model as a surrogate, it needs to approximate the input-to-output behavior of the original system, i.e., for the same input given to the full- and reduced-order models, the output signals are close to each other:

$$\|y - \hat{y}\| \leq \epsilon \cdot \|u\|, \tag{1.2}$$

with the output of the reduced-order model $\hat{y}$, in some appropriate norm, for a suitable tolerance ϵ and all admissible input signals u.

Depending on the underlying physical phenomena, dynamical systems (1.1) can inherit certain structures in the differential equations. The main concern in this thesis are mechanical systems. These usually result from the modeling process of mechanical structures such as bridges, buildings, or vehicles, and describe the time evolution process by differential equations involving second-order time derivatives. For example, linear time-invariant mechanical systems are given by

$$\begin{aligned} M\ddot{x}(t) + E\dot{x}(t) + Kx(t) &= B_{\mathrm{u}}u(t), \\ y(t) &= C_{\mathrm{p}}x(t) + C_{\mathrm{v}}\dot{x}(t), \end{aligned} \tag{1.3}$$

with the system matrices $M, E, K \in \mathbb{R}^{n_2 \times n_2}$, the input matrix $B_{\mathrm{u}} \in \mathbb{R}^{n_2 \times m}$ and the two output matrices $C_{\mathrm{p}}, C_{\mathrm{v}} \in \mathbb{R}^{p \times n_2}$. In principle, it would be possible to rewrite (1.3) into the more classical form (1.1) using substitution variables such that only differential equations with first-order time derivatives are used. However, this replacement process is often undesired. It doubles the number of differential equations describing the system, which increases the computational workload produced in the evaluation of the dynamical system. Over the last decades, a lot of computational tools for dynamical systems were extended to directly handle (1.3) in its original second-order form. In the context of model order reduction, it is desired that the computed surrogate models provide exactly the same structure as (1.3), since:

- this allows the use of the same computational tools as for the original systems,
- structure-preserving reduced-order models often yield a higher accuracy than unstructured variants with the same number of differential equations, and
- the system quantities of the structure-preserving reduced-order model could yield a physical reinterpretation, which gives further computational advantages or new insights into the modeling process.

The preservation of the system structure in the model reduction process is referred to as *structure-preserving model order reduction*. Besides the linear system case (1.3), also other classes of mechanical systems involving special nonlinearities will be treated in this thesis. However, these will be further explored in the corresponding chapters.

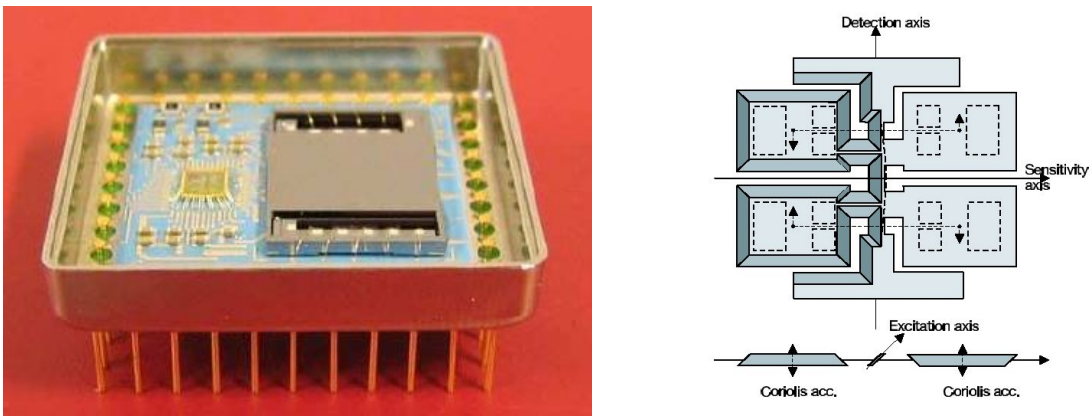
1.2 State of the art

The problem of structure-preserving model order reduction for linear mechanical systems is basically as old as the topic of linear model reduction itself. This amounts to the relevance of mechanical systems in practical applications. Modal truncation, as one of the oldest model reduction methods [75], got quickly extended to the second-order setting in various ways [73, 105, 125]. Even nowadays, structure-preserving modal truncation is the preferred approach for model reduction in engineering sciences due to its generality and computational simplicity [60]. However, a general problem of related approaches is the selection of appropriate system modes to approximate the original dynamics. The dominant pole algorithms [138] were developed as remedy to this problem, which in recent years were extended to large-scale sparse systems [161, 162] as well as to the general case of second-order systems like (1.3); see, for example, in [48, 163]. In practice, the modeling of internal damping of mechanical systems is often simplified to the use of combinations of the mass and stiffness terms of the system leading to so-called modally damped mechanical systems. This subclass of linear mechanical systems holds several advantageous properties that are currently not considered in theory or implementations

of structure-preserving dominant pole algorithms. This point will be discussed in this thesis, while also treating other problems of modal truncation concerning bounds for the approximation error and the limited approximation quality. Further details on modal truncation methods for linear first- and second-order systems can be found in Section 3.2.

A different question arising in model reduction is regions of approximation. Not always the complete frequency axis or infinitely long time simulations are needed in practice. Consequently, it is enough for surrogate models to only approximate frequency or time ranges of interest. For first-order linear systems, this led to the development of the frequency- and time-limited balanced truncation methods [47, 90, 130]. These methods were recently re-considered for structure-preserving model reduction for second-order systems (1.3) in [107, 108]. The authors selected only two ideas from the zoo of second-order balanced truncation methods [69, 143, 159] to transfer the ideas of limited model order reduction. Besides that, there is a general misconception regarding the problem of stability preservation when using second-order balanced truncation methods, and also the problem of applicability of the methods to the large-scale sparse system case. A more general transition from second-order balanced truncation to limited model order reduction is done in this thesis, discussing the problem of stability preservation and proposing numerical methods for the application in the large-scale sparse system setting. An introduction to (limited) balanced truncation and further details are shown in Section 3.4.

Another current research topic in model reduction is the approximation of nonlinear systems. In case of general nonlinearities, time simulations are usually used to gain information about the underlying system dynamics. This is, for example, the case in proper orthogonal decomposition (POD) or in the empirical Gramian framework; see, e.g., [71, 112, 114, 128, 133, 165, 186]. Besides strongly depending on the chosen control signals and time discretization schemes, also the nonlinearities need to be approximated in this setting. This usually amounts to some type of hyper-reduction method like the (discrete) empirical interpolation method ((D)EIM) [20, 71, 77]. Against this background, the focus of research changed in the last years to systems with specially structured nonlinearities, like bilinear and quadratic-bilinear systems [63, 95, 145, 146]. For these systems, intrusive model reduction methods were constructed that do not involve time simulations or the additional use of hyper-reduction methods. For overviews about developed model reduction methods for bilinear and quadratic-bilinear systems see the introductions of Chapters 5 and 6. However, all those newly developed approaches only cope with the case of first-order systems without any further internal structures of the differential equations. In other words, systems with internal structures such as the second-order time-derivatives from the mechanical system case or, for example, systems with internal time delays, cannot be handled by those methods. An important point in this thesis will be to close this gap and to develop model order reduction methods for systems with general internal structures involving bilinear and quadratic nonlinearities.



(a) Manufactured specimen.

(b) Schematic layout.

Figure 1.1: Design of the butterfly gyroscope [61, 149].

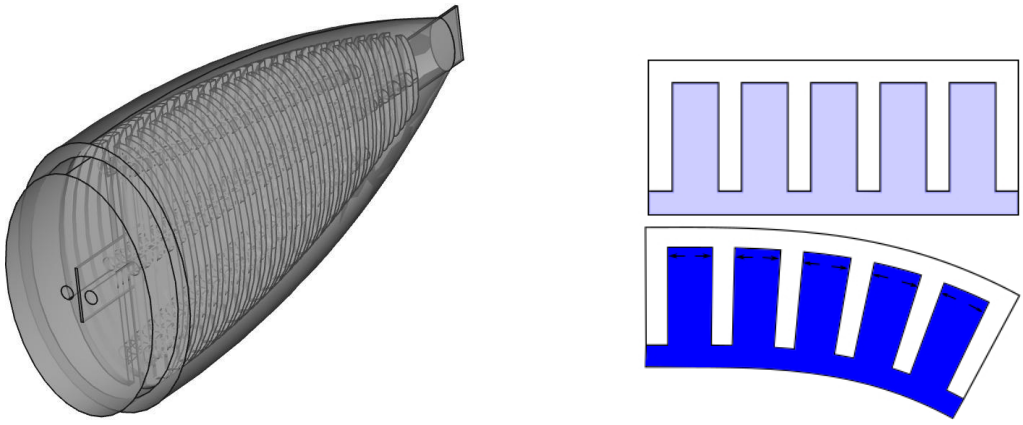
1.3 Motivating examples for mechanical systems

In this section, three motivating examples with underlying mechanical systems are used to illustrate the necessity of structure-preserving model order reduction in practical applications.

1.3.1 Butterfly gyroscope

The butterfly gyroscope is an open benchmark example for model order reduction methods from the Oberwolfach Benchmark Collection [61, 149]. It models a vibrating micro-mechanical gyroscope for the use in inertial navigation applications. The design of the chip itself is illustrated in Figure 1.1. The displacement field is described by linear three-dimensional partial differential equations from elastodynamics involving second-order time derivatives. Using a spatial finite element discretization yields a linear mechanical system of the form (1.3) described by $n_2 = 17\,361$ ordinary differential equations. The states are excited by a single input ($m = 1$) and measuring the displacement of the four wings in the three spatial directions gives $p = 12$ outputs. The internal damping behavior of the gyroscope is modeled by Rayleigh (or proportional) damping $E = \alpha M + \beta K$, with the coefficients $\alpha = 0$ and $\beta = 10^{-6}$.

In the practical process of improving the butterfly gyroscope, the mechanical system needs to be simulated a lot of times with different input signals to analyze the system's behavior with respect to important physical phenomena, for example, its sensitivity to shocks and vibrations. To perform the design process in a reasonable amount of time, it is essential to improve the simulation efficiency of the system. A remedy is the reduction of the number of describing/defining ordinary differential equations by model order reduction techniques. Thereby, the second-order system structure needs to be kept



(a) Transparent sketch. (b) Fluid chambers in relaxed and pressurized state.

Figure 1.2: Design and actuation principle of the artificial fishtail [168].

for the analysis process, and it is even more beneficial if additional mechanical properties like the symmetry and definiteness of the system matrices are preserved. Therefore, structure-preserving model reduction methods are required, here.

1.3.2 Artificial fishtail model

Autonomous underwater vehicles are an important and essential tool in environmental observation tasks [119]. The classical thruster-driven approach has been proven to be mostly inefficient and expensive [83], especially compared with the agile, fast and efficient locomotion that fish naturally developed by evolution [168]. For the construction of fish-like underwater vehicles, the artificial fishtail model was developed [168, 174, 175]. Three-dimensional partial differential equations are used to describe the deformation of a fishtail-shaped silicon structure; see Figure 1.2a. For the fish-like locomotion, the fluid elastomer actuation principle is used [137]. Therefore, the fishtail consists of two symmetric, ripped chambers, as shown in Figure 1.2a, which are alternately put under pressure; see Figure 1.2b. This bends the fishtail alternately into the corresponding directions leading to the typical "flapping" behavior that fish use for locomotion.

The fishtail has a complicated geometric structure, which is expressed in the discretization of the describing partial differential equations. Using the finite element method, the discretized equations are given by a linear mechanical system (1.3) with $n_2 = 779\,232$ ordinary differential equations. A single input ($m = 1$) is used to describe the pressure flow between the inner chambers and the displacement of the fishtail's tip is observed in all three spatial directions ($p = 3$). The internal damping behavior is modeled via the Rayleigh approach with $E = \alpha M + \beta K$, where $\alpha = 10^{-4}$ and $\beta = 2 \cdot 10^{-4}$. The size of the resulting system leads to a tremendous amount of computational resources needed

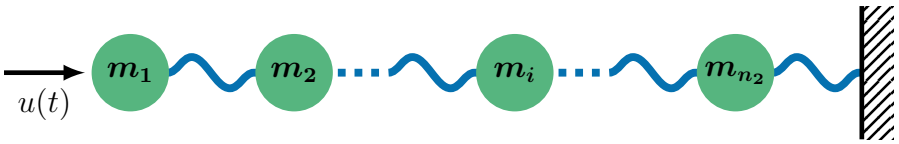


Figure 1.3: Schematic idea of the Toda lattice model with n_2 particles.

to perform simulations, e.g., the simulation of 2 s of the fishtail's behavior easily takes around 45 min of real-world computation time on the hardware described in Section 2.4.1. The full-order system is simply unbearable when it comes to real-time applications or the use of not so powerful hardware for computations, like an onboard chip. Therefore, structure-preserving model reduction is needed here to provide a suitable surrogate model described by only a few differential equations.

1.3.3 Toda lattice model

The Toda lattice [180] is a model that is used in solid-state physics to describe the motion of particles in a one-dimensional crystal structure; see Figure 1.3; by modeling the system as a single chain oscillator with "exponential springs" [70]. The dynamical system is classically given by considering the particle masses with nearest-neighbor interaction and the nonlinear Hamiltonian

$$\mathtt{H}(x;q) = \sum_{j=1}^{n_2} \frac{q_j^2}{2m_j} + \sum_{j=1}^{n_2-1} \frac{e^{k_j(x_j - x_{j+1})}}{k_j} + \frac{e^{k_{n_2} x_{n_2}}}{k_{n_2}} - x_1 - \sum_{j=1}^{n_2} \frac{1}{k_j},$$

where $x_j(t)$ is the displacement of the j-th particle from its initial position in the lattice, $q_j(t)$ the corresponding momentum, and n_2 the overall number of particles. Mass and stiffness coefficients m_j and k_j can be used as parametrization of different particle types and their interactions. To get the system description in terms of ordinary differential equations, the Hamiltonian needs to be differentiated with respect to displacement and momentum, which yields

$$\frac{\partial \mathtt{H}}{\partial q_j}(x;q) = \frac{q_j}{m_j},$$

for all $j = 1, \dots, n_2$, in case of the momenta, and

$$\begin{aligned}
\frac{\partial \mathtt{H}}{\partial x_1}(x;q) &= e^{k_1(x_1-x_2)} - 1,\\
\frac{\partial \mathtt{H}}{\partial x_2}(x;q) &= e^{k_2(x_2-x_3)} - e^{k_1(x_1-x_2)},\\
&\vdots\\
\frac{\partial \mathtt{H}}{\partial x_i}(x;q) &= e^{k_i(x_i-x_{i+1})} - e^{k_{i-1}(x_{i-1}-x_i)},\\
&\vdots\\
\frac{\partial \mathtt{H}}{\partial x_{n_2}}(x;q) &= e^{k_{n_2}x_{n_2}} - e^{k_{n_2-1}(x_{n_2-1}-x_{n_2})},
\end{aligned}$$

for the displacements. The equations of motion

$$\dot{x}(t) = \frac{\partial \mathtt{H}}{\partial q}(x;q), \quad \dot{q}(t) = -\frac{\partial \mathtt{H}}{\partial x}(x;q),$$

together with some additional internal damping, with coefficients $\gamma_k > 0$, results in a nonlinear mechanical system of the form

$$M\ddot{x}(t) + E\dot{x}(t) + f(x(t)) = g(t), \tag{1.4}$$

with initial conditions $x(0) = \dot{x}(0) = 0$ and external forcing $g(t)$, which models the excitation of the particles. The system matrices are then given by

$$M = \begin{bmatrix} m_1 & & \\ & \ddots & \\ & & m_{n_2} \end{bmatrix} \quad \text{and} \quad E = \begin{bmatrix} \gamma_1 & & \\ & \ddots & \\ & & \gamma_{n_2} \end{bmatrix},$$

and the nonlinear function in (1.4), which models the nonlinear springs, is

$$f(x(t)) = \begin{bmatrix} e^{k_1(x_1(t)-x_2(t))} - 1 \\ e^{k_2(x_2(t)-x_3(t))} - e^{k_1(x_1(t)-x_2(t))} \\ \vdots \\ e^{k_i(x_i(t)-x_{i+1}(t))} - e^{k_{i-1}(x_{i-1}(t)-x_i(t))} \\ \vdots \\ e^{k_{n_2}x_{n_2}(t)} - e^{k_{n_2-1}(x_{n_2-1}(t)-x_{n_2}(t))} \end{bmatrix}.$$

Usually, only a small amount of the particles in the model is of actual interest, which adds an algebraic output equation to (1.4), for example,

$$y(t) = C_{\mathrm{v}} x(t), \tag{1.5}$$

with $C_{\mathrm{v}} \in \mathbb{R}^{p \times n_2}$, to observe p linear combinations of the velocities of the particles of interest.

In practical applications with large crystal structures, the number of involved particles quickly increases, which makes the nonlinear system (1.4) arbitrarily large and, consequently, complicated to evaluate. When approximating the system (1.4) by a surrogate model, the approximation should preserve the mechanical system structure, i.e., the second-order time derivatives. Besides reinterpretation of the approximation, in presence of the nonlinearities in (1.4), it might turn out to be beneficial to preserve as many physical properties of the system as possible to provide, at the end, a suitable surrogate.

1.4 Outline of the thesis

This thesis is structured as follows. In Chapter 2, the basic mathematical theory and notations are introduced. It starts with concepts from linear and multilinear/tensor algebra, followed by notional conventions from functional analysis. Thereafter, a compact overview about linear systems theory is given with focus on first-order systems and extensions to the second-order case. For systems with bilinear and quadratic nonlinearities, different frequency domain representations are introduced before the chapter concludes with the setup for numerical experiments. This includes an introduction of the MORscore for the comparison of model reduction methods used in the numerical experiments of this thesis.

Chapter 3 introduces basic ideas of state-of-the-art model order reduction methods for linear systems that are needed later in this thesis. The chapter starts with the projection framework as the main construction approach for reduced-order models in first- and second-order form, here. Thereafter, three different types of model reduction methods are introduced. The first approach is modal truncation, where beside basic ideas for first- and second-order systems also the dominant pole algorithm is discussed. It follows an introduction of interpolation-based (moment matching) model reduction, including a short historical overview, the idea of tangential interpolation for model reduction and extensions to, not only, the case of second-order systems, but also linear systems with a more general internal structure. The last discussed type of model reduction methods is based on the balanced truncation approach. There, frequency- and time-limited variants for first-order systems are outlined, and a collection of formulas for structure-preserving second-order extensions of the classical (unlimited) balanced truncation method is shown.

In Chapter 4, new model reduction methods for linear second-order systems are discussed. Section 4.1 contains a structure-preserving extension of the dominant pole algorithm for modally damped second-order systems. A structured pole-residue formulation is developed and used to define dominant pole pairs of modally damped second-order systems. These ideas are then used to derive a structure-preserving dominant pole algorithm for which error bounds in the $\mathcal{H}_\infty$-norm are derived. A structure-preserving strategy to

overcome weaknesses in the approximation quality is proposed using structured interpolation. The new dominant pole algorithms are then tested using two benchmark examples and compared to other established structure-preserving model order reduction methods. On the other hand, Section 4.2 is concerned with the question of structure-preserving model reduction for second-order systems with localized approximation behavior in frequency and time domain. A structure-preserving extension to second-order systems for the frequency- and time-limited balanced truncation methods is proposed. To overcome problems with the preservation of stability in the reduced-order model, alternative approaches are discussed. To handle the arising large-scale sparse matrix equations, numerical procedures such as large-scale matrix equation solvers, an α-shift strategy and hybrid model order reduction methods are outlined. For two benchmark examples, the different resulting limited structure-preserving model reduction methods are computed. The results are compared to each other and to the classical approaches with global (unlimited) approximation behavior.

Inspired by bilinear mechanical systems, in Chapter 5, model order reduction for bilinear systems with a more general concept of internal structure is discussed. First, the frequency representation of bilinear systems, namely the subsystem transfer functions, is extended to the general structured setting using two different example structures as motivation. A new structure-preserving interpolation framework for these structured transfer functions of bilinear systems is then introduced. This includes results on matching interpolation conditions in explicit as well as implicit ways. For the case of single-input/single-output systems, numerical experiments are used to compare structured reduced-order models to unstructured ones. The interpolation theory is then extended to the case of structured parametric bilinear systems. Last, the idea of tangential interpolation is used to tackle structured bilinear multiple-input/multiple-output systems. Via different motivations, a unifying framework is developed that covers various ideas of tangential interpolation for bilinear systems at the same time. In numerical experiments, the different tangential interpolation methods are compared to each other, as well as to the alternative approach of matrix interpolation.

Chapter 6 is motivated by nonlinear mechanical systems but considers more general structures similar to the bilinear system case. The process of quadratic-bilinearization is used to derive structured quadratic-bilinear systems. Frequency representations in terms of subsystem transfer functions of quadratic-bilinear systems are then extended to the structured setting, and afterwards, a structure-preserving model reduction approach is proposed based on the interpolation of structured transfer functions. The Toda lattice model is used as a nonlinear mechanical system example to test the developed theory in numerical experiments.

This thesis is concluded in Chapter 7 with a summary of the results and an overview of open questions and research perspectives.

CHAPTER 2

MATHEMATICAL BASICS AND GENERAL SETTING

Contents

In this chapter, the mathematical preliminaries are summarized and the notation of this thesis is fixed. First, some basic terms and notation from tensor algebra and functional analysis are introduced in Section 2.1. Afterwards in Section 2.2, basic system-theoretic notion and concepts are considered for linear systems in first- and second-order form. Frequency representations of systems with special nonlinearities are discussed in Section 2.3. The chapter is concluded in Section 2.4 by the hardware and software setup used in all numerical experiments of this thesis, and by an introduction of the MORscore used for the comparison of model reduction methods.

2.1 Basic linear algebra concepts and notation

2.1.1 Tensor algebra

Before discussing tensors and some algebraic results for these, the following definition gives two important operations for matrices. Similar introductions to tensor algebra can be found in [63, 95].

Definition 2.1 (Vectorization and Kronecker product [91]):
Let $X = \begin{bmatrix} x_1 & \dots & x_{n_2} \end{bmatrix} \in \mathbb{C}^{n_1 \times n_2}$ be an arbitrary matrix with columns $x_j \in \mathbb{C}^{n_1}$, for $j = 1, \dots, n_2$. The *vectorization* of X is defined as the row concatenation of the columns of X:

$$\operatorname{vec}(X) := \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n_2} \end{bmatrix} \in \mathbb{C}^{n_1 n_2}.$$

Given another matrix $Y \in \mathbb{C}^{n_3 \times n_4}$, the *Kronecker product* of X with Y is defined to be

$$X \otimes Y = \begin{bmatrix} x_{11}Y & \cdots & x_{1n_2}Y \\ \vdots & & \vdots \\ x_{n_1 1}Y & \cdots & x_{n_1 n_2}Y \end{bmatrix} \in \mathbb{C}^{n_1 n_3 \times n_2 n_4}. \qquad \lozenge$$

Results and properties following from Definition 2.1 can be found in standard linear algebra textbooks, e.g., in [91, 115]. Additionally, the Hermitian transposed of a matrix $X \in \mathbb{C}^{n_1 \times n_2}$ will be denoted by $X^{\mathsf{H}} := \overline{X}^{\mathsf{T}} \in \mathbb{C}^{n_2 \times n_1}$.

In the last decades, tensors received more and more attention by different mathematical and engineering communities [98, 123, 127], especially in the application of low-rank approximations [97]. Formally, a tensor $\boldsymbol{X}$ is a multi-linear description that relates algebraic objects corresponding to vector spaces. It is usually interpreted as a number array of order d with its elements indexed by a product index set

$$\mathcal{I} = \mathcal{I}_1 \times \ldots \times \mathcal{I}_d,$$

with $|\mathcal{I}_j| = n_j$ and often assumed to be $\mathcal{I}_j = \{1, 2, \dots, n_j\}$, for $j = 1, \dots, d$. For example, $\boldsymbol{X} \in \mathbb{C}^{n_1 \times \ldots \times n_d}$ is a d-th-order tensor with entries from $\mathbb{C}$ and dimensions $n_1, \dots, n_d$. While tensors are often a good way to represent certain types of data, they are problematic when computations need to be performed. These are usually done via matrix representations of the tensors. While there are various ways of *flattening* tensors into matrices [98, 122], only the following definition will be of interest in this thesis.

Definition 2.2 (Tensor μ-mode matricizations [123]):
The *μ-mode matricization* $\boldsymbol{X}^{(\mu)} \in \mathbb{C}^{n_\mu \times n_1 \cdots n_{\mu-1} n_{\mu+1} \cdots n_d}$ of a tensor $\boldsymbol{X} \in \mathbb{C}^{n_1 \times \ldots \times n_d}$, with $1 \leq \mu \leq d$, is defined to be the mapping of tensor indices $(i_1, i_2, \ldots, i_d)$ onto matrix indices (i_μ, j) with

$$j = 1 + \sum_{k=1,\, k\neq\mu}^{d} (i_k - 1) J_k, \quad \text{where } J_k = \prod_{\ell=1,\, \ell\neq k}^{k-1} n_\ell. \qquad \lozenge$$

As illustration of Definition 2.2, consider the third-order tensor $\boldsymbol{X} \in \mathbb{C}^{2\times 2\times 3}$. Then, the matricizations of $\boldsymbol{X}$ read as follows

$$\boldsymbol{X}^{(1)} = \begin{bmatrix} \boldsymbol{X}_{(1,1,1)} & \boldsymbol{X}_{(1,2,1)} & \boldsymbol{X}_{(1,1,2)} & \boldsymbol{X}_{(1,2,2)} & \boldsymbol{X}_{(1,1,3)} & \boldsymbol{X}_{(1,2,3)} \\ \boldsymbol{X}_{(2,1,1)} & \boldsymbol{X}_{(2,2,1)} & \boldsymbol{X}_{(2,1,2)} & \boldsymbol{X}_{(2,2,2)} & \boldsymbol{X}_{(2,1,3)} & \boldsymbol{X}_{(2,2,3)} \end{bmatrix},$$
$$\boldsymbol{X}^{(2)} = \begin{bmatrix} \boldsymbol{X}_{(1,1,1)} & \boldsymbol{X}_{(2,1,1)} & \boldsymbol{X}_{(1,1,2)} & \boldsymbol{X}_{(2,1,2)} & \boldsymbol{X}_{(1,1,3)} & \boldsymbol{X}_{(2,1,3)} \\ \boldsymbol{X}_{(1,2,1)} & \boldsymbol{X}_{(2,2,1)} & \boldsymbol{X}_{(1,2,2)} & \boldsymbol{X}_{(2,2,2)} & \boldsymbol{X}_{(1,2,3)} & \boldsymbol{X}_{(2,2,3)} \end{bmatrix},$$
$$\boldsymbol{X}^{(3)} = \begin{bmatrix} \boldsymbol{X}_{(1,1,1)} & \boldsymbol{X}_{(2,1,1)} & \boldsymbol{X}_{(1,2,1)} & \boldsymbol{X}_{(2,2,1)} \\ \boldsymbol{X}_{(1,1,2)} & \boldsymbol{X}_{(2,1,2)} & \boldsymbol{X}_{(1,2,2)} & \boldsymbol{X}_{(2,2,2)} \\ \boldsymbol{X}_{(1,1,3)} & \boldsymbol{X}_{(2,1,3)} & \boldsymbol{X}_{(1,2,3)} & \boldsymbol{X}_{(2,2,3)} \end{bmatrix}.$$

As one can already see by this example, in case of third-order tensors, all matricizations can easily be converted into each other using matrix operations. Let a tensor $\boldsymbol{X} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$ be given with its 1-mode matricization

$$\boldsymbol{X}^{(1)} = \begin{bmatrix} X_1 & X_2 & \ldots & X_{n_3} \end{bmatrix},$$

where $X_j \in \mathbb{C}^{n_1 \times n_2}$ for all $j = 1, \ldots, n_3$. Then, the other two matricizations can be written as

$$\boldsymbol{X}^{(2)} = \begin{bmatrix} X_1^{\mathsf{T}} & X_2^{\mathsf{T}} & \ldots & X_{n_3}^{\mathsf{T}} \end{bmatrix} \quad \text{and} \quad \boldsymbol{X}^{(3)} = \begin{bmatrix} \operatorname{vec}(X_1) & \operatorname{vec}(X_2) & \ldots & \operatorname{vec}(X_{n_3}) \end{bmatrix}^{\mathsf{T}}.$$

Note that even with the elements of $\boldsymbol{X}$ to be from $\mathbb{C}$, the matricizations are only rearrangements of these and, therefore, involve only the transposed instead of the conjugate transposed operation.

An important point when working with matricizations of tensors is the multiplication with other matrices. Given a third-order tensor $\boldsymbol{X} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$ and three matrices $U \in \mathbb{C}^{n_1 \times m_1}$, $V \in \mathbb{C}^{n_2 \times m_2}$, $W \in \mathbb{C}^{n_3 \times m_3}$, then if the tensor $\boldsymbol{Y} \in \mathbb{C}^{m_1 \times m_2 \times m_3}$ is given by its 1-mode matricization such that

$$\boldsymbol{Y}^{(1)} = U^{\mathsf{H}} \boldsymbol{X}^{(1)} (W \otimes V), \tag{2.1}$$

equivalently $\boldsymbol{Y}$ can be computed by

$$\boldsymbol{Y}^{(2)} = V^{\mathsf{T}} \boldsymbol{X}^{(2)} (W \otimes \overline{U}) = \overline{V^{\mathsf{H}} \overline{\boldsymbol{X}^{(2)}} (\overline{W} \otimes U)}, \tag{2.2}$$
$$\boldsymbol{Y}^{(3)} = W^{\mathsf{T}} \boldsymbol{X}^{(3)} (V \otimes \overline{U}) = \overline{W^{\mathsf{H}} \overline{\boldsymbol{X}^{(3)}} (\overline{V} \otimes U)}; \tag{2.3}$$

see, e.g., [123]. In other words, the product of matrices with a tensor matricization is equivalently described by other matricizations of the resulting tensor. This allows formally to change the order of multiplications.

Another property of third-order tensors that is often used in the context of model order reduction approaches, e.g., in [4, 30, 92], is symmetry.

Definition 2.3 (Symmetric tensors [123]):
A tensor $\boldsymbol{X} \in \mathbb{C}^{n\times n\times n}$ is called symmetric if $\boldsymbol{X}^{(2)} = \boldsymbol{X}^{(3)}$ holds. ◊

For a symmetric tensor $\boldsymbol{X} \in \mathbb{C}^{n\times n\times n}$ and two arbitrary vectors $u, v \in \mathbb{C}^n$, it is easy to see by using (2.1) and (2.2) that

$$\boldsymbol{X}^{(1)}(u \otimes v) = \left(v^{\mathsf{T}}\boldsymbol{X}^{(2)}(u \otimes I_n)\right)^{\mathsf{T}} = \left(v^{\mathsf{T}}\boldsymbol{X}^{(3)}(u \otimes I_n)\right)^{\mathsf{T}} = \boldsymbol{X}^{(1)}(v \otimes u) \tag{2.4}$$

holds. But usually, the occurring tensors are not given in symmetric form. Since in [4, 30, 92], they are used in quadratic systems to be multiplied only with a vector in Kronecker product with itself, $\boldsymbol{X}^{(1)}(v \otimes v)$, it is possible to symmetrize the underlying tensor since this will not change the application of its 1-mode matricization on $(v \otimes v)$. A tensor $\boldsymbol{X} \in \mathbb{C}^{n\times n\times n}$ can be symmetrized by computing a new tensor $\widetilde{\boldsymbol{X}} \in \mathbb{C}^{n\times n\times n}$ such that

$$\widetilde{\boldsymbol{X}}^{(2)} = \frac{1}{2}(\boldsymbol{X}^{(2)} + \boldsymbol{X}^{(3)}).$$

2.1.2 Notion from vector calculus

Due to its heavy use in this thesis, an abbreviation for partial derivatives is introduced

$$\partial_{s_1^{j_1}\dots s_k^{j_k}} f(z_1, \dots, z_k) := \frac{\partial^{j_1+\dots+j_k} f}{\partial s_1^{j_1} \cdots \partial s_k^{j_k}}(z_1, \dots, z_k), \tag{2.5}$$

denoting the differentiation of a function $f\colon \mathbb{C}^k \to \mathbb{C}^\ell$ with respect to the complex variables $s_1, \dots, s_k$ and evaluated at $z_1, \dots, z_k \in \mathbb{C}$. In the sense of (2.5), the *Jacobian* of f will be denoted by

$$\nabla f(z_1, \dots, z_k) = \begin{bmatrix} \partial_{s_1} f(z_1, \dots, z_k) & \dots & \partial_{s_k} f(z_1, \dots, z_k) \end{bmatrix}, \tag{2.6}$$

to be the column concatenation of all partial derivatives.

In terms of the notation of general functions, this thesis will not involve any inverse functions, i.e., for a given $f\colon z \mapsto y$ the function f^{-1} will not denote the inverse mapping $y \mapsto z$ but the inversion of the resulting object of the function f. As example, consider the matrix valued function $\mathcal{K}\colon \mathbb{C} \to \mathbb{C}^{n\times n}$, which maps a complex variable onto an n-dimensional square matrix. Then

$$\mathcal{K}^{-1} := \mathcal{K}(.)^{-1} \tag{2.7}$$

denotes the inverse of the n-dimensional square matrix in the frequency points in which $\mathcal{K}$ is invertible.

These two types of abbreviations (2.5) and (2.7) will also occur combined. For example, given a second matrix-valued function $\mathcal{B}\colon \mathbb{C} \to \mathbb{C}^{n\times m}$, the partial derivative of the product with the inverse of $\mathcal{K}$ will be denoted by

$$\partial_{s_1^{j_1} s_2^{j_2}} (\mathcal{K}^{-1}\mathcal{B})(z_1, z_2) := \frac{\partial^{j_1+j_2} \mathcal{K}(.)^{-1}\mathcal{B}(.)}{\partial s_1^{j_1} \partial s_2^{j_2}}(z_1, z_2).$$

Further on, the usual misuse of notation from systems theory and numerical analysis will be applied in this thesis.

2.2 System-theoretic concepts for linear systems

This section is concerned with basic concepts for linear time-invariant systems. The points presented here are mainly taken from [9] but can also be found in other standard textbooks about systems theory or model order reduction; see, e.g., [10, 34, 113, 154, 177]. This section itself is additionally separated into the classical first-order systems and (mechanical) second-order systems.

2.2.1 First-order systems

Before the special case of mechanical (second-order) systems is considered, some properties of *first-order linear time-invariant (LTI) systems* are needed first. These systems have the form

$$\mathsf{G}_{\mathrm{L}} : \begin{cases} \mathsf{E}\dot{\mathsf{x}}(t) = \mathsf{A}\mathsf{x}(t) + \mathsf{B}u(t), \\ y(t) = \mathsf{C}\mathsf{x}(t), \end{cases} \tag{2.8}$$

with $\mathsf{E}, \mathsf{A} \in \mathbb{R}^{n_1\times n_1}$, $\mathsf{B} \in \mathbb{R}^{n_1\times m}$, $\mathsf{C} \in \mathbb{R}^{p\times n_1}$; E invertible, if not stated otherwise, and initial condition $\mathsf{x}(t_0) = \mathsf{x}_0$ with $\mathsf{x}_0 \in \mathbb{R}^{n_1}$. Default assumptions for (2.8) in model order reduction are $\mathsf{x}(t_0) = 0$ and $t_0 = 0$ to neglect the initial value's influence on the system's behavior. These assumptions are also made through-out this thesis. As in the general case of dynamical systems (1.1), the behavior of (2.8) is given via the three time-dependent functions: $u\colon \mathbb{R}_{\geq 0} \to \mathbb{R}^m$, the inputs that are used to control $\mathsf{x}\colon \mathbb{R}_{\geq 0} \to \mathbb{R}^{n_1}$, the internal states, to get the desired outputs $y\colon \mathbb{R}_{\geq 0} \to \mathbb{R}^p$.

Remark 2.4 (Feed-through terms):
A common modification of (2.8) in systems and control theory is the addition of a feed-through term $\mathsf{D} \in \mathbb{R}^{p\times m}$ to the output equation such that

$$y(t) = \mathsf{C}\mathsf{x}(t) + \mathsf{D}u(t).$$

This feed-through term will not play any role in this thesis, but all developed model reduction theory can be transferred to systems with feed-through term by preserving the original term in the reduced-order system $\widehat{\mathsf{D}} = \mathsf{D}$.

In some applications, the case $\widehat{\mathsf{D}} \neq \mathsf{D}$ is of particular interest. This can be treated in certain model reduction approaches, like interpolation methods, by additional modifications of the construction formulae; see, e.g., [24, 84]. ◊

The first-order system (2.8) can be found under different names in the literature, usually depending on the specific realization of the E matrix. The system (2.8) is called a *standard state-space system* in case of $\mathsf{E} = I_{n_1}$ and it is called a *generalized state-space system* if E is invertible but not the identity, i.e., when the states are described by a system of ODEs with a mass matrix. In case of E singular, (2.8) contains DAEs and is referred to as *descriptor system*. Furthermore, the system (2.8) is called *single-input/single-output (SISO)* in case of $m = p = 1$ and *multiple-input/multiple-output (MIMO)* otherwise. Since E is assumed to be invertible, the state of (2.8) is analytically given via the *variation of constants* principle with

$$\mathsf{x}(t) = e^{\mathsf{E}^{-1}\mathsf{A}t}\mathsf{x}_0 + \int_{t_0}^{t} e^{\mathsf{E}^{-1}\mathsf{A}(t-\tau)}\mathsf{E}^{-1}\mathsf{B}u(\tau)\mathrm{d}\tau. \tag{2.9}$$

Subsequently, the system output of (2.8) can be written as

$$y(t) = \mathsf{C}e^{\mathsf{E}^{-1}\mathsf{A}t}\mathsf{x}_0 + \int_{t_0}^{t} \mathsf{C}e^{\mathsf{E}^{-1}\mathsf{A}(t-\tau)}\mathsf{E}^{-1}\mathsf{B}u(\tau)\mathrm{d}\tau. \tag{2.10}$$

Definition 2.5 (System realizations and order [9, Definition 4.2]):
The quadruple $\mathsf{G}_\mathrm{L} = (\mathsf{A}, \mathsf{B}, \mathsf{C}, \mathsf{E}) \in \mathbb{R}^{n_1\times n_1} \times \mathbb{R}^{n_1\times m} \times \mathbb{R}^{p\times n_1} \times \mathbb{R}^{n_1\times n_1}$ is called a *realization* of the system (2.8). The *order* of (2.8) is defined to be the dimension of the corresponding state-space n_1. ◊

In general, the realization of a system is not unique in the sense of its input-to-output behavior, i.e., the same system can be described by different realizations. A system realization (2.8) is called *equivalent* to another realization $\widetilde{\mathsf{G}}_\mathrm{L} = (\widetilde{\mathsf{A}}, \widetilde{\mathsf{B}}, \widetilde{\mathsf{C}}, \widetilde{\mathsf{E}})$ if and only if there exist (invertible) transformation matrices $Z, T \in \mathbb{C}^{n_1\times n_1}$ such that

$$\widetilde{\mathsf{E}} = Z^\mathsf{H}\mathsf{E}T, \quad \widetilde{\mathsf{A}} = Z^\mathsf{H}\mathsf{A}T, \quad \widetilde{\mathsf{B}} = Z^\mathsf{H}\mathsf{B}, \quad \widetilde{\mathsf{C}} = \mathsf{C}T. \tag{2.11}$$

Therein, the matrix T yields a coordinate transformation $\tilde{\mathsf{x}} = T\mathsf{x}$ and Z transforms the describing equations. The change of one system to an equivalent one in the sense of (2.11) is referred to as *generalized state-space transformation*.

The following definition introduces some important system properties.

Definition 2.6 (Basic system properties [9, Definitions 4.2, 4.6, and 4.19]): The system (2.8) is called:

(a) *asymptotically stable* or *c-stable*, if all eigenvalues of the matrix pencil $\lambda\mathsf{E} - \mathsf{A}$, i.e., all $\lambda \in \mathbb{C}$ such that $\det(\lambda\mathsf{E} - \mathsf{A}) = 0$, have negative real parts.

(b) *controllable* in $[t_0, t_\mathrm{f}]$, if any initial state $\mathsf{x}(t_0)$ can be steered to any final state $\mathsf{x}(t_\mathrm{f})$ by an appropriate input signal $u(t)$ with finite energy.

(c) *observable* in $[t_0, t_\mathrm{f}]$, if the set of states such that $y(t) = \mathsf{C}\mathsf{x}(t) = 0$, for all $t \in [t_0, t_\mathrm{f}]$, contains only the zero state $\mathsf{x}(t) = 0$. ◊

Controllability and observability are important concepts in model order reduction to characterize system components that do not contribute substantially to the input-to-output behavior of the system. It can be shown that a system (2.8) is *minimal*, i.e., has the smallest possible order to describe exactly the input-to-output behavior, if and only if it is controllable and observable. There are a variety of different equivalent definitions and criteria for the system properties in Definition 2.6. Some can be found, for example, in [9, Chapters 4 and 5].

A useful tool to deal with systems of differential equations is the Laplace transformation. For a time domain function $f\colon \mathbb{R}_{\geq 0} \to \mathbb{R}^n$, its *Laplace transform* is defined to be

$$F(s) = \mathcal{L}\left\{f(t)\right\}(s) := \int_0^\infty f(t)e^{-st}\mathrm{d}t, \tag{2.12}$$

if the integral exists, with the complex frequency variable $s \in \mathbb{C}$. Applying now (2.12) to the linear system (2.8) results in an equivalent description in the complex frequency domain via algebraic equations rather than differential ones

$$\begin{aligned} s\mathsf{E}\mathsf{X}(s) - \mathsf{E}\mathsf{x}_0 &= \mathsf{A}\mathsf{X}(s) + \mathsf{B}U(s), \\ Y(s) &= \mathsf{C}\mathsf{X}(s), \end{aligned} \tag{2.13}$$

where $\mathsf{X}\colon \mathbb{C} \to \mathbb{C}^n$, $U\colon \mathbb{C} \to \mathbb{C}^m$, and $Y\colon \mathbb{C} \to \mathbb{C}^p$ are the Laplace transforms of the equally named time domain functions x, u and y, respectively. With the assumption that $\mathsf{x}_0 = 0$, the input-to-output behavior of (2.8) in the frequency domain can be directly described by

$$\begin{aligned} Y(s) &= \left(\mathsf{C}(s\mathsf{E} - \mathsf{A})^{-1}\mathsf{B}\right)U(s) \\ &=: \mathsf{G}_\mathrm{L}(s)U(s), \end{aligned}$$

where the complex, matrix-valued function

$$\mathsf{G}_\mathrm{L}(s) = \mathsf{C}(s\mathsf{E} - \mathsf{A})^{-1}\mathsf{B} \tag{2.14}$$

is called the *transfer function* of (2.8).

In model order reduction, the input-to-output behavior of (2.8) is approximated via a surrogate model of smaller order. For an analysis of the approximation quality, norms for dynamical systems are needed. The following definition states two commonly used system norms.

Definition 2.7 (System norms [9, Section 5.1.3]):
Assume (2.8) to be asymptotically stable with its transfer function (2.14).

(a) The $\mathcal{H}_2$-norm is defined as

$$\|\mathsf{G}_\mathrm{L}\|_{\mathcal{H}_2} := \sqrt{\frac{1}{2\pi}\int_{-\infty}^{\infty} \|\mathsf{G}_\mathrm{L}(\omega \mathrm{i})\|_F^2 \mathrm{d}\omega}.$$

(b) The $\mathcal{H}_\infty$-norm is defined as

$$\|\mathsf{G}_\mathrm{L}\|_{\mathcal{H}_\infty} := \sup_{\omega\in\mathbb{R}} \|\mathsf{G}_\mathrm{L}(\omega \mathrm{i})\|_2. \qquad \lozenge$$

While most of the time, the norms in Definition 2.7 are sufficient for studying stable systems, it should be noted that an important expansion of the $\mathcal{H}_\infty$-norm for systems with anti-stable parts, i.e., where eigenvalues of $\lambda\mathsf{E} - \mathsf{A}$ have positive real parts, is the $\mathcal{L}_\infty$-norm. This norm is analogously to the $\mathcal{H}_\infty$-norm defined as

$$\|\mathsf{G}_\mathrm{L}\|_{\mathcal{L}_\infty} := \sup_{\omega\in\mathbb{R}} \|\mathsf{G}_\mathrm{L}(\omega \mathrm{i})\|_2.$$

The norms in Definition 2.7 are defined using the system's transfer function in the frequency domain. Results from Parseval, Plancherel and Payley-Wiener give links between the time and frequency domain description of (2.8) in terms of norms and spaces [9, Proposition 5.1]. Roughly speaking, the approximation behavior of the transfer functions in the frequency domain is equivalent to the input-to-output approximation behavior in the time domain, i.e., the better the transfer function is approximated the smaller the time domain input-to-output error will be. In fact, the following two inequalities can be shown to hold in time domain (and also in frequency domain with accordingly changed functions and spaces):

$$\|y - \hat{y}\|_{L_2} \leq \|\mathsf{G}_\mathrm{L} - \widehat{\mathsf{G}}_\mathrm{L}\|_{\mathcal{H}_\infty} \|u\|_{L_2},$$
$$\|y - \hat{y}\|_{L_\infty} \leq \|\mathsf{G}_\mathrm{L} - \widehat{\mathsf{G}}_\mathrm{L}\|_{\mathcal{H}_2} \|u\|_{L_2},$$

for u and $y - \hat{y}$ in the appropriate spaces, and where $\hat{y}$ is the output signal of an approximating system corresponding to $\widehat{\mathsf{G}}_\mathrm{L}$. The two norms used above are the time

domain L_2- and L_∞-norms, which are defined by

$$\|x\|_{L_2} := \sqrt{\int_{t_0}^{t_\mathrm{f}} \|x(t)\|_2^2 \mathrm{d}t}, \tag{2.15}$$

$$\|x\|_{L_\infty} := \sup_{t\in[t_0,t_\mathrm{f}]} \|x(t)\|_\infty, \tag{2.16}$$

for a time domain function $x\colon \mathbb{R}_{\geq 0} \to \mathbb{R}^n$.

2.2.2 Second-order systems

The main interest of this thesis lies in mechanical systems. In the LTI case, these systems are usually described by differential equations with second-order time derivatives of the form

$$G_\mathrm{L} : \begin{cases} M\ddot{x}(t) + E\dot{x}(t) + Kx(t) = B_\mathrm{u}u(t), \\ y(t) = C_\mathrm{p}x(t) + C_\mathrm{v}\dot{x}(t), \end{cases} \tag{2.17}$$

with $M, E, K \in R^{n_2\times n_2}$, $B_\mathrm{u} \in \mathbb{R}^{n_2\times m}$, $C_\mathrm{p}, C_\mathrm{v} \in \mathbb{R}^{p\times n_2}$; M invertible, if not stated otherwise, and the initial conditions $x(t_0) = x_{\mathrm{p},0}$, $\dot{x}(t_0) = x_{\mathrm{v},0}$, with $x_{\mathrm{p},0}, x_{\mathrm{v},0} \in \mathbb{R}^{n_2}$. Systems of the form (2.17) are further on referred to as *second-order LTI systems*. The system matrices M, E, K are thereby known as *mass, damping and stiffness matrices*. Conform with the previous section, the default assumptions for systems like (2.17) will be zero initial conditions $x_{\mathrm{p},0} = x_{\mathrm{v},0} = 0$, with $t_0 = 0$. The Definition 2.5 is extended appropriately for (2.17). The order of (2.17) is the corresponding state-space dimension n_2, and the tuple

$$G_\mathrm{L} = (M, E, K, B_\mathrm{u}, C_\mathrm{p}, C_\mathrm{v})$$

is a realization of (2.17). In case of mechanical systems, M and K are usually symmetric positive definite, and $E + E^\mathsf{T}$ symmetric positive semi-definite. Often also E itself is symmetric positive semi-definite.

In principle, the theory of linear first-order systems (2.8) can be directly transferred to the second-order case by reformulating (2.17) as a first-order system. There exist infinitely many first-order realizations of (2.17). The most commonly used ones are summarized in the following; see, e.g., [151, 159].

The *first companion form realization* can be obtained by introducing the first-order state vector $\mathsf{x}^\mathsf{T} = \begin{bmatrix} x^\mathsf{T} & \dot{x}^\mathsf{T} \end{bmatrix}$. Reordering the lower-order dynamics to the right-hand side yields an equivalent description of (2.17) by a first-order system of the form (2.8), with the system matrices

$$\mathsf{E}_\mathrm{fc} = \begin{bmatrix} J_\mathrm{fc} & 0 \\ 0 & M \end{bmatrix}, \quad \mathsf{A}_\mathrm{fc} = \begin{bmatrix} 0 & J_\mathrm{fc} \\ -K & -E \end{bmatrix}, \quad \mathsf{B}_\mathrm{fc} = \begin{bmatrix} 0 \\ B_\mathrm{u} \end{bmatrix}, \quad \mathsf{C}_\mathrm{fc} = \begin{bmatrix} C_\mathrm{p} & C_\mathrm{v} \end{bmatrix}, \tag{2.18}$$

where $J_{\mathrm{fc}} \in \mathbb{R}^{n_2 \times n_2}$ is an arbitrary invertible matrix. The input-to-output behavior of (2.17) and the first-order system (2.8) with the matrices (2.18) is identical. A classical choice for the invertible matrix is $J_{\mathrm{fc}} = I_{n_2}$. In case of M, E, K symmetric and K invertible, another suitable choice for the invertible matrix is $J_{\mathrm{fc}} = -K$, since thereby E_{fc} and A_{fc} become symmetric. If additionally $B_{\mathrm{u}} = C_{\mathrm{v}}^{\mathsf{T}}$ and $C_{\mathrm{p}} = 0$ hold, the first companion form realization is also state-space symmetric.

A different realization is obtained by moving only the state without time derivative to the right-hand side. The *second companion form realization* of (2.17) is then given by

$$\mathsf{E}_{\mathrm{sc}} = \begin{bmatrix} E & M \\ J_{\mathrm{sc}} & 0 \end{bmatrix}, \quad \mathsf{A}_{\mathrm{sc}} = \begin{bmatrix} -K & 0 \\ 0 & J_{\mathrm{sc}} \end{bmatrix}, \quad \mathsf{B}_{\mathrm{sc}} = \begin{bmatrix} B_{\mathrm{u}} \\ 0 \end{bmatrix}, \quad \mathsf{C}_{\mathrm{sc}} = \begin{bmatrix} C_{\mathrm{p}} & C_{\mathrm{v}} \end{bmatrix}, \tag{2.19}$$

with $J_{\mathrm{sc}} \in \mathbb{R}^{n_2 \times n_2}$ an arbitrary invertible matrix. The default choice for J_{sc} in (2.19), if M is invertible, is $J_{\mathrm{sc}} = M$. Then, in case of M, E, K symmetric, the first-order system matrices E_{sc} and A_{sc} become symmetric, too. Also, the second companion form realization becomes state-space symmetric if additionally $B_{\mathrm{u}} = C_{\mathrm{p}}^{\mathsf{T}}$ and $C_{\mathrm{v}} = 0$ hold.

Since (2.18) and (2.19) are both realizations of the same second-order system, i.e., they are equivalent, the question of the corresponding transformation matrices in (2.11) arises to switch between the two realizations. One can easily prove that (2.18) can be transformed into (2.19) using the transformation matrices

$$Z_{\mathrm{fc2sc}} = \begin{bmatrix} J_{\mathrm{fc}}^{-\mathsf{T}} E^{\mathsf{T}} & J_{\mathrm{fc}}^{-\mathsf{T}} J_{\mathrm{sc}}^{\mathsf{T}} \\ I_{n_2} & 0 \end{bmatrix} \quad \text{and} \quad T_{\mathrm{fc2sc}} = \begin{bmatrix} I_{n_2} & 0 \\ 0 & I_{n_2} \end{bmatrix} = I_{2n_2}, \tag{2.20}$$

i.e., it holds

$$\mathsf{E}_{\mathrm{sc}} = Z_{\mathrm{fc2sc}}^{\mathsf{T}} \mathsf{E}_{\mathrm{fc}} T_{\mathrm{fc2sc}}, \quad \mathsf{A}_{\mathrm{sc}} = Z_{\mathrm{fc2sc}}^{\mathsf{T}} \mathsf{A}_{\mathrm{fc}} T_{\mathrm{fc2sc}}, \quad \mathsf{B}_{\mathrm{sc}} = Z_{\mathrm{fc2sc}}^{\mathsf{T}} \mathsf{B}_{\mathrm{fc}}, \quad \mathsf{C}_{\mathrm{sc}} = \mathsf{C}_{\mathrm{fc}} T_{\mathrm{fc2sc}}.$$

Note that the reverse transformation from second to first companion form is given by the inverse transformation matrices

$$Z_{\mathrm{fc2sc}}^{-1} = \begin{bmatrix} 0 & I_{n_2} \\ J_{\mathrm{sc}}^{-\mathsf{T}} J_{\mathrm{fc}}^{\mathsf{T}} & -J_{\mathrm{sc}}^{-\mathsf{T}} E^{\mathsf{T}} \end{bmatrix} \quad \text{and} \quad T_{\mathrm{fc2sc}}^{-1} = I_{2n_2}. \tag{2.21}$$

In practice, while both companion forms have different advantages, they can quickly run into numerical problems during computations due to the indefiniteness of the first-order system matrices. Therefore, a third first-order realization is mentioned here for later use. Assuming K to be invertible, the *strictly dissipative realization* of (2.17), as introduced in [151], is given by

$$\begin{aligned} \mathsf{E}_{\mathrm{sd}} &= \begin{bmatrix} K & \gamma M \\ \gamma M & M \end{bmatrix}, \quad \mathsf{A}_{\mathrm{sd}} = \begin{bmatrix} -\gamma K & K - \gamma E \\ -K & \gamma M - E \end{bmatrix}, \\ \mathsf{B}_{\mathrm{sd}} &= \begin{bmatrix} \gamma B_{\mathrm{u}} \\ B_{\mathrm{u}} \end{bmatrix}, \qquad \mathsf{C}_{\mathrm{sd}} = \begin{bmatrix} C_{\mathrm{p}} & C_{\mathrm{v}} \end{bmatrix}, \end{aligned} \tag{2.22}$$

with the parameter $0 < \gamma < \lambda_{\min}\left(E(M+\frac{1}{4}EK^{-1}E)^{-1}\right)$. It was shown in [151] that in case of mechanical systems with M, E, K symmetric positive definite, this realization is strictly dissipative, i.e., E_{sd} is symmetric positive definite and $\mathsf{A}_{\mathrm{sd}} + \mathsf{A}_{\mathrm{sd}}^{\mathsf{T}}$ is symmetric negative definite. Using the realization (2.22) gives numerical advantages in computational methods that work with projected spectra of $\lambda\mathsf{E} - \mathsf{A}$ rather than directly with the second-order system matrices. But applying (2.22) comes with the cost of increased computational complexity as there are no zero blocks in the matrix structure to make use of in computational operations, in contrast to (2.18) and (2.19).

As before, the strictly dissipative realization (2.22) is equivalent to the other two realizations (2.18) and (2.19) such that again the question of appropriate transformation matrices to switch between the realizations need to be answered. While in [151] only the transformation into (2.18) with a specific choice for J_{fc}, namely $J_{\mathrm{fc}} = K$, was shown, it can be observed that with

$$Z_{\mathrm{fc2sd}} = \begin{bmatrix} J_{\mathrm{fc}}^{-\mathsf{T}} K^{\mathsf{T}} & \gamma J_{\mathrm{fc}}^{-\mathsf{T}} M^{\mathsf{T}} \\ \gamma I_{n_2} & I_{n_2} \end{bmatrix} \quad \text{and} \quad T_{\mathrm{fc2sd}} = I_{2n_2}, \tag{2.23}$$

the more general case holds

$$\mathsf{E}_{\mathrm{sd}} = Z_{\mathrm{fc2sd}}^{\mathsf{T}} \mathsf{E}_{\mathrm{fc}} T_{\mathrm{fc2sd}}, \quad \mathsf{A}_{\mathrm{sd}} = Z_{\mathrm{fc2sd}}^{\mathsf{T}} \mathsf{A}_{\mathrm{fc}} T_{\mathrm{fc2sd}}, \quad \mathsf{B}_{\mathrm{sd}} = Z_{\mathrm{fc2sd}}^{\mathsf{T}} \mathsf{B}_{\mathrm{fc}}, \quad \mathsf{C}_{\mathrm{sd}} = \mathsf{C}_{\mathrm{fc}} T_{\mathrm{fc2sd}}.$$

The inverse transformation is given by

$$Z_{\mathrm{fc2sd}}^{-1} = \begin{bmatrix} (K - \gamma^2 M)^{-\mathsf{T}} J_{\mathrm{fc}}^{\mathsf{T}} & -\gamma (K - \gamma^2 M)^{-\mathsf{T}} M^{\mathsf{T}} \\ -\gamma (K - \gamma^2 M)^{-\mathsf{T}} J_{\mathrm{fc}}^{\mathsf{T}} & (K - \gamma^2 M)^{-\mathsf{T}} K^{\mathsf{T}} \end{bmatrix} \quad \text{and} \quad T_{\mathrm{fc2sd}}^{-1} = I_{2n_2}.$$

with the additional assumption that $K - \gamma^2 M$ is invertible. The transformation of the strictly dissipative realization into the second companion form realization follows then by applying (2.20) or (2.21) to the transformations above.

As in the first-order system case, realizations of second-order systems are an important point for the application of model reduction methods. In general, the realizations of two second-order systems G_{L} and $\widetilde{G}_{\mathrm{L}}$ are equivalent if and only if there exist $Z, T \in \mathbb{C}^{n_1 \times n_1}$, with $n_1 = 2n_2$, such that corresponding first-order realizations of G_{L} and $\widetilde{G}_{\mathrm{L}}$ are equivalent. This equivalence is in a certain sense unhandy due to the resulting difficult conditions on the transformation matrices to preserve the second-order structure. A more applicable special case of second-order system equivalence is given in the next definition.

Definition 2.8 (Restricted system equivalence, e.g., [159]):
Two second-order systems

$$G_{\mathrm{L}} = (M, E, K, B_{\mathrm{u}}, C_{\mathrm{p}}, C_{\mathrm{v}}) \quad \text{and} \quad \widetilde{G}_{\mathrm{L}} = (\widetilde{M}, \widetilde{E}, \widetilde{K}, \widetilde{B}_{\mathrm{u}}, \widetilde{C}_{\mathrm{p}}, \widetilde{C}_{\mathrm{v}})$$

are called *restricted equivalent*, if there exist transformation matrices $Z, T \in \mathbb{C}^{n_2 \times n_2}$ such that

$$\begin{aligned} M &= Z^{\mathsf{H}} \widetilde{M} T, \quad E = Z^{\mathsf{H}} \widetilde{E} T, \quad K = Z^{\mathsf{H}} \widetilde{K} T, \quad B_{\mathrm{u}} = Z^{\mathsf{H}} B_{\mathrm{u}}, \\ C_{\mathrm{p}} &= \widetilde{C}_{\mathrm{p}} T, \qquad C_{\mathrm{v}} = \widetilde{C}_{\mathrm{v}} T \end{aligned} \tag{2.24}$$

hold. The change between two second-order system realizations in the sense of (2.24) is called *restricted state-space transformation*. ◊

It can be shown that the restricted system equivalence is a special case of the general equivalence of second-order systems by observing that (2.24) is obtained by setting

$$\widetilde{Z} = \begin{bmatrix} Z_{11} & 0 \\ 0 & Z \end{bmatrix} \quad \text{and} \quad \widetilde{T} = \begin{bmatrix} T & 0 \\ 0 & T \end{bmatrix}$$

as a generalized state-space transformation (2.11) to the first companion form realization (2.18), where $Z_{11} \in \mathbb{C}^{n_2 \times n_2}$ is an arbitrary invertible matrix.

Analogously to the first-order case, second-order systems can be equivalently described in the frequency domain. Applying the Laplace transformation (2.12) to (2.17) yields

$$\begin{aligned} s^2 M X(s) - s M x_{\mathrm{p},0} - M x_{\mathrm{v},0} &= -s E X(s) + E x_{\mathrm{p},0} - K X(s) + B_{\mathrm{u}} U(s), \\ Y(s) &= C_{\mathrm{p}} X(s) + s C_{\mathrm{v}} X(s) - C_{\mathrm{v}} x_{\mathrm{p},0}. \end{aligned} \tag{2.25}$$

Using the assumption $x_{\mathrm{p},0} = M x_{\mathrm{v},0} = 0$ and reordering the terms to get a direct input-to-output relation in the frequency domain results in the second-order transfer function

$$G_{\mathrm{L}}(s) = (C_{\mathrm{p}} + s C_{\mathrm{v}})(s^2 M + s E + K)^{-1} B_{\mathrm{u}}, \tag{2.26}$$

with the complex variable $s \in \mathbb{C}$. Note that equivalently, inserting any first-order realization of (2.17), e.g., (2.18), (2.19), and (2.22), into the first-order transfer function formulation (2.14) also results in (2.26).

While most system properties of second-order systems are only characterized for their first-order form, e.g., controllability and observability, the concept of asymptotic stability easily transfers to the second-order case: A second-order system (2.17) is asymptotically stable (c-stable) if and only if all eigenvalues λ of the quadratic matrix pencil $\lambda^2 M + \lambda E + K$, i.e., all $\lambda \in \mathbb{C}$ such that $\det(\lambda^2 M + \lambda E + K) = 0$, have negative real parts.

2.3 Frequency domain representations of special nonlinear systems

The second part of this thesis is concerned with model order reduction of structured systems with special nonlinearities, namely bilinear and quadratic-bilinear systems.

Therein, the idea of frequency domain representation of these two system classes is needed. This will be discussed in this section for the case of unstructured first-order systems. The concepts presented here are based on the work in [166].

2.3.1 Bilinear control systems in frequency domain

The first system class discussed are first-order (unstructured) bilinear control systems of the form

$$\begin{aligned} \mathsf{E}\dot{\mathsf{x}}(t) &= \mathsf{A}\mathsf{x}(t) + \sum_{j=1}^{m} \mathsf{N}_j \mathsf{x}(t) u_j(t) + \mathsf{B}u(t), \\ y(t) &= \mathsf{C}\mathsf{x}(t) \end{aligned} \tag{2.27}$$

with $\mathsf{E}, \mathsf{A}, \mathsf{N}_j \in \mathbb{R}^{n_1 \times n_1}$, for $j = 1, \ldots, m$, $\mathsf{B} \in \mathbb{R}^{n_1 \times m}$, $\mathsf{C} \in \mathbb{R}^{p \times n_1}$; E invertible, if not stated otherwise, and the initial condition $\mathsf{x}(t_0) = 0$ with $t_0 = 0$. Bilinear control systems (2.27) form a special class of nonlinear dynamical systems as they only involve the multiplication of control and state variables, where the inputs are written as

$$u(t) = \begin{bmatrix} u_1(t) & u_2(t) & \ldots & u_m(t) \end{bmatrix}^{\mathsf{T}},$$

i.e., these systems are linear in state and control separately, but not in the multiplication of both [145]. Therefore, bilinear systems are an important link between linear systems and systems with stronger nonlinearities.

The general idea to make (2.27) more open to known model reduction techniques is to convert (2.27) into a series of linear-like systems using the Volterra series expansion [166]. Assume the input signal u to be one-sided, i.e., $u(t) = 0$ for $t \leq 0$, then the internal state of (2.27) can be rewritten into a series of states

$$\mathsf{x}(t) = \sum_{k=1}^{\infty} \mathsf{x}_k(t), \tag{2.28}$$

where the new states $\mathsf{x}_k(t)$ are given by a sequence of coupled linear subsystems

$$\begin{aligned} \mathsf{E}\dot{\mathsf{x}}_1(t) &= \mathsf{A}\mathsf{x}_1(t) + \mathsf{B}u(t), \\ \mathsf{E}\dot{\mathsf{x}}_k(t) &= \mathsf{A}\mathsf{x}_k(t) + \sum_{j=1}^{m} \mathsf{N}_j \mathsf{x}_{k-1}(t) u_j(t), \quad \text{for } k > 1. \end{aligned} \tag{2.29}$$

The subsystem outputs are then given by multiplying the new states in (2.29) with the output matrix C and, for the overall system (2.27), by multiplying (2.28) with C. The first subsystem ($k = 1$) in (2.29) resembles the classical linear case (2.8). All further subsystems ($k > 1$) are also linear in their differential states but come with new (artificial) input signals depending on the state of the previous subsystem and the entries of the

original input. In that sense, those systems (2.29) can be treated like in the linear case and solved for the states via the variation of constants formula (2.9). Using (2.9) and (2.10) for (2.28) and (2.29) yields the *Volterra series expansion* of (2.27) to be given by

$$y(t) = \sum_{k=1}^{\infty} \int_0^t \int_0^{t_1} \cdots \int_0^{t_{k-1}} \mathsf{g}_{\mathrm{B},k}(t_1, \ldots, t_k) \left(u(t - \sum_{i=1}^{j} t_i) \otimes \cdots \otimes u(t - t_1) \right) \mathrm{d}t_k \cdots \mathrm{d}t_1. \quad (2.30)$$

The time-dependent multivariate functions $\mathsf{g}_{\mathrm{B},k}$, for $k \geq 1$, are the *regular Volterra kernels* of (2.27), with

$$\begin{aligned} \mathsf{g}_{\mathrm{B},k}(t_1, \ldots, t_k) = \mathsf{C} e^{\mathsf{E}^{-1}\mathsf{A}t_k} \left(\prod_{j=1}^{k-1} (I_{m^{j-1}} \otimes \mathsf{E}^{-1}\mathsf{N})(I_{m^j} \otimes e^{\mathsf{E}^{-1}\mathsf{A}t_{k-j}}) \right) \\ \times (I_{m^{k-1}} \otimes \mathsf{E}^{-1}\mathsf{B}), \end{aligned} \quad (2.31)$$

where the bilinear terms were concatenated into $\mathsf{N} = \begin{bmatrix} \mathsf{N}_1 & \ldots & \mathsf{N}_m \end{bmatrix}$.

In the linear system case (2.8), the classical Laplace transformation (2.12) is used to transform the kernel in the variation of constants formula (2.10) into the transfer function (2.14) to describe the input-to-output behavior of the system in the frequency domain. In case of bilinear systems, the Volterra kernels (2.31) play this role and together with the multivariate extension of the Laplace transformation [166] yield the *regular subsystem transfer functions* of (2.27) to be given by

$$\begin{aligned} \mathsf{G}_{\mathrm{B},k}(s_1, \ldots, s_k) = \mathsf{C}(s_k\mathsf{E} - \mathsf{A})^{-1} \left(\prod_{j=1}^{k-1} (I_{m^{j-1}} \otimes \mathsf{N})(I_{m^j} \otimes (s_{k-j}\mathsf{E} - \mathsf{A})^{-1}) \right) \\ \times (I_{m^{k-1}} \otimes \mathsf{B}), \end{aligned} \quad (2.32)$$

with the complex variables $s_1, \ldots, s_k \in \mathbb{C}$. The compact expression (2.32) is actually the collection of the different combinations of multiplications of the linear dynamics with the m bilinear terms, i.e., by multiplying out the Kronecker products, (2.32) resembles the column concatenation of the multiplications with the different bilinear terms

$$\begin{aligned} \mathsf{G}_{\mathrm{B},k}(s_1, \ldots, s_k) = \Big[& \mathsf{C}(s_k\mathsf{E} - \mathsf{A})^{-1}\mathsf{N}_1 \cdots \mathsf{N}_1(s_1\mathsf{E} - \mathsf{A})^{-1}\mathsf{B}, \\ & \mathsf{C}(s_k\mathsf{E} - \mathsf{A})^{-1}\mathsf{N}_1 \cdots \mathsf{N}_2(s_1\mathsf{E} - \mathsf{A})^{-1}\mathsf{B}, \\ & \ldots \\ & \mathsf{C}(s_k\mathsf{E} - \mathsf{A})^{-1}\mathsf{N}_1 \cdots \mathsf{N}_m(s_1\mathsf{E} - \mathsf{A})^{-1}\mathsf{B}, \\ & \ldots \\ & \mathsf{C}(s_k\mathsf{E} - \mathsf{A})^{-1}\mathsf{N}_m \cdots \mathsf{N}_m(s_1\mathsf{E} - \mathsf{A})^{-1}\mathsf{B} \Big]. \end{aligned} \quad (2.33)$$

In case of SISO bilinear systems, $m = p = 1$, only a single bilinear term is present $\mathsf{N} = \mathsf{N}_1$. Then, the multivariate transfer functions (2.32) simplify essentially to

$$\mathsf{G}_{\mathrm{B},k}(s_1, \ldots, s_k) = \mathsf{C}(s_k\mathsf{E} - \mathsf{A})^{-1} \left(\prod_{j=1}^{k-1} \mathsf{N}(s_{k-j}\mathsf{E} - \mathsf{A})^{-1} \right) \mathsf{B}, \tag{2.34}$$

since all the Kronecker products become simple matrix multiplications. Note that (2.32) and (2.34) can also be formulated in case of a singular E matrix if the matrix pencil corresponding to the linear system part is regular, i.e., there exists a $\lambda_0 \in \mathbb{C}$ such that $\lambda_0\mathsf{E} - \mathsf{A}$ is invertible.

2.3.2 Quadratic-bilinear systems in frequency domain

The second system class to be discussed are quadratic-bilinear systems. These can be seen as an extension of bilinear systems by adding a quadratic nonlinearity.

2.3.2.1 Volterra series expansion of quadratic-bilinear systems

First-order (unstructured) quadratic-bilinear systems have the form

$$\begin{aligned} \mathsf{E}\dot{\mathsf{x}}(t) &= \mathsf{A}\mathsf{x}(t) + \mathsf{H}\big(\mathsf{x}(t) \otimes \mathsf{x}(t)\big) + \sum_{j=1}^{m} \mathsf{N}_j \mathsf{x}(t) u_j(t) + \mathsf{B}u(t), \\ y(t) &= \mathsf{C}\mathsf{x}(t), \end{aligned} \tag{2.35}$$

with $\mathsf{E}, \mathsf{A}, \mathsf{N}_j \in \mathbb{R}^{n_1 \times n_1}$, for $j = 1, \ldots, m$, $\mathsf{H} \in \mathbb{R}^{n_1 \times n_1^2}$, $\mathsf{B} \in \mathbb{R}^{n_1 \times m}$, $\mathsf{C} \in \mathbb{R}^{p \times n_1}$. Similar to the linear and bilinear system cases, the E matrix is assumed to be invertible, if not stated otherwise, and the initial condition of (2.35) is assumed to be $\mathsf{x}(t_0) = 0$ with $t_0 = 0$.

To get a frequency domain representation of (2.35) in terms of transfer functions, similar to the bilinear system case, the Volterra series expansion can be used [166]. Following the idea in [101], a scaled input signal $\alpha u(t)$, with scaling factor $0 < \alpha \in \mathbb{R}$, is applied to (2.35) and the state is assumed to have an analytic representation in terms of a power series

$$\mathsf{x}(t) = \sum_{k=1}^{\infty} \alpha^k \mathsf{x}_k(t), \tag{2.36}$$

where x_k are auxiliary states from linear subsystems. Now, inserting (2.36) into (2.35) yields a representation of the states x_k in terms of coupled linear subsystems by sorting the emerging components with respect to the power of the scaling factor α they are

multiplied with. The first three coupled subsystems are then given by

$$\begin{aligned}
\mathsf{E}\dot{\mathsf{x}}_1(t) &= \mathsf{A}\mathsf{x}_1(t) + \mathsf{B}u(t),\\
\mathsf{E}\dot{\mathsf{x}}_2(t) &= \mathsf{A}\mathsf{x}_2(t) + \mathsf{H}\Big(\mathsf{x}_1(t) \otimes \mathsf{x}_1(t)\Big) + \sum_{j=1}^{m} \mathsf{N}_j \mathsf{x}_1(t) u_j(t),\\
\mathsf{E}\dot{\mathsf{x}}_3(t) &= \mathsf{A}\mathsf{x}_3(t) + \mathsf{H}\Big(\mathsf{x}_1(t) \otimes \mathsf{x}_2(t) + \mathsf{x}_2(t) \otimes \mathsf{x}_1(t)\Big) + \sum_{j=1}^{m} \mathsf{N}_j \mathsf{x}_2(t) u_j(t).
\end{aligned}$$

Applying the variation of constants formula (2.10) to the coupled linear subsystems and reordering the variables yields a Volterra series expansion of (2.35) with

$$y(t) = \sum_{k=1}^{\infty} \int_0^t \int_0^{t_1} \cdots \int_0^{t_{k-1}} \mathsf{g}_{\mathrm{Q},k}(t_1, \ldots, t_k)\Big(u(t-t_1) \otimes \cdots \otimes u(t-t_k)\Big) \mathrm{d}t_k \cdots \mathrm{d}t_1. \tag{2.37}$$

The functions $\mathsf{g}_{\mathrm{Q},k}(t_1, \ldots, t_k)$ are the Volterra kernels of the corresponding Volterra series representation, e.g., symmetric kernels are used in (2.37). Applying the multivariate Laplace transformation [166] to (2.37) results in a frequency domain representation of (2.35). Depending on the chosen kernels in the Volterra series (2.37), there are different transfer function representations of (2.35) known in the literature. In the following, the three most commonly used types are described.

2.3.2.2 Symmetric subsystem transfer functions

Historically, the first concept to represent quadratic-bilinear systems in the frequency domain are the symmetric transfer functions [30, 101]. In case of MIMO systems (2.35), these transfer functions can, in general, be written as

$$\mathsf{G}_{\mathrm{Q,sym},k}(s_1, \ldots, s_k) = \mathsf{CS}_{\mathrm{Q,sym},k}(s_1, \ldots, s_k), \tag{2.38}$$

with the complex variables $s_1, \ldots, s_k \in \mathbb{C}$ and $k \geq 1$, following the recursion

$$\begin{aligned}
\mathsf{S}_{\mathrm{Q,sym},1}(s_1) &= (s_1\mathsf{E} - \mathsf{A})^{-1}\mathsf{B}, \\
\mathsf{S}_{\mathrm{Q,sym},k}(s_1, \ldots, s_k) &= \frac{1}{k!}\left(\left(\sum_{j=1}^{k} s_k\right)\mathsf{E} - \mathsf{A}\right)^{-1} \\
&\quad \times \left(\mathsf{H}\left(\sum_{j=1}^{k-1}\left(\sum_{\substack{1\leq\alpha_1<\ldots<\alpha_j\leq k \\ 1\leq\alpha_{j+1}<\ldots<\alpha_k\leq k \\ \alpha_i\neq\alpha_\ell \text{ for } i\neq\ell}} \mathsf{S}_{\mathrm{Q,sym},j}(s_{\alpha_1}, \ldots, s_{\alpha_j})\right.\right.\right. \\
&\qquad \left.\left.\left. \otimes\, \mathsf{S}_{\mathrm{Q,sym},k-j}(s_{\alpha_{j+1}}, \ldots, s_{\alpha_k})\right)\right)\right. \\
&\quad \left. + \mathsf{N}\left(I_m \otimes \left(\sum_{1\leq\beta_1<\ldots<\beta_{k-1}\leq k} \mathsf{S}_{\mathrm{Q,sym},k-1}(s_{\beta_1}, \ldots, s_{\beta_{k-1}})\right)\right)\right).
\end{aligned} \tag{2.39}$$

For illustration of (2.38) and (2.39), the first three symmetric subsystem transfer functions of (2.35) for the SISO case are given by

$$\begin{aligned}
\mathsf{G}_{\mathrm{Q,sym},1}(s_1) &= \mathsf{C}\mathsf{S}_{\mathrm{Q,sym},1}(s_1), \\
\mathsf{G}_{\mathrm{Q,sym},2}(s_1, s_2) &= \mathsf{C}\mathsf{S}_{\mathrm{Q,sym},2}(s_1, s_2), \\
\mathsf{G}_{\mathrm{Q,sym},3}(s_1, s_2, s_3) &= \mathsf{C}\mathsf{S}_{\mathrm{Q,sym},3}(s_1, s_2, s_3),
\end{aligned}$$

with the recursive terms

$$\begin{aligned}
\mathsf{S}_{\mathrm{Q,sym},1}(s_1) &= (s_1\mathsf{E} - \mathsf{A})^{-1}\mathsf{B}, \\
\mathsf{S}_{\mathrm{Q,sym},2}(s_1, s_2) &= \frac{1}{2}\Big((s_1 + s_2)\mathsf{E} - \mathsf{A}\Big)^{-1}\Big(\mathsf{H}\Big(\mathsf{S}_{\mathrm{Q,sym},1}(s_1) \otimes \mathsf{S}_{\mathrm{Q,sym},1}(s_2) \\
&\quad + \mathsf{S}_{\mathrm{Q,sym},1}(s_2) \otimes \mathsf{S}_{\mathrm{Q,sym},1}(s_1)\Big) \\
&\quad + \mathsf{N}\Big(\mathsf{S}_{\mathrm{Q,sym},1}(s_1) + \mathsf{S}_{\mathrm{Q,sym},1}(s_2)\Big)\Big),
\end{aligned}$$

for the first two subsystems, and

$$\begin{aligned}\mathsf{S}_{\mathrm{Q,sym},3}(s_1,s_2,s_3) = \frac{1}{6}\Big((s_1+s_2+s_3)\mathsf{E}-\mathsf{A}\Big)^{-1}\Big(\mathsf{H}\Big(&\mathsf{S}_{\mathrm{Q,sym},1}(s_1)\otimes\mathsf{S}_{\mathrm{Q,sym},2}(s_2,s_3)\\ &+\mathsf{S}_{\mathrm{Q,sym},1}(s_2)\otimes\mathsf{S}_{\mathrm{Q,sym},2}(s_1,s_3)+\mathsf{S}_{\mathrm{Q,sym},1}(s_3)\otimes\mathsf{S}_{\mathrm{Q,sym},2}(s_1,s_2)\\ &+\mathsf{S}_{\mathrm{Q,sym},2}(s_1,s_2)\otimes\mathsf{S}_{\mathrm{Q,sym},1}(s_3)+\mathsf{S}_{\mathrm{Q,sym},2}(s_1,s_3)\otimes\mathsf{S}_{\mathrm{Q,sym},1}(s_2)\\ &+\mathsf{S}_{\mathrm{Q,sym},2}(s_2,s_3)\otimes\mathsf{S}_{\mathrm{Q,sym},1}(s_1)\Big)\\ &+\mathsf{N}\Big(\mathsf{S}_{\mathrm{Q,sym},2}(s_1,s_2)+\mathsf{S}_{\mathrm{Q,sym},2}(s_1,s_3)+\mathsf{S}_{\mathrm{Q,sym},2}(s_2,s_3)\Big)\Big),\end{aligned}$$

for the third one. A general advantage of symmetric subsystem transfer functions is that, by construction, their evaluation is independent of the ordering of their frequency arguments. For example, the second symmetric subsystem transfer function always satisfies

$$\mathsf{G}_{\mathrm{Q,sym},2}(\sigma_1,\sigma_2) = \mathsf{G}_{\mathrm{Q,sym},2}(\sigma_2,\sigma_1),$$

for all $\sigma_1,\sigma_2\in\mathbb{C}$ in which $\mathsf{G}_{\mathrm{Q,sym},2}$ is defined. On the other hand, a drawback of symmetric transfer functions is the exponentially growing number of frequency-dependent terms in the recursion formula (2.39). This leads to high computational costs for the evaluation of higher-level symmetric subsystem transfer functions.

2.3.2.3 Regular subsystem transfer functions

The second type of transfer functions to be discussed was developed to compete with the problem of the exponentially growing number of frequency-dependent terms in symmetric transfer functions. Introduced in [4], the regular subsystem transfer functions of (2.35) for MIMO systems can be written as

$$\mathsf{G}_{\mathrm{Q,reg},k}(s_1,\ldots,s_k) = \mathsf{C}\mathsf{S}_{\mathrm{Q,reg},k}(s_1,\ldots,s_k), \tag{2.40}$$

with the complex variables $s_1,\ldots,s_k\in\mathbb{C}$ and $k\geq 1$, following the recursion

$$\begin{aligned}\mathsf{S}_{\mathrm{Q,reg},1}(s_1) &= (s_1\mathsf{E}-\mathsf{A})^{-1}\mathsf{B},\\ \mathsf{S}_{\mathrm{Q,reg},k}(s_1,\ldots,s_k) &= (s_kE-A)^{-1}\\ &\quad\times\Bigg(\mathsf{H}\Bigg(\sum_{j=1}^{k-1}\mathsf{S}_{\mathrm{Q,reg},j}(s_{k-j+1}-s_{k-j},\ldots,s_k-s_{k-j})\\ &\quad\otimes\mathsf{S}_{\mathrm{Q,reg},k-j}(s_1,\ldots,s_{k-j})\Bigg)\\ &\quad+\mathsf{N}(I_m\otimes\mathsf{S}_{\mathrm{Q,reg},k-1}(s_1,\ldots,s_{k-1}))\Bigg).\end{aligned} \tag{2.41}$$

For illustration of (2.40) and (2.41), and comparison to the symmetric transfer function case, the first three regular subsystem transfer functions of (2.35) for the SISO case are given by

$$\begin{aligned}
\mathsf{G}_{\mathrm{Q,reg},1}(s_1) &= \mathsf{CS}_{\mathrm{Q,reg},1}(s_1),\\
\mathsf{G}_{\mathrm{Q,reg},2}(s_1,s_2) &= \mathsf{CS}_{\mathrm{Q,reg},2}(s_1,s_2),\\
\mathsf{G}_{\mathrm{Q,reg},3}(s_1,s_2,s_3) &= \mathsf{CS}_{\mathrm{Q,reg},3}(s_1,s_2,s_3),
\end{aligned}$$

with the recursive terms

$$\begin{aligned}
\mathsf{S}_{\mathrm{Q,reg},1}(s_1) &= (s_1\mathsf{E}-\mathsf{A})^{-1}\mathsf{B},\\
\mathsf{S}_{\mathrm{Q,reg},2}(s_1,s_2) &= (s_2\mathsf{E}-\mathsf{A})^{-1}\Big(\mathsf{H}\big(\mathsf{S}_{\mathrm{Q,reg},1}(s_2-s_1)\otimes\mathsf{S}_{\mathrm{Q,reg},1}(s_1)\big)+\mathsf{NS}_{\mathrm{Q,reg},1}(s_1)\Big),\\
\mathsf{S}_{\mathrm{Q,reg},3}(s_1,s_2,s_3) &= (s_3\mathsf{E}-\mathsf{A})^{-1}\Big(\mathsf{H}\big(\mathsf{S}_{\mathrm{Q,reg},1}(s_3-s_2)\otimes\mathsf{S}_{\mathrm{Q,reg},2}(s_1,s_2)\\
&\quad+\mathsf{S}_{\mathrm{Q,reg},2}(s_2-s_1,s_3-s_1)\otimes\mathsf{S}_{\mathrm{Q,reg},1}(s_1)\big)+\mathsf{NS}_{\mathrm{Q,reg},2}(s_1,s_2)\Big).
\end{aligned}$$

In comparison to the symmetric subsystem transfer functions, the regular case has less recursive terms and is therefore easier to evaluate, while still corresponding to a Volterra series representation of (2.35) in terms of regular Volterra kernels. Also, note that the regular subsystem transfer functions of quadratic-bilinear systems are a direct extension of the regular transfer functions of purely bilinear systems (2.32).

2.3.2.4 Generalized transfer functions

Taking a closer look at (2.39) and (2.41) reveals that both transfer function types contain linear combinations of similarly structured terms, which are multiplications of the matrices from the linear, bilinear and quadratic system parts. In that sense and inspired by the purely bilinear system case (2.32), where the transfer functions are only products of the matrices from the linear and bilinear components, the authors of [92] suggested a generalized transfer function concept for quadratic-bilinear systems. These generalized transfer functions are restricted to using only multiplications of the different system terms. A simplification of this approach was used in [39] for systems with polynomial nonlinearities. Some reformulations of the ideas in [92] allow the extension of the generalized transfer functions to MIMO systems. Let the following function model the recursive application of the linear dynamics to the matrices corresponding to the

input, bilinear or quadratic system components:

$$\Gamma(\gamma, s_1, \ldots, s_k) = \begin{cases} (s_1\mathsf{E} - \mathsf{A})^{-1}\mathsf{B}, & \text{if } \gamma = (\mathrm{B}) \\ & \quad \text{and } k = 1, \\ (s_j\mathsf{E} - \mathsf{A})^{-1}\mathsf{N}\Big(I_m \otimes \Gamma(\gamma_2, s_1, \ldots, s_{k-1})\Big), & \text{if } \gamma = (\mathrm{N}, \gamma_2) \\ & \quad \text{and } k \geq 2, \\ (s_j\mathsf{E} - \mathsf{A})^{-1}\mathsf{H}\Big(\Gamma(\gamma_2, s_\ell, \ldots, s_{k-1}) & \text{if } \gamma = (\mathrm{H}, \gamma_2, \gamma_3) \\ \quad \otimes \Gamma(\gamma_3, s_1, \ldots, s_{\ell-1})\Big) & \quad \text{and } k \geq 3, \end{cases} \tag{2.42}$$

with the complex variables $s_1, \ldots, s_k \in \mathbb{C}$ and γ, a nested tuple with the possible elements H, N and B, and tuples of these. The number ℓ in the quadratic case is uniquely determined by the two sub-tuples γ_2 and γ_3. With (2.42), the generalized transfer functions of (2.35) are given by

$$\mathsf{G}^{\gamma}_{\mathrm{Q,gen},k}(s_1, \ldots, s_k) = \mathsf{C}\Gamma(\gamma, s_1, \ldots, s_k). \tag{2.43}$$

As in previous sections, the first three generalized transfer functions are considered for the SISO case as illustration and comparison to the other concepts. Note that due to the choice of γ, there may exist several different k-th-level generalized transfer functions. As in the symmetric and regular cases, the first transfer function is unique and resembles the linear system case

$$\mathsf{G}^{(\mathrm{B})}_{\mathrm{Q,gen},1}(s_1) = \mathsf{C}(s_1\mathsf{E} - \mathsf{A})^{-1}\mathsf{B}.$$

Also, the second transfer function is uniquely given by

$$\mathsf{G}^{(\mathrm{N},(\mathrm{B}))}_{\mathrm{Q,gen},2}(s_1, s_2) = \mathsf{C}(s_2\mathsf{E} - \mathsf{A})^{-1}\mathsf{N}(s_1\mathsf{E} - \mathsf{A})^{-1}\mathsf{B},$$

which is also the second regular subsystem transfer function of bilinear systems (2.27). For the third level, two different choices of transfer functions are possible depending on the nested tuple γ:

$$\begin{aligned} \mathsf{G}^{(\mathrm{N},(\mathrm{N},(\mathrm{B})))}_{\mathrm{Q,gen},3}(s_1, s_2, s_3) &= \mathsf{C}(s_3\mathsf{E} - \mathsf{A})^{-1}\mathsf{N}(s_2\mathsf{E} - \mathsf{A})^{-1}\mathsf{N}(s_1\mathsf{E} - \mathsf{A})^{-1}\mathsf{B}, \\ \mathsf{G}^{(\mathrm{H},(\mathrm{B}),(\mathrm{B}))}_{\mathrm{Q,gen},3}(s_1, s_2, s_3) &= \mathsf{C}(s_3\mathsf{E} - \mathsf{A})^{-1}\mathsf{H}\Big((s_2\mathsf{E} - \mathsf{A})^{-1}\mathsf{B} \otimes (s_1\mathsf{E} - \mathsf{A})^{-1}\mathsf{B}\Big). \end{aligned}$$

To further illustrate the role of the nested tuple γ, consider as example the SISO transfer function with $\gamma = (\mathrm{H}, (\mathrm{N}, (\mathrm{B})), (\mathrm{H}, (\mathrm{B}), (\mathrm{B})))$, which yields

$$\begin{aligned}
\mathsf{G}_{\mathrm{Q,gen},6}^{(\mathrm{H},(\mathrm{N},(\mathrm{B})),(\mathrm{H},(\mathrm{B}),(\mathrm{B})))}(s_1,\ldots,s_6) &= \mathsf{C}\Gamma((\mathrm{H},(\mathrm{N},(\mathrm{B})),(\mathrm{H},(\mathrm{B}),(\mathrm{B}))),s_1,\ldots,s_5,s_6)\\
&= \mathsf{C}(s_6\mathsf{E}-\mathsf{A})^{-1}\mathsf{H}\Big(\Gamma((\mathrm{N},(\mathrm{B})),s_4,s_5)\\
&\quad \otimes \Gamma((\mathrm{H},(\mathrm{B}),(\mathrm{B})),s_1,s_2,s_3)\Big)\\
&= \mathsf{C}(s_6\mathsf{E}-\mathsf{A})^{-1}\mathsf{H}\Big((s_5\mathsf{E}-\mathsf{A})^{-1}\mathsf{N}(s_4\mathsf{E}-\mathsf{A})^{-1}\mathsf{B}\\
&\quad \otimes (s_3\mathsf{E}-\mathsf{A})^{-1}\mathsf{H}\Big((s_2\mathsf{E}-\mathsf{A})^{-1}\mathsf{B}\otimes(s_1\mathsf{E}-\mathsf{A})^{-1}\mathsf{B}\Big)\Big).
\end{aligned}$$

Remark 2.9 (Transfer function levels):
The transfer function levels of the symmetric and regular cases do not necessarily correspond to those of the generalized transfer functions due to the additional freedom of choosing two unrelated frequency arguments for the quadratic term. For example, the second generalized transfer function is uniquely determined with only the bilinear terms involved, while in the symmetric and regular cases the quadratic term is already concerned in the second subsystem transfer functions. ◊

2.4 Setup for numerical experiments

Numerical experiments will be performed in this thesis for demonstration and comparison of the developed model order reduction techniques. To ensure proper, fair and reproducible computations and comparisons, the following two sections state hardware and software used in the computations as well as the basic idea of the MORscore used for the comparison of different model reduction methods.

2.4.1 Hardware and software environments

All experiments reported in this thesis have been carried out on nodes of the compute cluster `mechthild` at the *Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg*. The fundamental hardware and software specifications of the two types of compute nodes used for the experiments are listed in Table 2.1.

Each experiment was executed on a single node of either `standard` or `big-memory` type, depending on the demand for main memory of the experiment. It should be mentioned that large parts of the experiments could also be executed on less powerful hardware, e.g., with smaller amounts of main memory. In other words, the quantities in Table 2.1 are not necessarily the required computational resources for the evaluation of the dynamical systems or the computation of reduced-order models in the experiments. For the reason

Table 2.1: Hardware and software environments for numerical experiments.

CPU	2× Intel® Xeon® Silver 4110 (Skylake) @ 2.10 GHz (3.0 GHz Turbo)
Cores	2×8
RAM	192 GB DDR4 with ECC (`standard`)
	384 GB DDR4 with ECC (`big-memory`)
OS	CentOS Linux release 7.5.1804
Platform	`x86_64` (64 Bit)
MATLAB	9.7.0.1296695 (R2019b) [139]

of comparability and reproducibility, nevertheless, all computations were performed on the same mentioned hardware.

Furthermore, the following free, publicly available open-source MATLAB packages were used in the computations:

- *M-M.E.S.S.* version 2.0.1 [44, 167], for solving large-scale sparse matrix equations,
- *MORLAB* version 5.0 [55, 56], for model reduction methods and matrix equation solvers for medium-scale dense systems, and evaluation of linear systems in time and frequency domain,
- *SOLBT* version 3.0 [58], for solving large-scale sparse Lyapunov equations with right-hand side matrix functions arising in limited balanced truncation methods, and
- *SOMDDPA* version 2.0 [59], for the second-order modally damped dominant pole algorithm.

Code availability
The source codes and scripts used to compute the results presented in this thesis can be obtained from

doi:10.5281/zenodo.4650402

under the BSD-2-Clause license, and the computed results are available at

doi:10.5281/zenodo.4650422

under the CC BY 4.0 license. Both are authored by Steffen W. R. Werner.

2.4.2 Comparison of model reduction methods in the MORscore

To evaluate the performance of model reduction methods, a common approach is the comparison of model reduction errors for varying orders. Nevertheless, a one-by-one comparison for multiple different model reduction methods quickly becomes too cumbersome or too complex to extract proper decisions about the performance. Inspired by the so-called *minimal realization profiles* from the optimization community [74], the MORscore was introduced in [111] to compress the performance of model reduction methods in various measures into scalar values.

Definition 2.10 (MORscore [111]):
Given a graph $(r, \varepsilon(r)) \in \mathbb{N}_0 \times (0, 1]$ relating a reduced order r to a relative output error $\varepsilon(r)$ of a model reduction method, the normalized error graph $(\varphi_r, \varphi_{\varepsilon(r)})$ is determined via the two mappings:

$$\varphi_r \colon r \mapsto \frac{r}{r_{\max}}, \quad \text{and} \quad \varphi_{\varepsilon(r)} \colon \varepsilon(r) \mapsto \frac{\log_{10}\big(\varepsilon(r)\big)}{\lfloor \log_{10}(\epsilon_{\text{mach}}) \rfloor},$$

with a maximum reduced order $r_{\max} \in \mathbb{N}$ and the used machine precision $\epsilon_{\text{mach}} \in (0, 1]$. The *MORscore* is then defined to be the area under the normalized error graph $(\varphi_r, \varphi_{\varepsilon(r)})$.$\Diamond$

The normalized error graph in Definition 2.10 is a mapping of the relative model reduction error varying with the reduced order into the unit square such that the MORscore will be a value in $[0, 1]$. Note that in contrast to [111], Definition 2.10 actively includes the case of reduced-order models of order 0 with the corresponding relative approximation error of 1 as starting point of the error graphs. The maximum reduced order $r_{\max}$ should be a reasonable small number compared to the full system order, $r_{\max} \ll n$, since first, usually it is too computationally costly to compute all possible reduced-order models up to the original system size, and second, the MORscore would not show much difference if the minimal relative error for the model reduction methods is attained earlier than for the full order. In computations using double precision, the machine epsilon is given with $\epsilon_{\text{mach}} \approx 2.22 \cdot 10^{-16}$ such that $\lfloor \log_{10}(\epsilon_{\text{mach}}) \rfloor = -16$. The normalized error graph in Definition 2.10 assumes the relative error to be smaller or equal to 1. This is not always the case, for example, when using a time domain measure for an unstable performing reduced-order model, or when approximations are simply too bad. In these cases, the relative model reduction error is restricted to 1 as this becomes a 0 in the normalized error graph. In practical implementations, the MORscore can easily be computed using the trapezoidal rule (in MATLAB `trapz`). In general, a larger MORscore belongs to the better model reduction method. It can be interpreted as a faster decay of the considered error measure.

For the comparison of model reduction methods in this thesis, only approximate norms will be used for computational reasons. In time domain, approximations of the L_2- and

L_∞-norms from (2.15) and (2.16) are used. Therefore, let y be the output signal of the original system and $\hat{y}$ the output of the reduced-order model, the absolute error in the approximate L_2-norm is then given by

$$\|y - \hat{y}\|_{L_2} \approx \sqrt{\tau}\|\mathrm{vec}(y_\mathrm{h} - \hat{y}_\mathrm{h})\|_2, \tag{2.44}$$

where $y_\mathrm{h} \in \mathbb{R}^{p\times n_\tau}$ and $\hat{y}_\mathrm{h} \in \mathbb{R}^{p\times n_\tau}$ are the discretized output signals of the full and reduced-order models, respectively, in the time interval $[t_0, t_\mathrm{f}]$ and with step size τ. The absolute error in the approximate L_∞-norm is then given by

$$\|y - \hat{y}\|_{L_\infty} \approx \|\mathrm{vec}(y_\mathrm{h} - \hat{y}_\mathrm{h})\|_\infty; \tag{2.45}$$

see, e.g., [111]. In frequency domain, the absolute error in the approximated $\mathcal{H}_\infty/\mathcal{L}_\infty$-norm will be used with

$$\|\mathcal{G} - \widehat{\mathcal{G}}\|_{\mathcal{H}_\infty/\mathcal{L}_\infty} \approx \max_{\omega_k}\|\mathcal{G}(\omega_k\mathrm{i}) - \widehat{\mathcal{G}}(\omega_k\mathrm{i})\|_2, \tag{2.46}$$

for the full- and reduced-order transfer functions $\mathcal{G}$ and $\widehat{\mathcal{G}}$, and discrete frequency evaluation points $\omega_k \in [\omega_\mathrm{min}, \omega_\mathrm{max}]$. For a more diverse notation in this thesis and since $\mathcal{H}_\infty$- and $\mathcal{L}_\infty$-norms have the same definition, both will be denoted by $\mathcal{H}_\infty$ in the upcoming numerical experiments. Note that here only absolute errors are depicted for illustration of the approximate norms. For the MORscore, these absolute errors still need to be divided by the approximate norms of the output signal or transfer function of the full-order model (FOM). For example, in upcoming MORscore tables, columns denoted by L_∞ correspond to the approximate L_∞-norm measure (2.45) such that in the underlying error graphs, the approximate relative L_∞-error is used with

$$\frac{\|y - \hat{y}\|_{L_\infty}}{\|y\|_{L_\infty}} \approx \frac{\|\mathrm{vec}(y_\mathrm{h} - \hat{y}_\mathrm{h})\|_\infty}{\|\mathrm{vec}(y_\mathrm{h})\|_\infty}.$$

Adjustments and presentation of the approximate norms will be explained when needed in sections with numerical experiments.

CHAPTER 3

BASICS OF LINEAR MODEL ORDER REDUCTION

Contents

This chapter is used to introduce basic ideas and concepts from the literature for model order reduction of linear first- and second-order systems. In Section 3.1, the projection framework for model reduction is established as the main construction approach for reduced-order models in this thesis. Thereafter, state-of-the-art methods in modal truncation, structured interpolation and balanced truncation for first- and second-order systems are presented.

3.1 Model reduction by projection

In general, model order reduction describes the process of simplifying dynamical systems by reducing the internal state-space dimension and number of differential equations, leading to easier-to-evaluate models that can be used as surrogates in applications. For linear first-order systems

$$\mathsf{G}_{\mathrm{L}} : \begin{cases} \mathsf{E}\dot{\mathsf{x}}(t) = \mathsf{A}\mathsf{x}(t) + \mathsf{B}u(t), \\ y(t) = \mathsf{C}\mathsf{x}(t), \end{cases} \tag{2.8}$$

the model order reduction problem is given as the construction of reduced-order systems of the form

$$\widehat{\mathsf{G}}_{\mathrm{L}} : \begin{cases} \widehat{\mathsf{E}}\dot{\hat{\mathsf{x}}}(t) = \widehat{\mathsf{A}}\hat{\mathsf{x}}(t) + \widehat{\mathsf{B}}u(t), \\ \hat{y}(t) = \widehat{\mathsf{C}}\hat{\mathsf{x}}(t), \end{cases} \tag{3.1}$$

with $\widehat{\mathsf{E}}, \widehat{\mathsf{A}} \in \mathbb{R}^{r_1 \times r_1}$, $\widehat{\mathsf{B}} \in \mathbb{R}^{r_1 \times m}$, $\widehat{\mathsf{C}} \in \mathbb{R}^{p \times r_1}$ and a much smaller number of internal states and differential equations $r_1 \ll n_1$. The new system (3.1) is constructed to approximate the input-to-output behavior of the original system (2.8) in the sense of (1.2).

A commonly used approach for the construction of (3.1) is the *projection framework*. Therefore, let $V \in \mathbb{C}^{n_1 \times r_1}$ be a basis matrix of the underlying right projection space $\operatorname{span}(V)$ such that $x \approx V\hat{\mathsf{x}}$. Choosing a left projection space $\operatorname{span}(W)$ with a corresponding truncation matrix $W \in \mathbb{C}^{n_1 \times r_1}$, the reduced-order model (3.1) is computed by

$$\widehat{\mathsf{E}} = W^{\mathsf{H}}\mathsf{E}V, \quad \widehat{\mathsf{A}} = W^{\mathsf{H}}\mathsf{A}V, \quad \widehat{\mathsf{B}} = W^{\mathsf{H}}\mathsf{B}, \quad \widehat{\mathsf{C}} = \mathsf{C}V. \tag{3.2}$$

This is further described in, e.g., [9, 184]. In the context of finite element methods, $\operatorname{span}(V)$ would be known as the ansatz space and $\operatorname{span}(W)$ as the test space.

In principle, second-order systems (2.17) can be rewritten into first-order form, e.g., using one of the first-order realizations in Section 2.2.2, and then reduced by a model reduction method for first-order systems. This results in a reduced-order system of the form (3.1), which usually cannot be transformed back into second-order form. This yields certain disadvantages, as missing physical interpretation of the reduced-order system quantities, lesser approximation accuracy for the same reduced order or the change of tools used in applications for the system class. The main goal in this thesis is the construction of structure-preserving reduced-order models, i.e., given the original second-order system

$$G_{\mathrm{L}} : \begin{cases} M\ddot{x}(t) + E\dot{x}(t) + Kx(t) = B_{\mathrm{u}}u(t), \\ y(t) = C_{\mathrm{p}}x(t) + C_{\mathrm{v}}\dot{x}(t), \end{cases} \tag{2.17}$$

the task is to compute a reduced-order model of the same form

$$\widehat{G}_{\mathrm{L}} : \begin{cases} \widehat{M}\ddot{\hat{x}}(t) + \widehat{E}\dot{\hat{x}}(t) + \widehat{K}\hat{x}(t) = \widehat{B}_{\mathrm{u}}u(t), \\ \hat{y}(t) = \widehat{C}_{\mathrm{p}}\hat{x}(t) + \widehat{C}_{\mathrm{v}}\dot{\hat{x}}(t), \end{cases} \tag{3.3}$$

with $\widehat{M}, \widehat{E}, \widehat{K} \in \mathbb{R}^{r_2 \times r_2}$, $\widehat{B}_{\mathrm{u}} \in \mathbb{R}^{r_2 \times m}$, $\widehat{C}_{\mathrm{p}}, \widehat{C}_{\mathrm{v}} \in \mathbb{R}^{p \times r_2}$ and $r_2 \ll n_2$. Therefore, the projection framework (3.2) is extended in the sense of the restricted state-space transformation (Definition 2.8) for the construction of (3.3). Choosing two truncation matrices $V, W \in \mathbb{C}^{n_2 \times r_2}$, the reduced-order model (3.3) is then constructed by

$$\begin{aligned} &\widehat{M} = W^{\mathsf{H}}MV, \quad \widehat{E} = W^{\mathsf{H}}EV, \quad \widehat{K} = W^{\mathsf{H}}KV, \quad \widehat{B}_{\mathrm{u}} = W^{\mathsf{H}}B_{\mathrm{u}}, \\ &\widehat{C}_{\mathrm{p}} = C_{\mathrm{p}}V, \qquad \widehat{C}_{\mathrm{v}} = C_{\mathrm{v}}V. \end{aligned} \tag{3.4}$$

The following sections contain model reduction methods for first-order systems from three major methodologies and their existing extensions to second-order systems. All of these methods will use the projection frameworks (3.2) and (3.4).

3.2 Modal truncation and dominant poles

The modal truncation approach is one of the oldest ideas for model order reduction and based on the eigenvalues of the system matrices. It was first mentioned in [75] for the approximation of standard first-order LTI systems ($\mathsf{E} = I_n$). The following sections will give a short overview about the idea of the modal truncation method and an important extension considering the poles of the underlying system's transfer function.

3.2.1 Modal truncation method

In contrast to the original reference [75], consider here the case of generalized first-order systems (2.8). The classical modal truncation method from [75] belongs to the projection-based model reduction approaches. Thereby, the crucial point is the construction of the reduction bases V and W. In modal truncation, these matrices are chosen as parts of the eigenvector bases of the matrix pencil $\lambda E - A$. For simplicity, it is assumed that $\lambda\mathsf{E} - \mathsf{A}$ is diagonalizable. Let $0 \neq \mathsf{x}_i \in \mathbb{C}^n$ and $0 \neq \mathsf{y}_i \in \mathbb{C}^n$ be right and left eigenvectors of $\lambda\mathsf{E} - \mathsf{A}$ for the same eigenvalue $\lambda_i \in \mathbb{C}$, respectively, i.e., it holds

$$\mathsf{A}\mathsf{x}_i = \lambda_i\mathsf{E}\mathsf{x}_i \quad \text{and} \quad \mathsf{y}_i^{\mathsf{H}}\mathsf{A} = \lambda_i\mathsf{y}_i^{\mathsf{H}}\mathsf{E}. \tag{3.5}$$

Assuming also the scaling $\mathsf{y}_i^{\mathsf{H}}\mathsf{E}\mathsf{x}_i = 1$, the full eigenvector bases can be used for a state-space transformation (2.11) of (2.8), which yields

$$\begin{aligned}\dot{\tilde{\mathsf{x}}}(t) &= \begin{bmatrix}\lambda_1 & & \\ & \ddots & \\ & & \lambda_{n_1}\end{bmatrix}\tilde{\mathsf{x}}(t) + \begin{bmatrix}\tilde{\mathsf{b}}_1^{\mathsf{H}} \\ \vdots \\ \tilde{\mathsf{b}}_{n_1}^{\mathsf{H}}\end{bmatrix}u(t),\\ \tilde{y}(t) &= \begin{bmatrix}\tilde{\mathsf{c}}_1 & \dots & \tilde{\mathsf{c}}_{n_1}\end{bmatrix}\tilde{\mathsf{x}}(t).\end{aligned} \tag{3.6}$$

The input and output matrices have been transformed and partitioned according to the diagonal structure of the system matrix with $\tilde{\mathsf{b}}_1, \dots, \tilde{\mathsf{b}}_{n_1} \in \mathbb{C}^m$ and $\tilde{\mathsf{c}}_1, \dots, \tilde{\mathsf{c}}_{n_1} \in \mathbb{C}^p$. Due to the diagonal structure, the transformed system (3.6) decouples into n_1 independent subsystems, from which r_1 are chosen to build the reduced-order model. In other words, eigenvalues $\lambda_1, \dots, \lambda_{r_1}$ from the original matrix pencil are chosen (with appropriate re-ordering of the indices) to remain in the reduced-order model such that the truncation matrices are set to be the corresponding eigenvectors with $W = \begin{bmatrix}\mathsf{y}_1 & \dots & \mathsf{y}_{r_1}\end{bmatrix}$ and $V = \begin{bmatrix}\mathsf{x}_1 & \dots & \mathsf{x}_{r_1}\end{bmatrix}$. Other variants of the modal truncation method utilize, for example, bases of invariant subspaces corresponding to the chosen eigenvalues; see, e.g., [50].

For second-order systems (2.17), there have been a lot of different attempts for the extension of the modal truncation method. Overviews about developed techniques can be found in [60, 125]. Methods like static condensation (Guyan reduction) [105] or the Craig-Bampton method [73] belong to the class of modal truncation approaches by making use of the known structure of models resulting, e.g., from finite element methods. These methods are in need of a certain engineering expertise during the model reduction process and, therefore, not suited for automatic reduction in the sense that a common user applies the methods directly to the data. Both approaches and related techniques are not further discussed in this thesis.

In general, modal truncation methods for second-order systems can be related to the underlying quadratic eigenvalue problem of (2.17) (or parts of it). In that sense, the linear eigenvalue problems in (3.5) from the first-order case are replaced by the quadratic eigenvalue problems

$$(\lambda_i^2 M + \lambda_i E + K)x_i = 0 \quad \text{and} \quad y_i^{\mathsf{H}}(\lambda_i^2 M + \lambda_i E + K) = 0, \tag{3.7}$$

which need to be solved as before for left and right eigenvectors, and the corresponding eigenvalues. This shows the main advantage of modal truncation approaches as they can easily be generalized to other internal system structures by adapting the corresponding eigenvalue problem, and they are also very compatible in a computational sense since only eigenvalue problems have to be solved.

Some changes to the first-order case need to be noted. While (3.5) provides exactly n_1 eigenvalues, the quadratic eigenvalue problem (3.7) has $2n_2$ eigenvalues. Also, the

quadratic matrix pencil $\lambda^2 M + \lambda E + K$ is generically not diagonalizable, which means, out of the $2n_2$ eigenvalues only r_2 can be chosen to remain guaranteed in the reduced-order model. The missing r_2 eigenvalues can in principle be unrelated approximations, resulting from the truncation of the matrix pencil. As discussed in, e.g., [134], or used in [174], often the problem (3.7) is simplified by neglecting the damping term E for the computation of the model reduction basis. Assuming M and K to be symmetric positive definite, only the generalized eigenvalue problem

$$Kv_i = \omega_i^2 M v_i \tag{3.8}$$

is solved for the stiffness coefficients ω_i. The generalized eigenvectors v_i in (3.8) are then used to set up the model reduction bases $W = V = \begin{bmatrix} v_1 & \dots & v_{r_2} \end{bmatrix}$. This can yield good results and be very advantageous in terms of computational costs in case of special damping matrices E.

3.2.2 Dominant pole algorithms

A crucial problem in modal truncation methods is the choice of eigenvalues to remain in the reduced-order model. Note that the eigenvalues of $\lambda\mathsf{E} - \mathsf{A}$ are the potential poles of the corresponding system's transfer function. With the right and left eigenvector bases X and Y as in (3.6), it holds

$$\mathsf{Y}^{\mathsf{H}}\mathsf{E}\mathsf{X} = I_{n_1} \quad \text{and} \quad \mathsf{Y}^{\mathsf{H}}\mathsf{A}\mathsf{X} = \Lambda,$$

where $\Lambda = \operatorname{diag}(\lambda_1, \dots, \lambda_{n_1})$. Then, the transfer function of (2.8) can be rewritten in its *pole-resiude form*

$$\begin{aligned} \mathsf{G}_{\mathrm{L}}(s) &= \mathsf{C}(s\mathsf{E} - \mathsf{A})^{-1}\mathsf{B} \\ &= \mathsf{C}(s\mathsf{Y}^{-\mathsf{H}}\mathsf{X}^{-1} - \mathsf{Y}^{-\mathsf{H}}\Lambda\mathsf{X}^{-1})^{-1}\mathsf{B} \\ &= \mathsf{C}\mathsf{X}(sI_{n_1} - \Lambda)^{-1}\mathsf{Y}^{\mathsf{H}}\mathsf{B} \\ &= \sum_{k=1}^{n_1} \frac{(\mathsf{C}\mathsf{x}_k)(\mathsf{y}_k^{\mathsf{H}}\mathsf{B})}{s - \lambda_k}. \end{aligned} \tag{3.9}$$

The problem is now to identify those poles of (3.9) that contribute most to the transfer function's behavior. This leads to the following definition.

Definition 3.1 (Dominant poles [138, 161]):
A pole $\lambda_k \in \mathbb{C}$ of (2.14) is called *dominant* if

$$\frac{\|(\mathsf{C}\mathsf{x}_k)(\mathsf{y}_k^{\mathsf{H}}\mathsf{B})\|_2}{|\mathrm{Re}(\lambda_k)|} > \frac{\|(\mathsf{C}\mathsf{x}_j)(\mathsf{y}_j^{\mathsf{H}}\mathsf{B})\|_2}{|\mathrm{Re}(\lambda_j)|} \tag{3.10}$$

holds, for all $j \neq k$. ◊

The idea of *dominant pole algorithms* is now to compute the reduced-order model with modal truncation by choosing the r_1 most dominant poles with respect to the dominance measure (3.10) such that

$$\widehat{\mathsf{G}}_{\mathrm{L}}(s) = \sum_{k=1}^{r_1} \frac{(\mathsf{C}\mathsf{x}_k)(\mathsf{y}_k^{\mathsf{H}}\mathsf{B})}{s-\lambda_k} \approx \mathsf{G}_{\mathrm{L}}(s),$$

where the poles λ_k are assumed to be ordered with respect to (3.10). The method was originally developed in [138] and then extended to large-scale sparse systems in [161, 162]. Other dominance measures to alter the behavior of the constructed reduced-order models are suggested, for example, in [185] for $\mathcal{H}_2$-like norms or in [182] to match the low frequency behavior.

In case of second-order systems, a structure-preserving extension of the dominant pole algorithm was developed in [163] for SISO systems and in [48] for the MIMO case. In principle, the extensions consider the second-order system (2.17) in first-order form, e.g., (2.18), for which the pole-residue form can be written with $2n_2$ terms, i.e.,

$$\begin{aligned} G_{\mathrm{L}}(s) &= (C_{\mathrm{p}} + sC_{\mathrm{v}})(s^2M + sE + K)^{-1}B_{\mathrm{u}} \\ &= \sum_{k=1}^{2n_2} \frac{(\mathsf{C}\mathsf{x}_k)(\mathsf{y}_k^{\mathsf{H}}\mathsf{B})}{s-\lambda_k}, \end{aligned} \tag{3.11}$$

where C and B are here the output and input matrices of the chosen first-order realization, and x_k and y_k the right and left eigenvectors of the corresponding linearized eigenvalue problem (3.5). The structure-preserving dominant pole algorithm is then used to compute the r_2 most dominant poles in (3.11) with corresponding eigenvectors such that the reduced-order model is given by

$$\widehat{G}_{\mathrm{L}}(s) = \sum_{k=1}^{r_2} \frac{(\mathsf{C}\mathsf{x}_k)(\mathsf{y}_k^{\mathsf{H}}\mathsf{B})}{s-\lambda_k} + \sum_{k=1}^{r_2} \frac{(\mathsf{C}\tilde{\mathsf{x}}_k)(\tilde{\mathsf{y}}_k^{\mathsf{H}}\mathsf{B})}{s-\tilde{\lambda}_k},$$

where $\tilde{\lambda}_k$ are new poles introduced by the truncation of the quadratic eigenvalue problem. In case of complex conjugate poles and the original system being real, some of the $\tilde{\lambda}_k$ are set to be the complex conjugates of the dominant poles λ_k.

3.3 Interpolation and moment matching methods

A different approach is based on the idea of considering the system's transfer function as the object of interest to approximate. Inspired by the observation that in the cases of first- and second-order systems the transfer functions (2.14) and (2.26) are rational functions in the complex variable s, a lot of model reduction techniques and approaches were based on the construction of rational interpolants for the transfer functions. See [10]

for a general introduction to interpolatory model order reduction techniques and related realization methods. In the following sections, the basic ideas for first- and second-order, as well as for even more generally structured systems are recapped.

3.3.1 From moment matching to rational Krylov subspaces

The origins of interpolatory model reduction root in the theory of Padé approximations [18], i.e., the construction of rational approximants. Given a function $G\colon \mathbb{C} \to \mathbb{C}$, which is analytic in 0 and has the power series expansion

$$G(s) = \sum_{j=0}^{\infty} m_j s^j,$$

the coefficients $m_j \in \mathbb{C}$ are called the *moments* of G. The unique, rational function $R(s)$, with

$$R(s) = \frac{a_0 + a_1 s + \ldots + a_{j_1} s^{j_1}}{1 + b_1 s + \ldots b_{j_2} s^{j_2}} = \sum_{j=0}^{\infty} \widehat{m}_j s^j, \tag{3.12}$$

is called a *Padé approximation* of G, if $m_j = \widehat{m}_j$ holds for $j = 0, \ldots, j_1 + j_2$. In other words, a Padé approximation is a Hermite interpolating rational function in 0 of minimum degrees in nominator and denominator. The idea of Padé approximation was then first related to the standard case of SISO first-order systems (2.8) resulting in construction formulae for the coefficients in (3.12); see, e.g., [67, 173]. Based on this, in the last decades, the idea of rational interpolation got extended further and further, e.g., to the interpolation at ∞ in the partial realization problem [9, 96], to the interpolation in other and more frequency points than 0 also known as shifted Padé approximation [100, 184], or to efficient computational approaches in the projection framework (3.2) by Lanczos and Arnoldi methods, e.g., in [14, 87, 100]. Besides Padé approximation, the idea of rational interpolation of transfer functions for model order reduction is referred to as moment matching or Krylov subspace methods in the literature.

The construction of a rational interpolation for (2.8) can be efficiently done in the projection-based framework (3.2) by computing the truncation matrices V and W as bases of *rational Krylov subspaces*. For example, let $\sigma_1, \ldots, \sigma_k \in \mathbb{C}$ be interpolation points in which the transfer functions of the full-order system G_L and of the reduced-order system $\widehat{\mathsf{G}}_\mathrm{L}$, computed by (3.2), exist. Then one can show that if either

$$\operatorname{span}\left(\begin{bmatrix} (\sigma_1 \mathsf{E} - \mathsf{A})^{-1}\mathsf{B} & \ldots & (\sigma_k \mathsf{E} - \mathsf{A})^{-1}\mathsf{B} \end{bmatrix} \right) \subseteq \operatorname{span}(V) \tag{3.13}$$

or

$$\operatorname{span}\left(\begin{bmatrix} (\sigma_1 \mathsf{E} - \mathsf{A})^{-\mathsf{H}}\mathsf{C}^\mathsf{T} & \ldots & (\sigma_k \mathsf{E} - \mathsf{A})^{-\mathsf{H}}\mathsf{C}^\mathsf{T} \end{bmatrix} \right) \subseteq \operatorname{span}(W) \tag{3.14}$$

holds, the interpolation of the full-order transfer function follows

$$\mathsf{G}_{\mathrm{L}}(\sigma_1) = \widehat{\mathsf{G}}_{\mathrm{L}}(\sigma_1), \quad \ldots, \quad \mathsf{G}_{\mathrm{L}}(\sigma_k) = \widehat{\mathsf{G}}_{\mathrm{L}}(\sigma_k).$$

This approach can be extended to match additional Hermite interpolation conditions in an implicit or explicit way. For a more detailed inside of the theory about rational interpolation by projection for first-order systems see, e.g., [100].

As the previous example shows, big advantages of this approach and related methods are the cheap computational costs and the loose assumptions, since only a few shifted linear systems need to be solved and the transfer function must exist (or be complex differentiable in the Hermite interpolation case) in the interpolation points. This makes interpolatory methods a good alternative to other model order reduction techniques with stronger assumptions on the original system. On the other hand, a drawback of this approach is that interpolation by projection lacks stability preservation in many cases, which might lead to undesired results in time domain simulations while the approximation in the frequency domain can still be good due to the interpolation.

A crucial part for the approximation quality of the interpolating reduced-order models is the choice of interpolation points. This question was tried to be answered in different system norms leading in case of the $\mathcal{H}_2$-norm to the *iterative rational Krylov algorithm (IRKA)* [103,104,189] or to greedy approaches in the $\mathcal{H}_\infty$-norm case [6–8,11,80,82].

Transfer function interpolation does not only work in the frequency argument but can also be extended to the parametric system case

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

where $\mu \in \mathbb{R}^d$ is a vector of parameters, constant in time, allowing for different configurations of the system. Then, $\mathsf{E}(\mu)$, $\mathsf{A}(\mu)$, $\mathsf{B}(\mu)$ and $\mathsf{C}(\mu)$ are matrix-valued functions depending on the parameter configuration. In frequency domain, this gives the parametric equivalent to (2.14) with

$$\mathsf{G}_{\mathrm{L}}(s,\mu) = \mathsf{C}(\mu)\Big(s\mathsf{E}(\mu) - \mathsf{A}(\mu)\Big)^{-1}\mathsf{B}(\mu),$$

such that interpolation can be done in the frequency and parameter arguments leading to the additional choice of interpolation points in the parameter domian; see, e.g., [21].

3.3.2 Tangential interpolation for MIMO systems

In case of MIMO systems, the transfer functions are matrix-valued such that the corresponding interpolation problem changes from scalar to matrix interpolation. The interpolation of matrix-valued functions can be interpreted as classical (scalar) interpolation in each entry of the matrix-valued functions, i.e., it yields additional interpolation

conditions for the entries of matrices and results in larger reduced-order models to match these. The tangential interpolation problem instead considers the interpolation of matrix-valued functions along selected directions and can be interpreted as adding constraints to the matrix interpolation problem [19]. For given interpolation points $\sigma_1, \ldots, \sigma_k \in \mathbb{C}$, given function values $y_1, \ldots, y_k \in \mathbb{C}^p$ and right evaluation (tangential) directions $b_1, \ldots, b_k \in \mathbb{C}^m$, the task of *right tangential interpolation* is to find an interpolating function $L\colon \mathbb{C} \to \mathbb{C}^{p\times m}$ such that

$$L(\sigma_j)b_j = y_j^{\mathsf{H}} \tag{3.15}$$

holds for $j = 1, \ldots, k$. The *left and two-sided tangential interpolation problems* are defined in a similar way using left tangential directions.

It was then mentioned in [19] and utilized in [89] to use tangential interpolation for the purpose of model order reduction of linear unstructured MIMO systems. Therefore, the interpolant in (3.15) is restricted to a rational matrix-valued function and the function values are the system's transfer function evaluations into certain directions. The tangential interpolation problems in model reduction are formulated as follows: Given the original system's transfer function (2.14), the goal is to construct a reduced-order model with $\widehat{\mathsf{G}}_{\mathrm{L}}(s) = \widehat{\mathsf{C}}(s\widehat{\mathsf{E}} - \widehat{\mathsf{A}})^{-1}\widehat{\mathsf{B}}$ such that for given interpolation points $\sigma_1, \ldots, \sigma_k \in \mathbb{C}$, right directions $b^{(1)}, \ldots, b^{(k)} \in \mathbb{C}^m$ and left directions $c^{(1)}, \ldots, c^{(k)} \in \mathbb{C}^p$, the following interpolation conditions hold

$$\begin{aligned} \mathsf{G}_{\mathrm{L}}(\sigma_j)b^{(j)} &= \widehat{\mathsf{G}}_{\mathrm{L}}(\sigma_j)b^{(j)}, \\ \left(c^{(j)}\right)^{\mathsf{H}}\mathsf{G}_{\mathrm{L}}(\sigma_j) &= \left(c^{(j)}\right)^{\mathsf{H}}\widehat{\mathsf{G}}_{\mathrm{L}}(\sigma_j), \qquad \text{or} \\ \left(c^{(j)}\right)^{\mathsf{H}}\mathsf{G}_{\mathrm{L}}(\sigma_j)b^{(j)} &= \left(c^{(j)}\right)^{\mathsf{H}}\widehat{\mathsf{G}}_{\mathrm{L}}(\sigma_j)b^{(j)}, \end{aligned} \tag{3.16}$$

for $j = 1, \ldots, k$. It has been proven in various examples that tangential interpolation can be used to construct very accurate and smaller reduced-order models compared to the matrix interpolation approach. Also, it allows for a more dedicated choice of the reduced-order system size independent of the input and output dimensions. The tangential interpolation problems (3.16) were then extended to structure-preserving Hermite interpolation in [24] as discussed later in Section 3.3.4.

3.3.3 Extensions to second-order systems

As for other model reduction approaches, the extension of interpolation-based techniques to the second-order system case (2.17) got a lot of attention. Starting with Padé approximation methods for second-order systems [14, 87, 88], other concepts as rational Krylov subspaces [16], moments of transfer functions [152] and general rational interpolation [23, 169, 170] got extended as well. The same holds for related algorithms such as

choosing interpolation points, e.g., in case of second-order IRKA variants [188] or the modified iterative rational Arnoldi (MIRA) algorithm [62]. In principle, all those results boil down to replace the rational Krylov subspaces for first-order systems (3.13) and (3.14) by second-order variants: Given interpolation points $\sigma_1, \ldots, \sigma_k \in \mathbb{C}$, for which the full-order transfer function (2.26) and the reduced-order transfer function $\widehat{G}_{\mathrm{L}}$, computed by (3.4), exist. Then if either

$$\operatorname{span}\left(\begin{bmatrix} (\sigma_1^2 M + \sigma_1 E + K)^{-1} B_{\mathrm{u}} & \ldots & (\sigma_k^2 M + \sigma_k E + K)^{-1} B_{\mathrm{u}} \end{bmatrix} \right) \subseteq \operatorname{span}(V)$$

or

$$\operatorname{span}\left(\begin{bmatrix} (\sigma_1^2 M + \sigma_1 E + K)^{-\mathsf{H}} (C_{\mathrm{p}} + \sigma_1 C_{\mathrm{v}})^{\mathsf{H}} & \ldots & (\sigma_k^2 M + \sigma_k E + K)^{-\mathsf{H}} (C_{\mathrm{p}} + \sigma_k C_{\mathrm{v}})^{\mathsf{H}} \end{bmatrix} \right) \subseteq \operatorname{span}(W)$$

holds, the full-order system's transfer function is interpolated by a structure-preserving reduced-order model such that

$$G_{\mathrm{L}}(\sigma_1) = \widehat{G}_{\mathrm{L}}(\sigma_1), \quad \ldots, \quad G_{\mathrm{L}}(\sigma_k) = \widehat{G}_{\mathrm{L}}(\sigma_k).$$

More recently, the idea of structured optimality conditions got also extended to second-order systems [22, 144]. For brevity, those results are omitted here. The following section gives a more general framework for the interpolation of structured linear systems, which automatically encloses most of the above mentioned results for systems with first- and second-order structures.

3.3.4 Structured interpolation via rational Krylov subspaces

Consider for a moment the first-order unstructured system case (2.8). With the Laplace transformation, the dynamical system is described via two algebraic systems of equations in the frequency domain (2.13). The first one describes the input-to-state relation and the second one the state-to-output relation of the system. Inspired by much richer structured systems than (2.13), for example, such as (2.25), a general framework for structured systems and transfer functions was introduced in [24]. Therein, the authors consider the two linear systems of equations

$$\begin{aligned} \mathcal{K}(s)\mathcal{X}(s) &= \mathcal{B}(s)U(s), \\ Y(s) &= \mathcal{C}(s)\mathcal{X}(s), \end{aligned} \tag{3.17}$$

with matrix-valued functions $\mathcal{K}\colon \mathbb{C} \to \mathbb{C}^{n\times n}$, $\mathcal{B}\colon \mathbb{C} \to \mathbb{C}^{n\times m}$ and $\mathcal{C}\colon \mathbb{C} \to \mathbb{C}^{p\times n}$, as description of the input-to-state and state-to-output relations of linear dynamical systems in the frequency domain. Note that (3.17) contains (2.13) and (2.25) as particular instances. Assuming the problem to be regular, i.e., there exists an $s \in \mathbb{C}$ for which the

matrix functions are defined and $\mathcal{K}(s)$ is full-rank, the equations in (3.17) lead to the general formulation of *structured transfer functions*

$$\mathcal{G}_{\mathrm{L}}(s) = \mathcal{C}(s)\mathcal{K}(s)^{-1}\mathcal{B}(s), \tag{3.18}$$

describing the input-to-output behavior of a structured linear system in the frequency domain. The goal of *structured interpolation* is to construct an interpolant for (3.18) that has the same internal structure.

Considering the two system classes mentioned so far, the transfer functions of linear first-order systems (2.8) are given in the structured setting by

$$\mathcal{C}(s) = \mathsf{C}, \quad \mathcal{K}(s) = s\mathsf{E} - \mathsf{A}, \quad \mathcal{B}(s) = \mathsf{B},$$

or, in case of second-order systems (2.17), the matrix-valued functions are set to be

$$\mathcal{C}(s) = C_{\mathrm{p}} + sC_{\mathrm{v}}, \quad \mathcal{K}(s) = s^2M + sE + K, \quad \mathcal{B}(s) = B_{\mathrm{u}}.$$

3.3.4.1 Structured-preserving model reduction by projection

For the construction of structured linear reduced-order models, the projection approach as in (3.2) and (3.4) is generalized for systems described by (3.18). Given two full-rank truncation matrices $W, V \in \mathbb{C}^{n\times r}$, reduced-order models of (3.18) are constructed by

$$\widehat{\mathcal{C}}(s) = \mathcal{C}(s)V, \quad \widehat{\mathcal{K}}(s) = W^{\mathsf{H}}\mathcal{K}(s)V, \quad \widehat{\mathcal{B}}(s) = W^{\mathsf{H}}\mathcal{B}(s). \tag{3.19}$$

The structured reduced-order linear system $\widehat{\mathcal{G}}_{\mathrm{L}}$ is then given by the underlying reduced-order matrices from (3.19) and provides the corresponding structured reduced-order transfer function

$$\widehat{\mathcal{G}}_{\mathrm{L}}(s) = \widehat{\mathcal{C}}(s)\widehat{\mathcal{K}}(s)^{-1}\widehat{\mathcal{B}}(s). \tag{3.20}$$

In general, model reduction by projection (3.19) is structure-preserving. Every matrix-valued function can be affinely decomposed with respect to its arguments, e.g., in case of the frequency-dependent term $\mathcal{K}(s)$, it can be written as

$$\mathcal{K}(s) = \sum_{j=1}^{n_{\mathcal{K}}} h_{\mathcal{K},j}(s)\mathcal{K}_j, \tag{3.21}$$

with scalar functions $h_{\mathcal{K},j}\colon \mathbb{C} \to \mathbb{C}$ depending on frequency and constant matrices $\mathcal{K}_j \in \mathbb{C}^{n\times n}$, for $j = 1, \ldots, n_{\mathcal{K}}$. The choice of the scalar functions $h_{\mathcal{K},j}$ in (3.21) encodes the internal structure of the system. In the worst case scenario, the number of terms in (3.21) would be $n_{\mathcal{K}} = n^2$, where $\mathcal{K}_j$ are elementary matrices with only a single non-zero entry in each matrix. However, for common structured examples the number of terms

in (3.21) is comparably small with $n_{\mathcal{K}} \ll n$. Otherwise, there are other approaches like the discrete empirical interpolation method (DEIM) to approximate the matrix-valued functions [34]. Using (3.21), the corresponding reduced-order matrix function is then given by

$$\widehat{\mathcal{K}}(s) = W^{\mathsf{H}}\mathcal{K}(s)V = \sum_{j=1}^{n_{\mathcal{K}}} h_{\mathcal{K},j}(s)W^{\mathsf{H}}\mathcal{K}_j V = \sum_{j=1}^{n_{\mathcal{K}}} h_{\mathcal{K},j}(s)\widehat{\mathcal{K}}_j, \tag{3.22}$$

where $\widehat{\mathcal{K}}_j \in \mathbb{C}^{r\times r}$ are small constant matrices. Since the scalar functions $h_{\mathcal{K},j}(s)$, which encode the structure of the system, do not change between (3.21) and (3.22), the internal structure of the matrix function and consequently the system structure is preserved in the reduced-order model. This works analogously for the other matrix-valued functions in (3.19). For first- and second-order systems, this directly resembles the previously used projection approaches (3.2) and (3.4).

3.3.4.2 Structured interpolation

The goal in structured interpolation is now to construct the truncation matrices V and W in (3.19) such that

$$\mathcal{G}_{\mathrm{L}}(\sigma_j) = \widehat{\mathcal{G}}_{\mathrm{L}}(\sigma_j) \tag{3.23}$$

holds, for $j = 1, \ldots, k$, and given interpolation points $\sigma_1, \ldots, \sigma_k \in \mathbb{C}$. The following proposition gives conditions on the projection spaces $\operatorname{span}(V)$ and $\operatorname{span}(W)$ associated with the truncation matrices to satisfy not only (3.23) but also Hermite interpolation conditions.

Proposition 3.2 (Structured linear interpolation [24]):
Let $\mathcal{G}_{\mathrm{L}}$ be a linear system, described by (3.18), and $\widehat{\mathcal{G}}_{\mathrm{L}}$ the reduced-order linear system described by (3.20) and constructed by projection (3.19). Let the matrix functions $\mathcal{C}$, $\mathcal{K}^{-1}$, $\mathcal{B}$ and $\widehat{\mathcal{K}}^{-1}$ be complex differentiable in the point $\sigma \in \mathbb{C}$, and let $k, \theta \in \mathbb{N}_0$ be derivative orders.

(a) If $\operatorname{span}\left(\partial_{s^j}(\mathcal{K}^{-1}\mathcal{B})(\sigma)\right) \subseteq \operatorname{span}(V)$, for $j = 0, \ldots, k$, then

$$\partial_{s^j}\mathcal{G}_{\mathrm{L}}(\sigma) = \partial_{s^j}\widehat{\mathcal{G}}_{\mathrm{L}}(\sigma)$$

holds for $j = 0, \ldots, k$.

(b) If $\operatorname{span}\left(\partial_{s^j}(\mathcal{K}^{-\mathsf{H}}\mathcal{C}^{\mathsf{H}})(\sigma)\right) \subseteq \operatorname{span}(W)$, for $i = 0, \ldots, \theta$, then

$$\partial_{s^i}\mathcal{G}_{\mathrm{L}}(\sigma) = \partial_{s^i}\widehat{\mathcal{G}}_{\mathrm{L}}(\sigma)$$

holds for $i = 0, \ldots, \theta$.

(c) If V and W are constructed as in Parts (a) and (b), then, additionally, it holds

$$\partial_{s^j} \mathcal{G}_{\mathrm{L}}(\sigma) = \partial_{s^j} \widehat{\mathcal{G}}_{\mathrm{L}}(\sigma),$$

for $j = 0, \ldots, k + \theta + 1$. ◊

The original version of Proposition 3.2 was directly formulated for the case of tangential interpolation, i.e., with the multiplication of $\mathcal{B}$ with an input direction $b \in \mathbb{C}^m$ and of $\mathcal{C}$ with an output direction $c \in \mathbb{C}^p$. The matrix interpolation results follow from [24, Theorem 1] by concatenating the resulting projection spaces such that $b = I_m$ and $c = I_p$ are used. The underlying idea in the proof of Proposition 3.2 is the construction of appropriate projectors onto the underlying projection spaces $\operatorname{span}(V)$ and $\operatorname{span}(W)$ by using the basis matrices V and W, and parts of the transfer function. For the notation in upcoming proofs let

$$P_{\mathrm{V}}(s) := V(W^{\mathsf{H}}\mathcal{K}(s)V)^{-1}W^{\mathsf{H}}\mathcal{K}(s) \qquad \text{and} \tag{3.24}$$

$$P_{\mathrm{W}}(s) := W(W^{\mathsf{H}}\mathcal{K}(s)V)^{-\mathsf{H}}V^{\mathsf{H}}\mathcal{K}(s)^{\mathsf{H}}, \tag{3.25}$$

with $s \in \mathbb{C}$, denote special frequency-dependent projectors onto $\operatorname{span}(V)$ and $\operatorname{span}(W)$, respectively. In consequence, given vectors $v \in \operatorname{span}(V)$ and $w \in \operatorname{span}(W)$, it holds

$$v = P_{\mathrm{V}}(s)v \quad \text{and} \quad w = P_{\mathrm{W}}(s)w, \tag{3.26}$$

for all $s \in \mathbb{C}$ for which P_{V} and P_{W} exist.

As for the first- and second-order system cases, the choice of interpolation points is crucial for the quality of the computed approximation. An idea that was developed lately is related to the computation of the $\mathcal{H}_\infty$-norm via structure-preserving interpolation [6–8, 172] leading to an $\mathcal{H}_\infty$-norm minimizing selection of interpolation points for model order reduction [26, 27]. Alternatively, instead of approximating the exact $\mathcal{H}_\infty$-error, estimators can be used for the same purpose [82]. There is no extension of the IRKA method compliant with the general structured system case (3.18), except for some special cases [22, 144, 155–157, 188]. Nevertheless, the idea of constructing $\mathcal{H}_2$-optimal reduced-order models was extended to general transfer functions in [25] with the *transfer function iterative rational Krylov algorithm (TF-IRKA)*. The method uses the *Loewner framework* from [140] to construct an unstructured interpolating first-order system (2.8) from frequency data and iterates this in an IRKA-like algorithm. It has been shown to be very efficient in practice to use TF-IRKA for (3.18) to obtain good interpolation points, which then can be used for structured interpolation via Proposition 3.2.

Remark 3.3 (Averaging subspaces):
A common drawback of interpolation methods is their error behavior. While being exact in the interpolation points (and possible derivatives), the error away from these

points can increase a lot depending on the actual transfer function behavior. A quite often used approach to counter that was lately reformulated for the computation of minimal realizations of linear structured parametric systems via dominant subspaces [41]. The general idea is to solve the linear systems in Proposition 3.2 for a large amount of interpolation points. Then, a rank-revealing orthogonalization method, like pivoted QR or the singular value decomposition (SVD), is used to obtain orthogonal basis matrices with appropriate ordering of the basis contributions. Finally, these bases are truncated to the desired reduced order. In principle, this method approximates the full projection spaces corresponding to the interpolation conditions by lower-order ones and tries to fetch the most important features. Therefore, the approximation results depend on the chosen rank-revealing orthogonalization method and will likely not satisfy any interpolation conditions anymore. ◊

3.4 Balanced truncation approaches

The introduction to balanced truncation presented here is mainly taken from [57]. Balanced truncation is a projection-based model reduction approach for first-order systems (2.8) using energy considerations to identify parts of the state only contributing marginally to the input-to-output behavior of the system. Originally it was developed in [147] for the standard system case, with $\mathsf{E} = I_{n_1}$. The extension of the balanced truncation method to descriptor systems (E non-invertible) was done in [179]. Assuming (2.8) to be asymptotically stable and E to be invertible, the system Gramians of (2.8) are defined by

$$\begin{aligned} \mathsf{P}_\infty &:= \frac{1}{2\pi}\int_{-\infty}^{+\infty}(\omega\mathrm{i}\mathsf{E}-\mathsf{A})^{-1}\mathsf{B}\mathsf{B}^\mathsf{T}(-\omega\mathrm{i}\mathsf{E}-\mathsf{A})^{-\mathsf{T}}\mathrm{d}\omega = \int_0^{+\infty} e^{\mathsf{E}^{-1}\mathsf{A}t}\mathsf{E}^{-1}\mathsf{B}\mathsf{B}^\mathsf{T}\mathsf{E}^{-\mathsf{T}}e^{\mathsf{A}^\mathsf{T}\mathsf{E}^{-\mathsf{T}}t}\mathrm{d}t, \\ \mathsf{E}^\mathsf{T}\mathsf{Q}_\infty\mathsf{E} &:= \frac{1}{2\pi}\int_{-\infty}^{+\infty}\mathsf{E}^\mathsf{T}(-\omega\mathrm{i}\mathsf{E}-\mathsf{A})^{-\mathsf{T}}\mathsf{C}^\mathsf{T}\mathsf{C}(\omega\mathrm{i}\mathsf{E}-\mathsf{A})^{-1}\mathsf{E}\mathrm{d}\omega = \int_0^{+\infty} e^{\mathsf{A}^\mathsf{T}\mathsf{E}^{-\mathsf{T}}t}\mathsf{C}^\mathsf{T}\mathsf{C}e^{\mathsf{E}^{-1}\mathsf{A}t}\mathrm{d}t, \end{aligned} \tag{3.27}$$

with P_∞, the *infinite controllability Gramian*, and $\mathsf{E}^\mathsf{T}\mathsf{Q}_\infty\mathsf{E}$, the *infinite observability Gramian*. Due to the integration up to infinity, the Gramians are equivalently defined in frequency and time domain. It can be shown that the two matrices P_∞ and Q_∞ from (3.27) are also given as the unique, symmetric positive semi-definite solutions of the two dual Lyapunov equations

$$\begin{aligned} \mathsf{A}\mathsf{P}_\infty\mathsf{E}^\mathsf{T} + \mathsf{E}\mathsf{P}_\infty\mathsf{A}^\mathsf{T} + \mathsf{B}\mathsf{B}^\mathsf{T} &= 0, \\ \mathsf{A}^\mathsf{T}\mathsf{Q}_\infty\mathsf{E} + \mathsf{E}^\mathsf{T}\mathsf{Q}_\infty\mathsf{A} + \mathsf{C}^\mathsf{T}\mathsf{C} &= 0. \end{aligned} \tag{3.28}$$

Algorithm 3.1: Balanced truncation square-root method.

Input: System matrices E, A, B, C from (2.8).

Output: Matrices of the reduced-order system $\widehat{\mathsf{E}}$, $\widehat{\mathsf{A}}$, $\widehat{\mathsf{B}}$, $\widehat{\mathsf{C}}$.

1 Compute Cholesky factorizations $\mathsf{P}_\infty = \mathsf{R}_\infty \mathsf{R}_\infty^\mathsf{T}$, $\mathsf{Q}_\infty = \mathsf{L}_\infty \mathsf{L}_\infty^\mathsf{T}$ of the solutions of the Lyapunov equations (3.28).

2 Compute the singular value decomposition

$$\mathsf{L}_\infty^\mathsf{T} \mathsf{E} \mathsf{R}_\infty = \begin{bmatrix} \mathsf{U}_1 & \mathsf{U}_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & \\ & \Sigma_1 \end{bmatrix} \begin{bmatrix} \mathsf{T}_1^\mathsf{T} \\ \mathsf{T}_2^\mathsf{T} \end{bmatrix},$$

with $\Sigma_1 = \operatorname{diag}(\varsigma_1, \ldots, \varsigma_{r_1})$ containing the r_1 largest Hankel singular values.

3 Construct the projection matrices

$$V = \mathsf{R}_\infty \mathsf{T}_1 \Sigma_1^{-\frac{1}{2}} \quad \text{and} \quad W = \mathsf{L}_\infty \mathsf{U}_1 \Sigma_1^{-\frac{1}{2}}.$$

4 Compute the reduced-order model by

$$\widehat{\mathsf{E}} = W^\mathsf{T} \mathsf{E} V = I_{r_1}, \quad \widehat{\mathsf{A}} = W^\mathsf{T} \mathsf{A} V, \quad \widehat{\mathsf{B}} = W^\mathsf{T} \mathsf{B}, \quad \widehat{\mathsf{C}} = \mathsf{C} V.$$

A measure for the influence of states to the input-to-output behavior of the system are the *Hankel singular values*. These are defined to be the positive square roots of the eigenvalues of the multiplied system Gramians $\mathsf{P}_\infty \mathsf{E}^\mathsf{T} \mathsf{Q}_\infty \mathsf{E}$. The main idea of balanced truncation is to balance the system such that the Gramians are equal and diagonal

$$\mathsf{P}_\infty = \mathsf{E}^\mathsf{T} \mathsf{Q}_\infty \mathsf{E} = \begin{bmatrix} \varsigma_1 & & & \\ & \varsigma_2 & & \\ & & \ddots & \\ & & & \varsigma_{n_1} \end{bmatrix},$$

with the Hankel singular values $\varsigma_1 \geq \varsigma_2 \geq \ldots \geq \varsigma_{n_1} \geq 0$, and then to truncate states corresponding to small Hankel singular values [147]. The complete balanced truncation method using the square-root balancing formula is summarized in Algorithm 3.1.

The balanced truncation method provides an a-priori error bound in the $\mathcal{H}_\infty$-norm

$$\|\mathsf{G}_\mathrm{L} - \widehat{\mathsf{G}}_\mathrm{L}\|_{\mathcal{H}_\infty} \leq 2 \sum_{k=r_1+1}^{n_1} \varsigma_k, \tag{3.29}$$

where G_L is the transfer function of the original model (2.14) and $\widehat{\mathsf{G}}_\mathrm{L}$ of the reduced-order model computed by Algorithm 3.1. The bound (3.29) depends only on the truncated

Hankel singular values, which allows an adaptive choice of the reduced order with respect to the resulting $\mathcal{H}_\infty$-error. Also, this method preserves the stability of the original model, i.e., since G_L was asymptotically stable also $\widehat{\mathsf{G}}_\mathrm{L}$ is.

The application of the balanced truncation method to large-scale sparse systems is possible by approximating the Cholesky factors of the Gramians via low-rank factors $\mathsf{P}_\infty \approx Z_{\mathsf{R}_\infty} Z_{\mathsf{R}_\infty}^\mathsf{T}$, $\mathsf{E}^\mathsf{T}\mathsf{Q}_\infty\mathsf{E} \approx \mathsf{E}^\mathsf{T} Z_{\mathsf{L}_\infty} Z_{\mathsf{L}_\infty}^\mathsf{T}\mathsf{E}$, with $Z_{\mathsf{R}_\infty} \in \mathbb{R}^{n_1 \times k_{\mathsf{R}_\infty}}$, $Z_{\mathsf{L}_\infty} \in \mathbb{R}^{n_1 \times k_{\mathsf{L}_\infty}}$ and $k_{\mathsf{R}_\infty}, k_{\mathsf{L}_\infty} \ll n_1$; see, e.g., [187]. The approximation of the Gramians is reasonable due to a fast singular value decay that occurs due to the low-rank right-hand sides [17]. For the computation of those factors, appropriate low-rank techniques are well developed [51].

3.4.1 Frequency-limited balanced truncation

Often due to physical limitations, only localized approximations of the system's behavior in time or frequency domain are needed. A suitable method to localize the approximation behavior of the balanced truncation method in the frequency domain is the frequency-limited balanced truncation method [90]. The idea is based on restricting the frequency representation of the system Gramians (3.27) to the requested range of interest on the frequency axis. The *frequency-limited Gramians* of (2.8) are then defined to be

$$\begin{aligned} \mathsf{P}_\Omega &:= \frac{1}{2\pi}\int_\Omega (\omega\mathsf{i}\mathsf{E} - \mathsf{A})^{-1}\mathsf{B}\mathsf{B}^\mathsf{T}(-\omega\mathsf{i}\mathsf{E} - \mathsf{A})^{-\mathsf{T}}\mathrm{d}\omega, \\ \mathsf{E}^\mathsf{T}\mathsf{Q}_\Omega\mathsf{E} &:= \frac{1}{2\pi}\int_\Omega \mathsf{E}^\mathsf{T}(-\omega\mathsf{i}\mathsf{E} - \mathsf{A})^{-\mathsf{T}}\mathsf{C}^\mathsf{T}\mathsf{C}(\omega\mathsf{i}\mathsf{E} - \mathsf{A})^{-1}\mathsf{E}\mathrm{d}\omega, \end{aligned} \tag{3.30}$$

with the frequency range of interest $\Omega = [-\omega_2, -\omega_1] \cup [\omega_1, \omega_2] \subset \mathbb{R}$. It can be shown that the left-hand sides of (3.30) are also given by the unique, symmetric positive semi-definite solutions of the two dual Lyapunov equations

$$\begin{aligned} \mathsf{A}\mathsf{P}_\Omega\mathsf{E}^\mathsf{T} + \mathsf{E}\mathsf{P}_\Omega\mathsf{A}^\mathsf{T} + \mathsf{B}_\Omega\mathsf{B}^\mathsf{T} + \mathsf{B}\mathsf{B}_\Omega^\mathsf{T} &= 0, \\ \mathsf{A}^\mathsf{T}\mathsf{Q}_\Omega\mathsf{E} + \mathsf{E}^\mathsf{T}\mathsf{Q}_\Omega\mathsf{A} + \mathsf{C}_\Omega^\mathsf{T}\mathsf{C} + \mathsf{C}^\mathsf{T}\mathsf{C}_\Omega &= 0. \end{aligned} \tag{3.31}$$

The new right-hand side matrices $\mathsf{B}_\Omega := \mathsf{E}\mathsf{F}_\Omega\mathsf{B}$ and $\mathsf{C}_\Omega := \mathsf{C}\mathsf{F}_\Omega\mathsf{E}$ contain the frequency-dependent matrix function

$$\begin{aligned} \mathsf{F}_\Omega &= \mathrm{Re}\left(\frac{\mathsf{i}}{\pi}\ln\left((\mathsf{A} + \omega_1\mathsf{i}\mathsf{E})^{-1}(\mathsf{A} + \omega_2\mathsf{i}\mathsf{E})\right)\right)\mathsf{E}^{-1} \\ &= \mathsf{E}^{-1}\,\mathrm{Re}\left(\frac{\mathsf{i}}{\pi}\ln\left((\mathsf{A} + \omega_2\mathsf{i}\mathsf{E})(\mathsf{A} + \omega_1\mathsf{i}\mathsf{E})^{-1}\right)\right), \end{aligned} \tag{3.32}$$

with $\ln(.)$ the principal branch of the matrix logarithm. Note that in case of $\omega_1 = 0$, i.e., $\Omega = [-\omega_2, \omega_2]$, the matrix function (3.32) can alternatively be simplified to

$$\mathsf{F}_\Omega = \mathrm{Re}\left(\frac{\mathsf{i}}{\pi}\ln\left(-\mathsf{E}^{-1}\mathsf{A} - \omega_2\mathsf{i}I_{n_1}\right)\right)\mathsf{E}^{-1} = \mathsf{E}^{-1}\,\mathrm{Re}\left(\frac{\mathsf{i}}{\pi}\ln\left(-\mathsf{A}\mathsf{E}^{-1} - \omega_2\mathsf{i}I_{n_1}\right)\right);$$

Algorithm 3.2: Frequency-limited balanced truncation square-root method.

Input: System matrices E, A, B, C from (2.8), frequency range of interest Ω.

Output: Matrices of the reduced-order system $\widehat{\mathsf{E}}$, $\widehat{\mathsf{A}}$, $\widehat{\mathsf{B}}$, $\widehat{\mathsf{C}}$.

1 Compute Cholesky factorizations $\mathsf{P}_\Omega = \mathsf{R}_\Omega \mathsf{R}_\Omega^\mathsf{T}$, $\mathsf{Q}_\Omega = \mathsf{L}_\Omega \mathsf{L}_\Omega^\mathsf{T}$ of the solutions of the frequency-limited Lyapunov equations (3.31).

2 Follow the Steps 2–4 in Algorithm 3.1.

see, e.g., [47]. The frequency-limited Gramians can be extended to an arbitrary number of frequency bands, i.e., for

$$\Omega = \bigcup_{k=1}^{\ell} \Big([-\omega_{2k}, \omega_{2k-1}] \cup [\omega_{2k-1}, \omega_{2k}] \Big),$$

with $0 < \omega_1 < \ldots < \omega_\ell$. In this case, the matrix function (3.32) needs to be modified to

$$\begin{aligned} F_\Omega &= \operatorname{Re}\left(\frac{\mathsf{i}}{\pi} \ln\left(\prod_{k=1}^{\ell} (\mathsf{A} + \omega_{2k-1}\mathsf{i}\mathsf{E})^{-1} (\mathsf{A} + \omega_{2k}\mathsf{i}\mathsf{E}) \right) \right) \mathsf{E}^{-1} \\ &= \mathsf{E}^{-1} \operatorname{Re}\left(\frac{\mathsf{i}}{\pi} \ln\left(\prod_{k=1}^{\ell} (\mathsf{A} + \omega_{2k}\mathsf{i}\mathsf{E}) (\mathsf{A} + \omega_{2k-1}\mathsf{i}\mathsf{E})^{-1} \right) \right). \end{aligned}$$

See [47] for a more detailed discussion of the theory addressed above. The extension of this method to the large-scale system case can also be found in [47] and an extension to descriptor systems in [117]. The resulting frequency-limited balanced truncation method with square-root balancing is summarized in Algorithm 3.2.

3.4.2 Time-limited balanced truncation

The counterpart of the frequency-limited balanced truncation in time domain is the time-limited balanced truncation method [90]. This approach aims for the approximation of the system in a limited time interval $\Theta = [t_0, t_\mathrm{f}]$, where $0 \leq t_0 < t_\mathrm{f}$. Basis is the limitation of the time domain representation of the system Gramians (3.27). The *time-limited Gramians* of (2.8) are then defined to be

$$\begin{aligned} \mathsf{P}_\Theta &:= \int_{t_0}^{t_\mathrm{f}} e^{\mathsf{E}^{-1}\mathsf{A}t} \mathsf{E}^{-1} \mathsf{B}\mathsf{B}^\mathsf{T} \mathsf{E}^{-\mathsf{T}} e^{\mathsf{A}^\mathsf{T}\mathsf{E}^{-\mathsf{T}}t} \mathrm{d}t, \\ \mathsf{E}^\mathsf{T}\mathsf{Q}_\Theta\mathsf{E} &:= \int_{t_0}^{t_\mathrm{f}} e^{\mathsf{A}^\mathsf{T}\mathsf{E}^{-\mathsf{T}}t} \mathsf{C}^\mathsf{T}\mathsf{C} e^{\mathsf{E}^{-1}\mathsf{A}t} \mathrm{d}t. \end{aligned} \tag{3.33}$$

Algorithm 3.3: Time-limited balanced truncation square-root method.

Input: System matrices E, A, B, C from (2.8), time range of interest Θ.

Output: Matrices of the reduced-order system $\widehat{\mathsf{E}}$, $\widehat{\mathsf{A}}$, $\widehat{\mathsf{B}}$, $\widehat{\mathsf{C}}$.

1 Compute Cholesky factorizations $\mathsf{P}_\Theta = \mathsf{R}_\Theta \mathsf{R}_\Theta^\mathsf{T}$, $\mathsf{Q}_\Theta = \mathsf{L}_\Theta \mathsf{L}_\Theta^\mathsf{T}$ of the solutions of the time-limited Lyapunov equations (3.34).

2 Follow the Steps 2–4 in Algorithm 3.1.

It can be shown that the left-hand sides of (3.33) are also given via the unique, positive semi-definite solutions of the two following dual Lyapunov equations

$$\begin{aligned} \mathsf{A}\mathsf{P}_\Theta\mathsf{E}^\mathsf{T} + \mathsf{E}\mathsf{P}_\Theta\mathsf{A}^\mathsf{T} + \mathsf{B}_{t_0}\mathsf{B}_{t_0}^\mathsf{T} - \mathsf{B}_{t_\mathrm{f}}\mathsf{B}_{t_\mathrm{f}}^\mathsf{T} &= 0, \\ \mathsf{A}^\mathsf{T}\mathsf{Q}_\Theta\mathsf{E} + \mathsf{E}^\mathsf{T}\mathsf{Q}_\Theta\mathsf{A} + \mathsf{C}_{t_0}^\mathsf{T}\mathsf{C}_{t_0} - \mathsf{C}_{t_\mathrm{f}}^\mathsf{T}\mathsf{C}_{t_\mathrm{f}} &= 0, \end{aligned} \tag{3.34}$$

where the new right-hand side matrices $\mathsf{B}_{t_{0/\mathrm{f}}} = \mathsf{E}e^{\mathsf{E}^{-1}\mathsf{A}t_{0/\mathrm{f}}}\mathsf{E}^{-1}\mathsf{B} = e^{\mathsf{A}\mathsf{E}^{-1}t_{0/\mathrm{f}}}\mathsf{B}$ and $\mathsf{C}_{t_{0/\mathrm{f}}} = \mathsf{C}e^{\mathsf{E}^{-1}\mathsf{A}t_{0/\mathrm{f}}}$ contain the matrix exponential. In case of $t_0 = 0$, the right-hand sides of (3.34) simplify to $\mathsf{B}_0 = \mathsf{B}$ and $\mathsf{C}_0 = \mathsf{C}$. A more detailed discussion of the time-limited theory, especially for the large-scale sparse system case, can be found in [130]. The extension of the theory to descriptor systems is given in [106]. It can be noted that considering more than one time interval at once $[t_{0,1}, t_{\mathrm{f},1}] \cup \cdots \cup [t_{0,\ell}, t_{\mathrm{f},\ell}]$ is not practical. Usually, one cannot expect a good approximation behavior in the intermediate time intervals since the time simulation strongly depends on the initial values at the beginning of each interval, which might be badly approximated. Instead, it is common to take the smallest and largest time points in the intervals to construct a new overarching time interval $[t_{0,\min}, t_{\mathrm{f},\max}]$, where $t_{0,\min} = \min\{t_{0,1}, \ldots, t_{0,\ell}\}$ and $t_{0,\max} = \max\{t_{\mathrm{f},1}, \ldots, t_{\mathrm{f},\ell}\}$ such that

$$\bigcup_{k=1}^{\ell} [t_{0,k}, t_{\mathrm{f},k}] \subset [t_{0,\min}, t_{\mathrm{f},\max}] = \Theta.$$

Note that with the same argumentation, it is not recommended choosing t_0 different from the actual initial time point of the full time simulation. The resulting time-limited balanced truncation method is summarized in Algorithm 3.3.

3.4.3 Second-order balanced truncation approaches

Over time, there have been many attempts for the generalization of the classical balanced truncation method to second-order systems [69, 143, 159]. The goal was to provide a structure-preserving model reduction technique with the benefits of the balanced truncation method in terms of stability preservation and an a-priori error bound. All of those attempts for a second-order balanced truncation method are based on the same first-order realization of (2.17), namely the first companion form realization (2.18).

Table 3.1: Second-order balanced truncation formulas [57]. The $*$ denotes factors of the SVDs not needed, and thus not accumulated in practical computations. The notation uses (3.36).

Type	SVD(s)	Transformation	Reference
v	$U\Sigma T^{\mathsf{T}} = L_{\infty,\mathrm{v}}^{\mathsf{T}} M R_{\infty,\mathrm{v}}$	$W = L_{\infty,\mathrm{v}} U_1 \Sigma_1^{-\frac{1}{2}},\ V = R_{\infty,\mathrm{v}} T_1 \Sigma_1^{-\frac{1}{2}}$	[159]
fv	$*\,\Sigma T^{\mathsf{T}} = L_{\infty,\mathrm{p}}^{\mathsf{T}} J_{\mathrm{fc}} R_{\infty,\mathrm{p}}$	$W = V,\ V = R_{\infty,\mathrm{p}} T_1 \Sigma_1^{-\frac{1}{2}}$	[143]
vpm	$U\Sigma T^{\mathsf{T}} = L_{\infty,\mathrm{p}}^{\mathsf{T}} J_{\mathrm{fc}} R_{\infty,\mathrm{v}}$	$W = M^{-\mathsf{T}} J_{\mathrm{fc}}^{\mathsf{T}} L_{\infty,\mathrm{p}} U_1 \Sigma_1^{-\frac{1}{2}},\ V = R_{\infty,\mathrm{v}} T_1 \Sigma_1^{-\frac{1}{2}}$	[159]
pm	$U\Sigma T^{\mathsf{T}} = L_{\infty,\mathrm{p}}^{\mathsf{T}} J_{\mathrm{fc}} R_{\infty,\mathrm{p}}$	$W = M^{-\mathsf{T}} J_{\mathrm{fc}}^{\mathsf{T}} L_{\infty,\mathrm{p}} U_1 \Sigma_1^{-\frac{1}{2}},\ V = R_{\infty,\mathrm{p}} T_1 \Sigma_1^{-\frac{1}{2}}$	[159]
pv	$U\Sigma T^{\mathsf{T}} = L_{\infty,\mathrm{v}}^{\mathsf{T}} M R_{\infty,\mathrm{p}}$	$W = L_{\infty,\mathrm{v}} U_1 \Sigma_1^{-\frac{1}{2}},\ V = R_{\infty,\mathrm{p}} T_1 \Sigma_1^{-\frac{1}{2}}$	[159]
vp	$*\Sigma T^{\mathsf{T}} = L_{\infty,\mathrm{p}}^{\mathsf{T}} J_{\mathrm{fc}} R_{\infty,\mathrm{v}}$, $U** = L_{\infty,\mathrm{v}}^{\mathsf{T}} M R_{\infty,\mathrm{p}}$	$W = L_{\infty,\mathrm{v}} U_1 \Sigma_1^{-\frac{1}{2}},\ V = R_{\infty,\mathrm{v}} T_1 \Sigma_1^{-\frac{1}{2}}$	[159]
p	$*\,\Sigma T^{\mathsf{T}} = L_{\infty,\mathrm{p}}^{\mathsf{T}} J_{\mathrm{fc}} R_{\infty,\mathrm{p}}$, $U** = L_{\infty,\mathrm{v}}^{\mathsf{T}} M R_{\infty,\mathrm{v}}$	$W = L_{\infty,\mathrm{v}} U_1 \Sigma_1^{-\frac{1}{2}},\ V = R_{\infty,\mathrm{p}} T_1 \Sigma_1^{-\frac{1}{2}}$	[159]
so	$U_{\mathrm{p}}\Sigma_{\mathrm{p}}T_{\mathrm{p}}^{\mathsf{T}} = L_{\infty,\mathrm{p}}^{\mathsf{T}} J_{\mathrm{fc}} R_{\infty,\mathrm{p}}$, $U_{\mathrm{v}}\Sigma_{\mathrm{v}}T_{\mathrm{v}} = L_{\infty,\mathrm{v}}^{\mathsf{T}} M R_{\infty,\mathrm{v}}$	$W_{\mathrm{p}} = L_{\infty,\mathrm{p}} U_{\mathrm{p},1} \Sigma_{\mathrm{p},1}^{-\frac{1}{2}},\ V_{\mathrm{p}} = R_{\infty,\mathrm{p}} T_{\mathrm{p},1} \Sigma_{\mathrm{p},1}^{-\frac{1}{2}}$, $W_{\mathrm{v}} = L_{\infty,\mathrm{v}} U_{\mathrm{v},1} \Sigma_{\mathrm{v},1}^{-\frac{1}{2}},\ V_{\mathrm{v}} = R_{\infty,\mathrm{v}} T_{\mathrm{v},1} \Sigma_{v,1}^{-\frac{1}{2}}$	[69]

Consider the first-order system Gramians (3.27), alternatively given by (3.28), using the first companion form realization (2.18) for the second-order system (2.17). Then, the Gramians are partitioned according to the block structure in (2.18) such that

$$\mathsf{P}_\infty = \begin{bmatrix} P_{\infty,\mathrm{p}} & P_{\infty,12} \\ P_{\infty,12}^{\mathsf{T}} & P_{\infty,\mathrm{v}} \end{bmatrix} \quad \text{and} \quad \mathsf{E}^{\mathsf{T}}\mathsf{Q}_\infty\mathsf{E} = \begin{bmatrix} J_{\mathrm{fc}}^{\mathsf{T}} Q_{\infty,\mathrm{p}} J_{\mathrm{fc}} & J_{\mathrm{fc}}^{\mathsf{T}} Q_{\infty,12} M \\ M^{\mathsf{T}} Q_{\infty,12}^{\mathsf{T}} J_{\mathrm{fc}} & M^{\mathsf{T}} Q_{\infty,\mathrm{v}} M \end{bmatrix}, \tag{3.35}$$

where $P_{\infty,\mathrm{p}}$, $J_{\mathrm{fc}}^{\mathsf{T}} Q_{\infty,\mathrm{p}} J_{\mathrm{fc}}$ are the so-called *infinite position Gramians* of (2.17) and $P_{\infty,\mathrm{v}}$, $M^{\mathsf{T}} Q_{\infty,\mathrm{v}} M$ the *infinite velocity Gramians*. Due to P_∞ and Q_∞ being symmetric positive semi-definite, also the matrices defining the position and velocity Gramians are symmetric positive semi-definite and can be written in terms of their Cholesky factorizations

$$P_{\infty,\mathrm{p}} = R_{\infty,\mathrm{p}} R_{\infty,\mathrm{p}}^{\mathsf{T}},\quad P_{\infty,\mathrm{v}} = R_{\infty,\mathrm{v}} R_{\infty,\mathrm{v}}^{\mathsf{T}},\quad Q_{\infty,\mathrm{p}} = L_{\infty,\mathrm{p}} L_{\infty,\mathrm{p}}^{\mathsf{T}},\quad Q_{\infty,\mathrm{v}} = L_{\infty,\mathrm{v}} L_{\infty,\mathrm{v}}^{\mathsf{T}}. \tag{3.36}$$

Algorithm 3.4: Second-order balanced truncation square-root method.

Input: System matrices M, E, K, B_u, C_p, C_v from (2.17).

Output: Matrices of the reduced-order system $\widehat{M}$, $\widehat{E}$, $\widehat{K}$, $\widehat{B}_\mathrm{u}$ $\widehat{C}_\mathrm{p}$, $\widehat{C}_\mathrm{v}$.

1 Compute Cholesky factorizations $\mathsf{P}_\infty = \mathsf{R}_\infty \mathsf{R}_\infty^\mathsf{T}$, $\mathsf{Q}_\infty = \mathsf{L}_\infty \mathsf{L}_\infty^\mathsf{T}$ of the solutions of the first-order Lyapunov equations (3.28), where the realization (2.18) is used.

2 Partition the Cholesky factors according to the first-order formulation

$$\mathsf{R}_\infty = \begin{bmatrix} R_{\infty,\mathrm{p}} \\ R_{\infty,\mathrm{v}} \end{bmatrix} \quad \text{and} \quad \mathsf{L}_\infty = \begin{bmatrix} L_{\infty,\mathrm{p}} \\ L_{\infty,\mathrm{v}} \end{bmatrix}.$$

3 Compute the SVDs and transformation matrices as in Table 3.1.

4 Compute the reduced-order model by either (3.4) for the methods *p*, *pm*, *pv*, *vp*, *vpm*, *v* and *fv*, or by (3.37) for *so*.

Based on these, the different second-order balanced truncation methods are defined by balancing certain combinations of the four second-order Gramians. For most of the approaches, the resulting balanced truncation is computed as second-order projection method (3.4), where the different choices for W and V can be found in Table 3.1. Therein, the different transformation formulas are summarized and denoted by their type as used in the corresponding references. The subscript 1 matrices denote the part of the SVDs corresponding to the r_2 largest singular values.

In contrast to the balancing methods that describe the reduced-order model by (3.4), the second-order balanced truncation (*so*) from [69] computes the reduced-order model by

$$\begin{aligned} \widehat{M} &= S\left(W_\mathrm{v}^\mathsf{T} M V_\mathrm{v}\right) S^{-1}, & \widehat{E} &= S\left(W_\mathrm{v}^\mathsf{T} E V_\mathrm{v}\right) S^{-1}, & \widehat{K} &= S\left(W_\mathrm{v}^\mathsf{T} K V_\mathrm{p}\right), \\ \widehat{B}_\mathrm{u} &= S\left(W_\mathrm{v}^\mathsf{T} B_\mathrm{u}\right), & \widehat{C}_\mathrm{p} &= C_\mathrm{p} V_\mathrm{p}, & \widehat{C}_\mathrm{v} &= C_\mathrm{v} V_\mathrm{v} S^{-1}, \end{aligned} \tag{3.37}$$

where $S = W_\mathrm{p} J_\mathrm{fc} V_\mathrm{v}$ and the transformation matrices W_p, W_v, V_p, V_v are given in the last row of Table 3.1. This type of balancing can be seen as a projection method for the first-order realization (2.18) with a recovering of the second-order structure afterwards.

The general second-order balanced truncation square-root method is summarized in Algorithm 3.4.

Remark 3.4 (Second-order vs. classical balanced truncation methods):
In contrast to the first-order balanced truncation described in Section 3.4, none of the second-order balanced truncation methods provides an error bound in the $\mathcal{H}_\infty$-norm or can guarantee stability preservation in the general case. A collection of examples for the stability issue is given in [159]. In case of mechanical systems with M, E, K symmetric positive definite and $C_\mathrm{v} = 0$, it can be shown that the position-velocity

balancing (pv) as well as the free-velocity balancing (fv) are both stability preserving. Note that the position-velocity balancing also belongs to the class of balanced truncation approaches, which define the system Gramians as integral in the frequency domain using relations of the underlying transfer function (2.26). These balancing approaches have been generalized in [64] for systems with integro-differential equations. ◊

Recently, a new approach for the model reduction of passive second-order systems was suggested in [76]. This method is based on the positive-real balanced truncation and makes use of structure recovery rather than structure preservation, since first an unstructured reduced-order model is computed and then modified into the second-order form. This approach will not be further considered in this thesis.

CHAPTER 4

LINEAR MECHANICAL SYSTEMS

Contents

This chapter is concerned with newly developed model order reduction techniques for linear second-order systems (2.17). First, an extension of the idea of dominant pole algorithms (Section 3.2.2) is presented in Section 4.1 for an important subclass of linear mechanical systems. Afterwards in Section 4.2, the limited balanced truncation

methods (Sections 3.4.1 and 3.4.2) are extended in a structure-preserving fashion to the second-order system case.

4.1 Second-order modally damped dominant pole algorithm

An important problem in the work with mechanical systems is the modeling of the damping term. A common choice for the internal damping behavior of the system is to use combinations of the stiffness and mass matrices. This results in modally damped mechanical systems; see, e.g., [148, 183]. In the following, the structured pole-residue form of modally damped systems is developed to get a new dominance measure and a structure-preserving dominant pole algorithm for modally damped mechanical systems. Also, error bounds in the $\mathcal{H}_\infty$-norm are proposed as well as a structure-preserving approach to improve the approximation quality. The resulting algorithms are then tested using two benchmark examples. The general ideas presented here and Algorithm 4.1 are published in [27, 168].

4.1.1 Structured pole-residue form

Modally damped mechanical systems are a special subclass of mechanical second-order systems (2.17) of the form

$$\begin{aligned} M\ddot{x}(t) + E\dot{x}(t) + Kx(t) &= B_{\mathrm{u}}u(t), \\ y(t) &= C_{\mathrm{p}}x(t), \end{aligned} \tag{4.1}$$

with $M, E, K \in \mathbb{R}^{n_2 \times n_2}$ symmetric positive definite, $B_{\mathrm{u}} \in \mathbb{R}^{n_2 \times m}$ and $C_{\mathrm{p}} \in \mathbb{R}^{p \times n_2}$, where the damping and stiffness matrices commutate with respect to the inverse mass matrix, i.e., it holds

$$EM^{-1}K = KM^{-1}E. \tag{4.2}$$

The transfer function of (4.1) is given by

$$G_{\mathrm{L}}(s) = C_{\mathrm{p}}(s^2M + sE + K)^{-1}B_{\mathrm{u}}. \tag{4.3}$$

The modal damping approach is a common method to model the internal damping behavior of mechanical systems due to its convenient properties and wide understanding. Often applied special cases are, for example, Rayleigh (or proportional) damping with

$$E_{\mathrm{ray}} = \alpha M + \beta K, \tag{4.4}$$

where $\alpha, \beta \in \mathbb{R}_{\geq 0}$, as used in the introductory examples (Sections 1.3.1 and 1.3.2), or a scaled version of critical damping

$$E_{\text{crit}} = 2\delta M^{\frac{1}{2}} \sqrt{M^{-\frac{1}{2}} K M^{-\frac{1}{2}}} M^{\frac{1}{2}},$$

with $\delta \in \mathbb{R}_{>0}$; see, e.g., [52, 53, 181].

While in general the pole-residue formulation of second-order systems (3.11) can be obtained using a first-order realization, modally damped systems yield a specially structured pole-residue form. This structured pole-residue form of (4.3) was used in [22] to derive $\mathcal{H}_2$-optimality conditions for modally damped second-order systems. Here, it will be the basis for a dominant pole algorithm (cf. Section 3.2.2). To get the structured pole-residue form of (4.3), consider first the generalized eigenvalue problem

$$Kx_k = \omega_k^2 M x_k,$$

for the eigenvalues ω_k^2, with $\omega_k \in \mathbb{R}_{>0}$, and eigenvectors $0 \neq x_k \in \mathbb{R}^{n_2}$. Due to the symmetry of M and K, the left and right eigenvectors are identical, and both matrices are simultaneously diagonalizable. Collecting all eigenvalues and eigenvectors into matrices yields

$$KX = MX\Omega^2,$$

where $\Omega = \operatorname{diag}(\omega_1, \ldots, \omega_{n_2})$ and $X = \begin{bmatrix} x_1 & \ldots & x_{n_2} \end{bmatrix}$. By appropriately scaling the eigenvector basis, one gets

$$X^{\mathsf{T}} M X = \Omega^{-1} \quad \text{and} \quad X^{\mathsf{T}} K X = \Omega. \tag{4.5}$$

With the modal damping assumption (4.2), the damping term can be diagonalized using the same eigenvector basis, i.e.,

$$X^{\mathsf{T}} E X = 2\Xi, \tag{4.6}$$

with $\Xi = \operatorname{diag}(\xi_1, \ldots, \xi_{n_2})$, the damping ratios. Then using (4.5) and (4.6), the structured pole-residue form of (4.3) is given by

$$\begin{aligned} G_{\text{L}}(s) &= C_{\text{p}}(s^2 M + sE + K)^{-1} B_{\text{u}} \\ &= C_{\text{p}}(s^2 X^{-\mathsf{T}} \Omega^{-1} X^{-1} + 2sX^{-\mathsf{T}} \Xi X^{-1} + X^{-\mathsf{T}} \Omega X^{-1})^{-1} B_{\text{u}} \\ &= C_{\text{p}} X (s^2 \Omega^{-1} + 2s\Xi + \Omega)^{-1} X^{\mathsf{T}} B_{\text{u}} \\ &= \sum_{k=1}^{n_2} \frac{\omega_k (C_{\text{p}} x_k)(x_k^{\mathsf{T}} B_{\text{u}})}{(s - \lambda_k^+)(s - \lambda_k^-)}, \end{aligned} \tag{4.7}$$

where the eigenvalues of the underlying quadratic eigenvalue problem (3.7) (the potential poles of (4.3)) can be determined as pairwise solutions of quadratic equations using

$$\lambda_k^{\pm} = -\omega_k \xi_k \pm \omega_k \sqrt{\xi_k^2 - 1}, \tag{4.8}$$

for $k = 1, \ldots, n_2$.

The most important difference between the structured pole-residue form (4.7) and the unstructured variant (3.11) is the number of summed terms. While the unstructured version for (4.1) has $2n_2$ terms corresponding to the single poles and residues, the structured version has only n_2 terms due to the pairwise appearing poles (3.7) corresponding to single residues each.

Next, the idea of dominant poles (3.10) needs to be extended to the structured pole-residue form (4.7). A first extension idea was used in [168]. The approach therein considered

$$\frac{\|\omega_k (C_\mathrm{p} x_k)(x_k^\mathsf{T} B_\mathrm{u})\|_2}{\operatorname{Re}(\lambda_k^+) \operatorname{Re}(\lambda_k^-)}$$

as measure for dominance. This can be seen as an easy straight-forward extension of the dominance measure from the first-order system case (3.10), as it considers the distance of the poles to the imaginary axis individually. Looking back to the origins of dominant pole algorithms [138], the idea of the dominance measure is to identify those pole-residue terms in the sum (3.9), which have potentially the biggest influence on the transfer function behavior in an $\mathcal{H}_\infty$ sense. Considering a single term of the structured pole-residue form (4.7) in the $\mathcal{H}_\infty$-norm shows

$$\begin{aligned}
\left\| \frac{\omega_k (C_\mathrm{p} x_k)(x_k^\mathsf{T} B_\mathrm{u})}{(s - \lambda_k^+)(s - \lambda_k^-)} \right\|_{\mathcal{H}_\infty} &= \sup_{f \in \mathbb{R}} \left\| \frac{\omega_k (C_\mathrm{p} x_k)(x_k^\mathsf{T} B_\mathrm{u})}{(f\mathrm{i} - \lambda_k^+)(f\mathrm{i} - \lambda_k^-)} \right\|_2 \\
&= \|\omega_k (C_\mathrm{p} x_k)(x_k^\mathsf{T} B_\mathrm{u})\|_2 \left(\max_{f \in \mathbb{R}} \frac{1}{|(f\mathrm{i} - \lambda_k^+)(f\mathrm{i} - \lambda_k^-)|} \right) \\
&= \|\omega_k (C_\mathrm{p} x_k)(x_k^\mathsf{T} B_\mathrm{u})\|_2 \left(\frac{1}{\min\limits_{f \in \mathbb{R}} |(f\mathrm{i} - \lambda_k^+)(f\mathrm{i} - \lambda_k^-)|} \right).
\end{aligned}$$

For the remaining minimum in the denominator, one has to remember that the modally damped system (4.1) was considered to be real, i.e., its poles can only occur in either real or complex conjugate pairs. With that in mind, it is easy to show that

$$\arg\min_{f \in \mathbb{R}} |(f\mathrm{i} - \lambda_k^+)(f\mathrm{i} - \lambda_k^-)| = \pm \operatorname{Im}(\lambda_k^+) = \mp \operatorname{Im}(\lambda_k^-), \tag{4.9}$$

holds, i.e., the $\mathcal{H}_\infty$-norm of a single pole-residue term is given by

$$\left\| \frac{\omega_k (C_\mathrm{p} x_k)(x_k^\mathsf{T} B_\mathrm{u})}{(s - \lambda_k^+)(s - \lambda_k^-)} \right\|_{\mathcal{H}_\infty} = \frac{\|\omega_k (C_\mathrm{p} x_k)(x_k^\mathsf{T} B_\mathrm{u})\|_2}{|\operatorname{Re}(\lambda_k^+)(\operatorname{Im}(\lambda_k^+)\mathrm{i} - \lambda_k^-)|} = \frac{\|\omega_k (C_\mathrm{p} x_k)(x_k^\mathsf{T} B_\mathrm{u})\|_2}{|(\operatorname{Im}(\lambda_k^-)\mathrm{i} - \lambda_k^+) \operatorname{Re}(\lambda_k^-)|}.$$

This leads to the following definition of dominant pole pairs.

Definition 4.1 (Modally damped dominant pole pairs):
A pole pair $(\lambda_k^+, \lambda_k^-)$ of the modally damped second-order system (4.1) is called *dominant* if, with the corresponding eigenvectors $0 \neq x_k, x_j \in \mathbb{R}^{n_2}$ scaled as in (4.5), it holds

$$\frac{\|\omega_k(C_\mathrm{p}x_k)(x_k^\mathsf{T}B_\mathrm{u})\|_2}{|\mathrm{Re}(\lambda_k^+)(\mathrm{Im}(\lambda_k^+)\mathfrak{i} - \lambda_k^-)|} > \frac{\|\omega_j(C_\mathrm{p}x_j)(x_j^\mathsf{T}B_\mathrm{u})\|_2}{|\mathrm{Re}(\lambda_j^+)(\mathrm{Im}(\lambda_j^+)\mathfrak{i} - \lambda_j^-)|}$$

for all $j \neq k$. ◊

Note that the new dominance measure in Definition 4.1 and the idea in [168] are, in fact, identical in case of real pole pairs but not for complex conjugate ones.

A new dominant pole algorithm can now be developed, which computes the r_2 most dominant pole pairs of (4.7) such that the reduced-order model is given by

$$\widehat{G}_\mathrm{L}(s) = \sum_{k=1}^{r_2} \frac{\omega_k(C_\mathrm{p}x_k)(x_k^\mathsf{T}B_\mathrm{u})}{(s-\lambda_k^+)(s-\lambda_k^-)} \approx G_\mathrm{L}(s),$$

using an appropriate ordering in (4.7) with respect to Definition 4.1. The preserved structure in the pole-residue form enforces the reduced-order model to be also a modally damped second-order system.

4.1.2 Computing dominant pole pairs

After introducing the definition of dominant pole pairs and the structured pole-residue form of modally damped second-order systems, an algorithm for the computation of dominant pole pairs and reduced-order models is needed. The algorithmic ideas presented here are based on [48, Algorithm 1] leading to the dominant pole algorithm for modally damped second-order systems as summarized in Algorithm 4.1. The resulting algorithm can be found similarly in [168].

All dominant pole algorithms are based on observing the transfer function behavior close to system poles. For (4.3), it holds

$$g(s) := \frac{1}{\sigma_{\max}(G_\mathrm{L}(s))} \to 0, \tag{4.10}$$

when s approaches a pole of G_L. Here, $\sigma_{\max}(X)$ denotes the largest singular value of a matrix X. In principle, dominant pole algorithms apply a Newton scheme to (4.10) to find the zeros of $g(s)$, i.e., the poles of the transfer function. The convergence behavior of this Newton scheme is analyzed in [164]. It resembles an iteration over solutions of linear systems of the form

$$(\sigma^2 M + \sigma E + K)v = \tilde{u} \quad \text{and} \quad (\sigma^2 M + \sigma E + K)^\mathsf{H} w = \tilde{y}, \tag{4.11}$$

for the solution vectors $v, w \in \mathbb{C}^{n_2}$, the shift $\sigma \in \mathbb{C}$ and right-hand side vectors $\tilde{u}, \tilde{y} \in \mathbb{C}^{n_2}$. The Newton scheme would additionally involve an update of the solutions v and w as suggested in [163], but it is mentioned in [48] that in case of deflation with the system's input and output vectors, the Newton update becomes obsolete. In general, the right-hand sides in (4.11) would consist of the system's input and output matrices. But following the ideas in [48, 162], a tangential approach is used in (4.11) to compress multiple input and output vectors in case of MIMO systems. This tangential approach sets

$$\tilde{u} = B_{\mathrm{u}} u, \quad \text{and} \quad \tilde{y} = C_{\mathrm{p}} y,$$

with u and y pre-selected directions, usually chosen to be singular vectors corresponding to the largest singular value of the transfer function for selected shifts. Overall, this gives Step 4 in Algorithm 4.1.

In [48], the solutions of the linear systems, here (4.11), are collected into left and right projection bases W and V. These bases are then used for the subspace acceleration approach. For modally damped systems (4.1), one-sided projection, i.e., setting $V = W$ in (3.4), preserves the modal damping property in intermediate reduced-order models. Assume that only a single basis $V \in \mathbb{R}^{n_2 \times r}$ is given, the subspace acceleration approach [162] truncates the original system (4.1) to get an intermediate reduced-order model

$$\widetilde{G}_{\mathrm{L}} = (\widetilde{M}, \widetilde{E}, \widetilde{K}, \widetilde{B}_{\mathrm{u}}, \widetilde{C}_{\mathrm{p}}, 0), \tag{4.12}$$

with

$$\widetilde{M} = V^{\mathsf{T}} M V, \quad \widetilde{E} = V^{\mathsf{T}} E V, \quad \widetilde{K} = V^{\mathsf{T}} K V, \quad \widetilde{B}_{\mathrm{u}} = V^{\mathsf{T}} B_{\mathrm{u}}, \quad \widetilde{C}_{\mathrm{p}} = C_{\mathrm{p}} V.$$

The truncated system (4.12) is small, namely of dimension $r \ll n_2$, and has exactly the same structure and properties as the original system, i.e., $\widetilde{M}, \widetilde{E}, \widetilde{K}$ are symmetric positive definite and $\widetilde{E}\widetilde{M}^{-1}\widetilde{K} = \widetilde{K}\widetilde{M}^{-1}\widetilde{E}$ holds. Therefore, the formulae (4.5), (4.6), and (4.8) can be used to compute all poles of (4.12) in pairs with their corresponding eigenvectors. The pole pairs of (4.12) are approximations to the pole pairs of (4.1) and by back-projection, also the corresponding eigenvectors are approximated. Consequently, the pole pairs and residues of the full-order system can be approximated.

The intermediate structure-preservation is an important point in the application of the theory of dominant pole pairs of modally damped mechanical systems. Step 5 in Algorithm 4.1 suggests the concatenation of the left and right projection bases from [48] to preserve input and output information of the original system (4.1). Two different special cases, and their combination, can occur here:

(i) Real shifts σ_j lead to $\mathrm{Im}(v_j) = \mathrm{Im}(w_j) = 0$, which results in only $\mathrm{Re}(v_j)$ and $\mathrm{Re}(w_j)$ extending the projection space.

(ii) As observed in [48], close to exact poles $v_j \to \overline{w}_j$ holds, i.e., the real and imaginary parts of v_j and w_j provide the same information to the subspace. Consequently, only v_j or w_j should be used.

It is necessary to take care of the different occurring special cases in an implementation of Algorithm 4.1.

Afterwards, the approximation quality is evaluated for the most dominant pole pairs of (4.12) by computing the corresponding residuals. While in theory, the residuals for both poles of a pair are identical, they can differ in finite arithmetic. If the newly found pole pair with corresponding eigenvector is exact enough, it is deflated using one of the approaches in [163] with an underlying first-order realization of (4.1). Otherwise, the most dominant approximation of a pole pair is chosen as shifts in (4.11) in the next iteration step.

The complete *second-order modally damped dominant pole algorithm (SOMDDA)* is summarized in Algorithm 4.1. In the context of model reduction, the eigenvector matrix X from the output of the algorithm is then used as basis of the projection spaces, i.e., reduced-order models are computed by (3.4) with $W = V = X$. The latest version of an implementation of Algorithm 4.1 in MATLAB is published in [59].

Remark 4.2 (Alternative dominance measures):
While Definition 4.1 is the recommended measure for choosing dominant poles, different alternatives can be used in Algorithm 4.1 to get other desired results or to change the practical convergence behavior of the algorithm. The following measures are implemented in [59]:

(i) dominance in the $\mathcal{H}_\infty$-sense (Definition 4.1): $\dfrac{\|R_k\|_2}{|\mathrm{Re}(\lambda_k^+)(\mathrm{Im}(\lambda_k^+)\mathfrak{i} - \lambda_k^-)|}$,

(ii) product of real parts as in [168]: $\dfrac{\|R_k\|_2}{\mathrm{Re}(\lambda_k^+)\,\mathrm{Re}(\lambda_k^-)}$,

(iii) the absolute value of the rightmost pole: $\dfrac{\|R_k\|_2}{|\lambda_k^+|}$,

(iv) distance to the imaginary axis of the rightmost pole: $\dfrac{\|R_k\|_2}{|\mathrm{Re}(\lambda_k^+)|}$,

(v) product of pole pair: $\dfrac{\|R_k\|_2}{|\lambda_k^+\lambda_k^-|}$,

where $R_k = \omega_k(C_\mathrm{p}x_k)(x_k^\mathsf{T}B_\mathrm{u})$ is the residue corresponding to the pole pair $(\lambda_k^+, \lambda_k^-)$. ◊

While the first two measures were discussed before, the measures in Parts (iii) and (iv) in Remark 4.2 correspond to the classical definition of dominant poles (Definition 3.1) with taking only the component into account, which is potentially closer to the imaginary axis. The last measure in Remark 4.2 combines the ideas of (ii) and (iii).

Algorithm 4.1: Second-order modally damped dominant pole algorithm.

Input: System matrices M, E, K symmetric positive definite with $EM^{-1}K = KM^{-1}E$, B_u, C_p from (4.1), initial shift σ_1, residual tolerance $0 < \tau \ll 1$, number of requested pole pairs k_want.

Output: Eigenvector matrix X, dominant pole pairs $\lambda^{\pm} = \begin{bmatrix} \lambda_1^{\pm} & \dots & \lambda_k^{\pm} \end{bmatrix}$.

Initialize $V = X = [\,]$, $\lambda^{\pm} = [\,]$, $k = 0$, $j = 1$.

Compute the left and right singular vectors y_0 and u_0 of $\sigma_{\max}(G_\mathrm{L}(\sigma_1))$.

while $k < k_\mathrm{want}$ **do**

Solve the linear systems of equations

$$\left(\sigma_j^2 M + \sigma_j E + K\right) v_j = B_\mathrm{u} u_k \quad \text{and} \quad \left(\overline{\sigma}_j^2 M + \overline{\sigma}_j E + K\right) w_j = C_\mathrm{p}^\mathsf{T} y_k.$$

Expand the projection basis

$$V = \texttt{orth}\left(\begin{bmatrix} V & \operatorname{Re}(v_j) & \operatorname{Im}(v_j) & \operatorname{Re}(w_j) & \operatorname{Im}(w_j) \end{bmatrix}\right).$$

Compute the most dominant eigentriple $(\theta_j^+, \theta_j^-, \tilde{x}_j)$ of

$$\widetilde{G}_\mathrm{L} = (V^\mathsf{T} M V, V^\mathsf{T} E V, V^\mathsf{T} K V, V^\mathsf{T} B_\mathrm{u}, C_\mathrm{p} V, 0),$$

using (4.5), (4.6) and (4.8).

Compute the corresponding eigenvector and residuals

$$\begin{aligned} x_j &= V \tilde{x}_j, \\ r_j^+ &= \left(\left(\theta_j^+\right)^2 M + \theta_j^+ E + K\right) x_j, \\ r_j^- &= \left(\left(\theta_j^-\right)^2 M + \theta_j^- E + K\right) x_j. \end{aligned}$$

if $\max(\|r_j^+\|_2, \|r_j^-\|_2) < \tau$ **then**

Set $k = k + 1$ and $X = \begin{bmatrix} X & x_j \end{bmatrix}$, $\lambda^{\pm} = \begin{bmatrix} \lambda^{\pm} & \theta_j^{\pm} \end{bmatrix}$.

Deflate newly found eigentriple.

Update right and left singular vectors y_k and u_k of $\sigma_{\max}(H(\theta_j^+))$.

Set $\sigma_{j+1} = \theta_j^+$ and $j = j + 1$.

Restart if necessary.

4.1.3 Bounding the approximation error in the $\mathcal{H}_\infty$-norm

In practical situations, it is advantageous to be able to guarantee a certain approximation quality of the computed reduced-order model in a given system norm (Definition 2.7). In the unstructured first-order case (2.8), with the assumption of diagonalizability, the error in the $\mathcal{H}_\infty$-norm for modal truncation methods can be bounded by rewriting (3.6) into

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix}\mathsf{x}_1(t)\\ \mathsf{x}_2(t)\end{bmatrix} = \begin{bmatrix}\mathsf{A}_1 & 0\\ 0 & \mathsf{A}_2\end{bmatrix}\begin{bmatrix}\mathsf{x}_1(t)\\ \mathsf{x}_2(t)\end{bmatrix} + \begin{bmatrix}\mathsf{B}_1\\ \mathsf{B}_2\end{bmatrix} u(t),$$
$$y(t) = \begin{bmatrix}\mathsf{C}_1 & \mathsf{C}_2\end{bmatrix}\begin{bmatrix}\mathsf{x}_1\\ \mathsf{x}_2\end{bmatrix},$$

where the subscript-1 matrices belong to the reduced-order model, with A_1 containing the r_1 chosen eigenvalues, and the subscript-2 matrices are the truncated parts. Then, the $\mathcal{H}_\infty$-approximation error can generally be bounded by

$$\|\mathsf{G}_\mathrm{L} - \widehat{\mathsf{G}}_\mathrm{L}\|_{\mathcal{H}_\infty} \leq \frac{\|\mathsf{B}_2\|_2\|\mathsf{C}_2\|_2}{\min\limits_{\lambda\in\Lambda(\mathsf{A}_2)}|\mathrm{Re}(\lambda)|}; \tag{4.13}$$

see [50, 99]. A similar bound can be derived for the modal truncation of modally damped second-order systems. Consider the matrix of appropriately scaled eigenvectors X from (4.5) and (4.6) such that the transfer function (4.3) can be written with diagonal system matrices

$$G_\mathrm{L}(s) = \begin{bmatrix}C_1 & C_2\end{bmatrix}\left(s^2\begin{bmatrix}\Omega_1^{-1} & 0\\ 0 & \Omega_2^{-1}\end{bmatrix} + 2s\begin{bmatrix}\Xi_1 & 0\\ 0 & \Xi_2\end{bmatrix} + \begin{bmatrix}\Omega_1 & 0\\ 0 & \Omega_2\end{bmatrix}\right)^{-1}\begin{bmatrix}B_1\\ B_2\end{bmatrix}.$$

Again, the subscript-1 matrices belong to the reduced-order model and subscript 2 are the truncated parts. Then, the $\mathcal{H}_\infty$-error can be bounded by

$$\begin{aligned}\|\mathsf{G}_\mathrm{L} - \widehat{\mathsf{G}}_\mathrm{L}\|_{\mathcal{H}_\infty} &= \sup_{f\in\mathbb{R}}\|C_2(-f^2\Omega_2^{-1} + 2f\mathrm{i}\Xi_2 + \Omega_2)^{-1}B_2\|_2\\ &\leq \frac{\|C_2\|_2\|B_2\|_2}{\min\limits_{\lambda^\pm\in\Lambda(\Omega_2,2\Xi_2,\Omega_2^{-1})}|\mathrm{Re}(\lambda^+)(\mathrm{Im}(\lambda^+)\mathrm{i} - \lambda^-)|},\end{aligned} \tag{4.14}$$

where $\Lambda(\Omega_2, 2\Xi_2, \Omega_2^{-1})$ is the set of all truncated eigenvalues, i.e., the set of all eigenvalues of the quadratic eigenvalue problem using the truncated system matrices

$$(\lambda^2\Omega_2^{-1} + 2\lambda\Xi_2 + \Omega_2^{-1})x = 0.$$

The general problem of both bounds (4.13) and (4.14) is that for the norm of the truncated parts of input and output matrices, the full eigenvector basis is needed. This is usually not computable for large-scale systems.

An alternative to (4.14) can be found using the structured pole-residue form (4.7), where the poles are ordered with respect to Definition 4.1. Using the triangle inequality and (4.9), one obtains

$$\begin{aligned}
\|\mathsf{G}_{\mathrm{L}} - \widehat{\mathsf{G}}_{\mathrm{L}}\|_{\mathcal{H}_\infty} &= \sup_{f\in\mathbb{R}} \left\| \sum_{k=1}^{n_2} \frac{\omega_k(C_{\mathrm{p}}x_k)(x_k^{\mathsf{T}}B_{\mathrm{u}})}{(f\mathrm{i}-\lambda_k^+)(f\mathrm{i}-\lambda_k^-)} - \sum_{k=1}^{r_2} \frac{\omega_k(C_{\mathrm{p}}x_k)(x_k^{\mathsf{T}}B_{\mathrm{u}})}{(f\mathrm{i}-\lambda_k^+)(f\mathrm{i}-\lambda_k^-)} \right\|_2 \\
&= \sup_{f\in\mathbb{R}} \left\| \sum_{k=r_2+1}^{n_2} \frac{\omega_k(C_{\mathrm{p}}x_k)(x_k^{\mathsf{T}}B_{\mathrm{u}})}{(f\mathrm{i}-\lambda_k^+)(f\mathrm{i}-\lambda_k^-)} \right\|_2 \\
&\leq \sup_{f\in\mathbb{R}} \sum_{k=r_2+1}^{n_2} \frac{\|\omega_k(C_{\mathrm{p}}x_k)(x_k^{\mathsf{T}}B_{\mathrm{u}})\|_2}{|(f\mathrm{i}-\lambda_k^+)(f\mathrm{i}-\lambda_k^-)|} \\
&= \sum_{k=r_2+1}^{n_2} \frac{\|\omega_k(C_{\mathrm{p}}x_k)(x_k^{\mathsf{T}}B_{\mathrm{u}})\|_2}{\min\limits_{f\in\mathbb{R}}|(f\mathrm{i}-\lambda_k^+)(f\mathrm{i}-\lambda_k^-)|} \\
&= \sum_{k=r_2+1}^{n_2} \frac{\|\omega_k(C_{\mathrm{p}}x_k)(x_k^{\mathsf{T}}B_{\mathrm{u}})\|_2}{|\mathrm{Re}(\lambda_k^+)(\mathrm{Im}(\lambda_k^+)\mathrm{i}-\lambda_k^-)|},
\end{aligned} \tag{4.15}$$

where r_2 is the order of the reduced-order model and the number of preserved dominant pole pairs. As for the previous bound (4.14), the new bound (4.15) would, in principle, need the computation of all truncated pole pairs of the modally damped system (4.1), which is infeasible in practice. This issue can be overcome by using the ordering of the pole pairs with respect to the $\mathcal{H}_\infty$-based dominance measure (Definition 4.1), i.e., it holds that

$$\frac{\|\omega_k(C_{\mathrm{p}}x_k)(x_k^{\mathsf{T}}B_{\mathrm{u}})\|_2}{|\mathrm{Re}(\lambda_k^+)(\mathrm{Im}(\lambda_k^+)\mathrm{i}-\lambda_k^-)|} \geq \frac{\|\omega_j(C_{\mathrm{p}}x_j)(x_j^{\mathsf{T}}B_{\mathrm{u}})\|_2}{|\mathrm{Re}(\lambda_j^+)(\mathrm{Im}(\lambda_j^+)\mathrm{i}-\lambda_j^-)|},$$

for all $j > k$. This can be used to over-estimate the dominance measure of non-computed pole pairs. Therefore, assume that $k_{\mathrm{want}} \geq r_2$ pole pairs were computed via Algorithm 4.1, then one can bound the $\mathcal{H}_\infty$ error by

$$\begin{aligned}
\|\mathsf{G}_{\mathrm{L}} - \widehat{\mathsf{G}}_{\mathrm{L}}\|_{\mathcal{H}_\infty} \leq & \sum_{k=r_2+1}^{k_{\mathrm{want}}} \frac{\|\omega_k(C_{\mathrm{p}}x_k)(x_k^{\mathsf{T}}B_{\mathrm{u}})\|_2}{|\mathrm{Re}(\lambda_k^+)(\mathrm{Im}(\lambda_k^+)\mathrm{i}-\lambda_k^-)|} \\
& + (n_2 - k_{\mathrm{want}}) \frac{\|\omega_{k_{\mathrm{want}}}(C_{\mathrm{p}}x_{k_{\mathrm{want}}})(x_{k_{\mathrm{want}}}^{\mathsf{T}}B_{\mathrm{u}})\|_2}{|\mathrm{Re}(\lambda_{k_{\mathrm{want}}}^+)(\mathrm{Im}(\lambda_{k_{\mathrm{want}}}^+)\mathrm{i}-\lambda_{k_{\mathrm{want}}}^-)|}.
\end{aligned} \tag{4.16}$$

In contrast to (4.15), the new bound (4.16) is computable in practice, under the assumption that the most dominant poles were computed correctly. Also, the new bound (4.16) becomes sharper if more pole pairs are computed since (4.16) approaches (4.15) for $k_{\mathrm{want}} \to n_2$. It is a common approach in modal truncation to compute more poles than actually needed to increase the chance for sparse eigenvalue solvers to actually compute

the desired eigenvalues. The gap between (4.15) and (4.16) depends on the decay of the dominance measure, as well as the size of the original system n_2 and the number of computed dominant pole pairs k_{want}. It will become larger if $k_{\text{want}} \ll n_2$ and the dominance measure continues to decay after the computed k_{want} pole pairs. In practice, the constant multiplied with the last dominance measure term will be dominated by the size of the original system, which results in a vast overestimation of the $\mathcal{H}_\infty$-error in cases where the dominance measure of the last computed pole pair is not small enough but could decay further for upcoming pairs.

In general, the bounds (4.15) and (4.16) imply that the dominant pole algorithm for modally damped second-order systems provides good approximations if the dominance measure (Definition 4.1) decays fast. On the other hand, a slow decay, or even stagnation, of the dominance measure indicates difficulties in approximating the original system via its pole pairs.

4.1.4 Basis enrichment via rational Krylov subspaces

While modal truncation approaches are known to well approximate input-to-output behavior related to single system poles, i.e., peaks in the frequency response behavior, they usually fail to approximate "flat" regions or behavior that is determined by clusters of poles. Therefore, it is recommended for model order reduction methods to enrich the modal truncation basis with additional basis vectors to improve the approximation quality especially in those regions, where the behavior of the system poles is less dominant. An efficient model reduction approach to improve the approximation behavior of the reduced-order model in desired frequency regions are interpolatory methods (Krylov subspace methods); see Section 3.3.

First, consider the first-order system (2.8). Let V_{mt} and W_{mt} be right and left basis matrices for modal truncation, i.e., it holds

$$\mathsf{x}_i \in \operatorname{span}(V_{\text{mt}}) \quad \text{and} \quad \mathsf{y}_i \in \operatorname{span}(W_{\text{mt}}) \tag{4.17}$$

for $1 \leq i \leq r_1$ and x_i, y_i from (3.5) corresponding to the chosen eigenvalues λ_i. Because of (4.17), reduced-order models computed by (3.2) preserve the chosen system poles if $\operatorname{span}(V_{\text{mt}})$ and $\operatorname{span}(W_{\text{mt}})$ are contained in the final projection spaces. Given now two other truncation bases V_2 and W_2, e.g., constructed by transfer function interpolation, the final truncation bases can be constructed via the underlying projection spaces such that

$$\operatorname{span}(V) \supseteq \operatorname{span}\left(\begin{bmatrix} V_{\text{mt}} & V_2 \end{bmatrix}\right) \quad \text{and} \quad \operatorname{span}(W) \supseteq \operatorname{span}\left(\begin{bmatrix} W_{\text{mt}} & W_2 \end{bmatrix}\right) \tag{4.18}$$

hold. Since (4.17) translates into (4.18) by construction, reduced-order models constructed by projection (3.2) with the basis matrices as in (4.18) also preserve the chosen poles from the modal truncation approach, independent of the second chosen model reduction

basis. While a lot of projection-based model order reduction methods could be used for the construction of V_2 and W_2, interpolation-based methods are the recommended choice. They have cheap computational costs and, in contrast to many other methods, the interpolation property is given via subspace conditions in Proposition 3.2, i.e., the interpolation conditions satisfied by V_2 and W_2 are inherited in (4.18) such that also the final reduced-order model fulfills the same interpolation conditions.

The modal truncation method with basis enrichment via rational Krylov subspaces was used in [178] for second-order systems to accelerate simulations of machine tools via reduced-order models. Therein, only the undamped second-order system with $E = 0$:

$$\begin{aligned} M\ddot{x}(t) + Kx(t) &= B_{\mathrm{u}}u(t), \\ y(t) &= C_{\mathrm{p}}x(t), \end{aligned}$$

is considered for the generation of the modal and Krylov bases. But the idea of basis enrichment can similarly be used for modally damped second-order systems (4.1). A suitable structure-preserving projection method for the basis enrichment is given using the theory from Section 3.3.3 and Proposition 3.2. The resulting structure-preserving dominant pole algorithm with basis enrichment is summarized in Algorithm 4.2. The following remarks give some ideas for an explicit implementation of the algorithm.

Remark 4.3 (Number of dominant pole pairs):
As mentioned in Section 4.1.3, the second-order modally damped dominant pole algorithm is comparably cheap in computational costs and can be run for more than the desired number of dominant pole pairs, resulting in the choice of the k_{mt} most dominant poles to remain in the reduced-order model. This number of remaining pole pairs k_{mt} can be adaptively chosen, for example, by the $\mathcal{H}_\infty$-error bound in (4.16), by observing stagnation of the computed dominance measure (Definition 4.1), or by truncating dominant pole pairs with dominance measure below a given tolerance. Especially, a drop in the dominance measure indicates a good point for truncating pole pairs. ◊

Remark 4.4 (Choosing interpolation points for basis enrichment):
The choice of interpolation algorithms gives quite an amount of freedom to the user in terms of realizing Algorithm 4.2. The only restriction done in Algorithm 4.2 is the requirement of a single resulting basis matrix V_{kry} to preserve the modal damping property of the original system. In general, interpolation can be performed via a one-sided projection anyway (cf. Proposition 3.2). In case of interpolation via two-sided projection, to additionally match the frequency sensitivities, the two computed basis matrices V and W can be combined into a single basis by concatenation

$$V_{\mathrm{kry}} = \mathtt{orth}\left(\begin{bmatrix} V & W \end{bmatrix}\right).$$

Two example choices for the interpolation points are outlined below:

Algorithm 4.2: SOMDDPA with basis enrichment via structured interpolation.

Input: System matrices M, E, K symmetric positive definite with $EM^{-1}K = KM^{-1}E$, B_{u}, C_{p} from (4.1), number of pole pairs k_{mt} in the reduced-order model.

Output: Matrices of the modally damped reduced-order system $\widehat{M}, \widehat{E}, \widehat{K}, \widehat{B}_{\mathrm{u}}, \widehat{C}_{\mathrm{p}}$.

1 Compute the eigenvector basis X for $k_{\mathrm{want}} \geq k_{\mathrm{mt}}$ pole pairs using Algorithm 4.1 with the system matrices $M, E, K, B_{\mathrm{u}}, C_{\mathrm{p}}$, an initial shift $\sigma_1 \in \mathbb{C}$ and the residual tolerance $0 < \tau \ll 1$.

2 Partition $X = \begin{bmatrix} X_1 & X_2 \end{bmatrix}$, with X_1 the eigenvectors corresponding to the k_{mt} most dominant pole pairs.

3 Compute a real interpolation basis V_{kry} by any interpolation algorithm based on Proposition 3.2 with

$$\mathcal{C}(s) = C_{\mathrm{p}}, \quad \mathcal{K}(s) = s2M + sE + K, \quad \mathcal{B}(s) = B_{\mathrm{u}}.$$

4 Compute the orthogonal truncation basis

$$V = \texttt{orth}\left(\begin{bmatrix} X_1 & V_{\mathrm{kry}} \end{bmatrix}\right).$$

5 Compute the reduced-order model

$$\widehat{M} = V^{\mathsf{T}}MV, \quad \widehat{E} = V^{\mathsf{T}}EV, \quad \widehat{K} = V^{\mathsf{T}}KV, \quad \widehat{B}_{\mathrm{u}} = V^{\mathsf{T}}B_{\mathrm{u}}, \quad \widehat{C}_{\mathrm{p}} = C_{\mathrm{p}}V.$$

(a) As the transfer function on the imaginary axis is enough for stable systems to describe their input-to-output behavior, interpolation points could be chosen as complex conjugate pairs on the imaginary axis in the frequency range of interest. Simple and efficient choices are, for example, logarithmically equidistant points or to choose the points in the intervals spanned by the imaginary parts of the computed dominant poles.

(b) Using the projection-based $\mathcal{H}_\infty$-norm computation from [6–8, 172] or error estimators [82] allows for a similar greedy model reduction approach as described in [26, 27] such that interpolation points that minimize the $\mathcal{H}_\infty$-approximation error can be computed. The combination with the dominant pole algorithm corresponds to an initialization of the greedy interpolation procedure ([26, 27]) with a reduced-order model from Algorithm 4.1. An advantageous side effect of this approach when using the algorithms for $\mathcal{H}_\infty$-norm approximation is the potentially very accurate, final $\mathcal{H}_\infty$-error as by-product. ◊

4.1.5 Numerical experiments

In this section, the new SOMDDA approaches are tested and compared to classical structure-preserving methods for second-order systems from Chapter 3. Therefore, the two linear benchmark examples from Chapter 1 are used, namely the butterfly gyroscope and the artificial fishtail model. The following list is an overview about the model reduction methods used in the comparisons and their notation:

SOMDDPA denotes the pure structure-preserving dominant pole algorithm from Algorithm 4.1,

SOMDDPA+StrInt(equi./$\mathcal{H}_\infty$) is the structure-preserving dominant pole algorithm with basis enrichment from Algorithm 4.2, where only one-sided interpolation (Proposition 3.2 Part (a)) is used with the interpolation points chosen either logarithmically equidistant on the imaginary axis (**equi.**) or via a successive greedy $\mathcal{H}_\infty$-selection ($\boldsymbol{\mathcal{H}_\infty}$),

MT is the classical modal truncation method (Section 3.2.1) computing the eigenvectors of the smallest eigenvalues of (3.8) as truncation basis,

SOBT(*p*/*pm*/*pv*/*vp*/*vpm*/*v*/*fv*/*so*) is the second-order balanced truncation method (Section 3.4.3) with the balancing formulae from Table 3.1,

StrInt(equi./$\mathcal{H}_\infty$/IRKA) denotes the structure-preserving interpolation method using the one-sided interpolation (Proposition 3.2 Part (a)) and the interpolation points chosen either logarithmically equidistant on the imaginary axis (**equi.**), via $\mathcal{H}_\infty$-greedy selection ($\boldsymbol{\mathcal{H}_\infty}$) or as $\mathcal{H}_2$-optimal points from TF-IRKA (**IRKA**),

StrInt(avg.) computes the reduced-order model by approximating an oversampled interpolation subspace as in Remark 3.3 using the pivoted QR decomposition for the basis truncation.

The numerical comparison of the different methods will be done using the MORscore from Section 2.4.2, where the table columns corresponding to the time domain measures (2.44) and (2.45) are denoted by L_2 and L_∞, respectively, and for the frequency domain measure (2.46) by $\mathcal{H}_\infty$. For a more detailed discussion, a practical reduced order r_2 is selected, for which the best performing methods are compared in frequency and time domains using pointwise relative errors. In frequency domain, this will be

$$\epsilon_{\mathrm{rel}}(\omega) := \frac{\|G_{\mathrm{L}}(\omega\mathrm{i}) - \widehat{G}_{\mathrm{L}}(\omega\mathrm{i})\|_2}{\|G_{\mathrm{L}}(\omega\mathrm{i})\|_2}, \tag{4.19}$$

with the frequency range of interest $\omega \in [\omega_{\min}, \omega_{\max}] \subset \mathbb{R}$, and in time domain

$$\epsilon_{\mathrm{rel}}(t) := \frac{\|y(t) - \hat{y}(t)\|_2}{\|y(t)\|_2}, \tag{4.20}$$

Table 4.1: MORscores for the butterfly gyroscope example with reduced orders from 1 to 30, and the percentage of stable reduced-order models.

Method	$\mathcal{H}_\infty$	L_2	L_∞	Stab. ratio
SOMDDPA	0.2540	0.2188	0.2090	1.0000
SOMDDPA+StrInt(equi.)	0.1964	0.1980	0.1912	1.0000
SOMDDPA+StrInt($\mathcal{H}_\infty$)	0.2943	0.2328	0.2295	1.0000
MT	0.2094	0.1739	0.1677	1.0000
SOBT(p)	0.3147	0.2799	0.2758	1.0000
SOBT(pm)	0.1669	0.1170	0.1115	0.1333
SOBT(pv)	0.3165	0.2666	0.2634	1.0000
SOBT(vp)	0.1153	0.0539	0.0473	0.5000
SOBT(vpm)	0.0965	0.0725	0.0709	0.0000
SOBT(v)	0.3007	0.2665	0.2604	0.9667
SOBT(fv)	0.2714	0.2506	0.2455	1.0000
SOBT(so)	0.3079	0.2446	0.2383	0.8333
StrInt(equi.)	0.1355	0.1486	0.1433	1.0000
StrInt($\mathcal{H}_\infty$)	0.2853	0.2281	0.2238	1.0000
StrInt(IRKA)	0.2474	0.2118	0.2083	1.0000
StrInt(avg.)	0.2234	0.2095	0.2018	1.0000

with the time interval $t \in [t_0, t_\mathrm{f}]$ used for simulations.

4.1.5.1 Butterfly gyroscope

The butterfly gyroscope is a mechanical system with $n_2 = 17\,361$ second-order differential equations, $m = 1$ input and $p = 12$ position outputs. It is used as motivational example in Section 1.3.1. The internal damping is modeled via the Rayleigh approach (4.4), $E = \alpha M + \beta K$, with $\alpha = 0$ and $\beta = 10^{-6}$. Therefore, this benchmark example belongs to the class of modally damped second-order systems (4.1).

The resulting MORscores for all implemented methods are shown in Table 4.1. The number of dominant pole pairs to reside in the reduced-order model for the SOMDDPA+StrInt approaches was fixed to 6. This choice was taken for the practical reason of comparability to the other methods. For the time domain simulation in the interval $[0, 0.01]\,\mathrm{s}$, the input was chosen to be a piecewise constant white noise signal

$$u(t) = \eta(t_j), \quad \text{for } t_j \le t < t_{j+1},$$

with $j = 0, \ldots, 99$, equidistant time steps $t_j = j \cdot \frac{0.01}{99}$ and presampled Gaussian white noise $\eta(t)$. The last column of Table 4.1 shows the relative amount of asymptotically

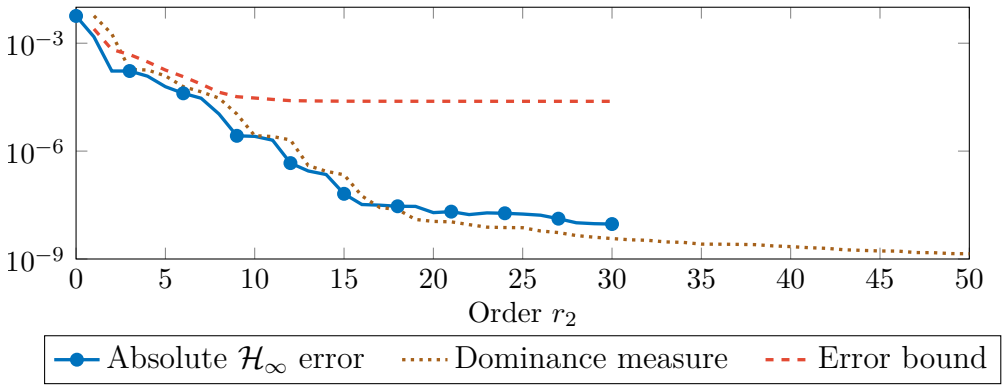


Figure 4.1: Comparison of dominance measure, $\mathcal{H}_\infty$-error bound (4.16) and absolute $\mathcal{H}_\infty$-error of SOMDDPA for the butterfly gyroscope example.

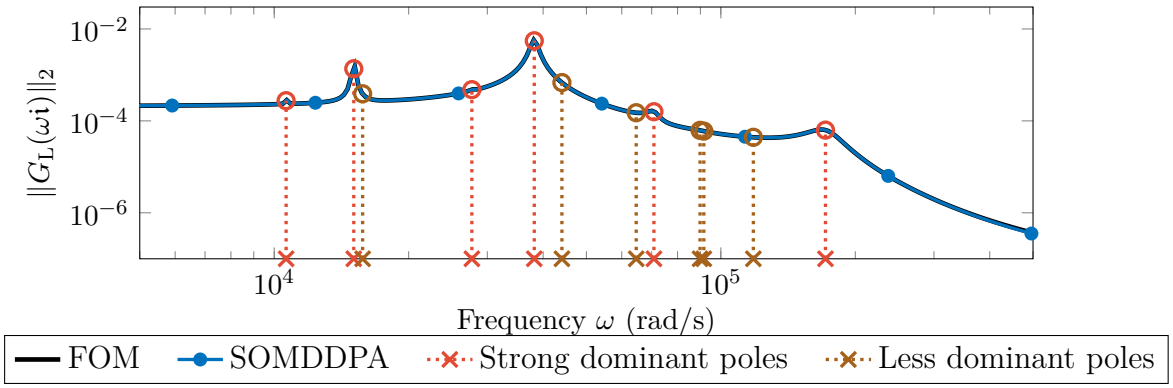
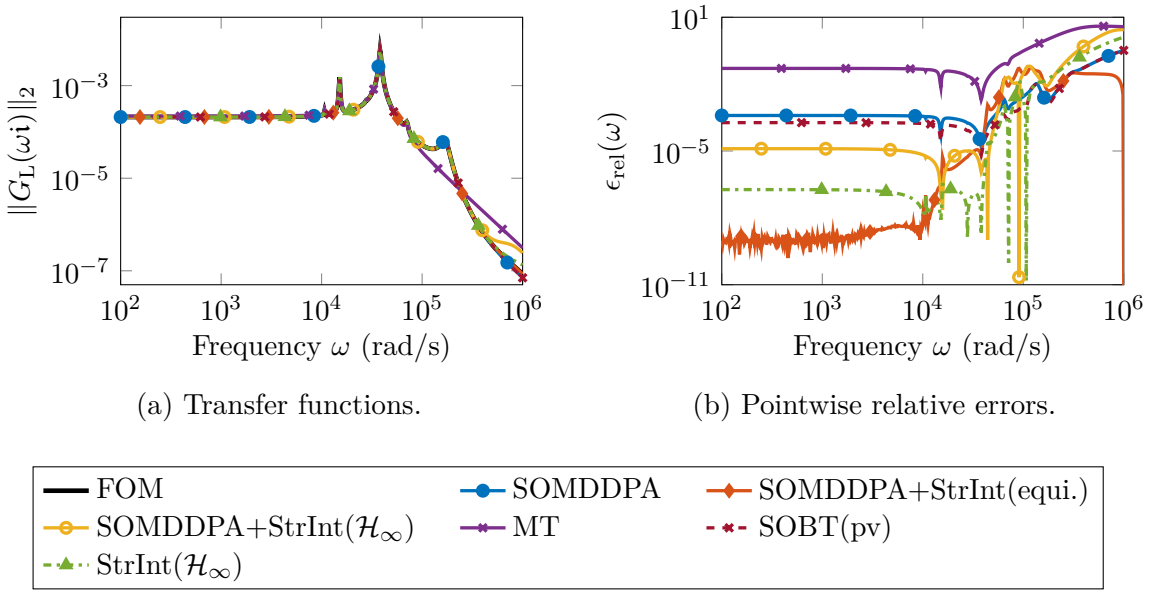


Figure 4.2: Projection of complex dominant poles onto the frequency axis and relation to the transfer function behavior for the butterfly gyroscope example.

stable reduced-order models, e.g., $0.8333 \cdot 30 \approx 25$ stable reduced-order models were computed for SOBT(so). By construction, the modal truncation and interpolation approaches always produce asymptotically stable reduced-order models. In general, beside some outliers within the SOBT methods, all techniques perform reasonably well in this example.

Taking a close look at the new approaches, one can observe that the pure SOMDDPA and SOMDDPA+StrInt($\mathcal{H}_\infty$) perform exceptionally better than the classical modal truncation method, MT. Also, SOMDDPA+StrInt(equi.) is still able to outperform the MT approach in time domain despite its very simple subspace enrichment strategy of equidistant interpolation points. Another interesting observation is that both SOMDDPA+StrInt methods perform better than their pure interpolation counterparts StrInt(equi.) and StrInt($\mathcal{H}_\infty$). Overall comparing the MORscores, the best of the dominant pole approaches is SOMDDPA+StrInt($\mathcal{H}_\infty$), which is only outperformed by some



(a) Transfer functions.

(b) Pointwise relative errors.

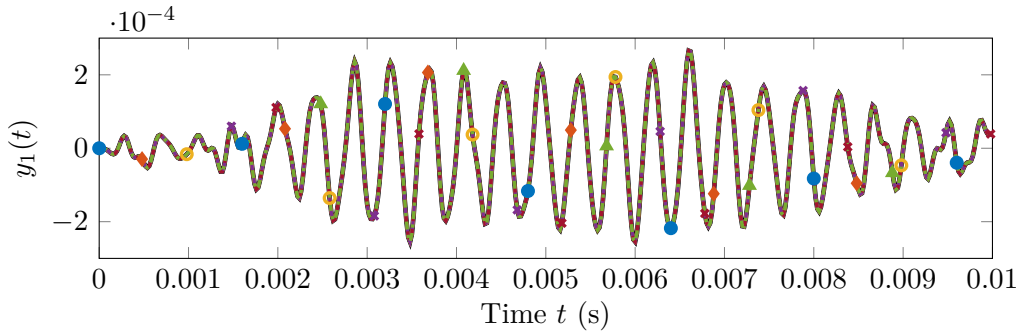
Figure 4.3: Frequency domain results for the butterfly gyroscope example.

of the second-order balanced truncation methods SOBT($p/pv/v/so$).

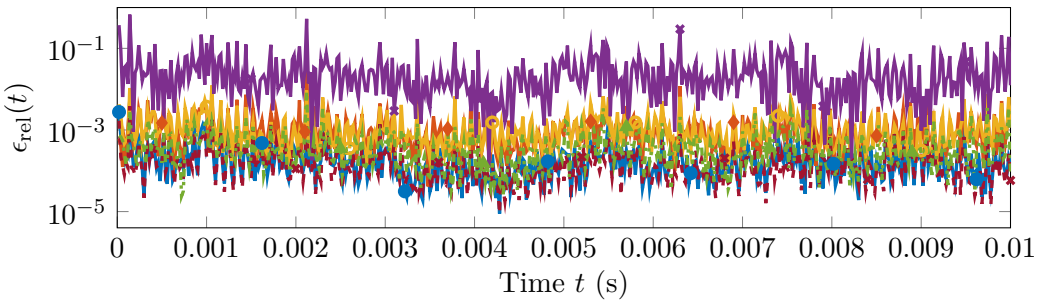
Figure 4.1 is used to compare the approximate error bound (4.16) with the actual $\mathcal{H}_\infty$-error of SOMDDPA and the corresponding dominance measure from Definition 4.1. Thereby, SOMDDPA was used to compute up to 50 dominant pole pairs of the original system, while reduced-order models were only computed up to order 30. One can see that in the beginning, the dominance measure nicely decays and the error bound captures very well the actual error behavior. However, after order 7 the error bound flattens out and stops tracking the reduction error. This is due to the weaker decay of the dominance measure arising after order $r_2 = 20$ in the order of magnitude of 10^{-8} and the multiplication with approximately n_2 in the error bound. Figure 4.1 also shows, as discussed in Section 4.1.3, that the stagnation of the dominance measure indicates the stagnation of the approximation error. Following the decay of the dominance measure, a good amount of dominant pole pairs to keep in the reduced-order model for the basis enrichment strategy would be between 15 and 20.

For a better understanding of the influence of dominant poles on the transfer function behavior, Figure 4.2 shows the first 12 pole pairs projected onto the imaginary axis and the transfer function of the full and reduced-order SOMDDPA model of order $r_2 = 12$. The "strong dominant poles" are the first 6 most dominant pole pairs and the "less dominant poles" the following 6 pairs. One can directly observe how the strong dominant poles resemble the peaks of the transfer function and lead to a matching approximation in these regions.

For a more detailed comparison, the reduced order $r_2 = 12$ is picked also for the other



(a) First output entry $y_1(t)$ of the time simulation.



(b) Pointwise relative errors of the complete output.

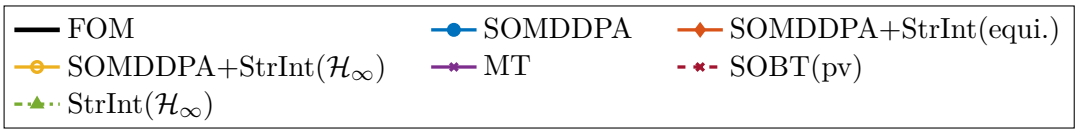


Figure 4.4: Time domain results for the butterfly gyroscope example.

model reduction methods. To keep the upcoming plots clearly arranged, only the best performing approaches from the second-order balancing and interpolation-based methods are chosen, namely SOBT(pv) and StrInt($\mathcal{H}_\infty$). Figure 4.3 shows the results in frequency domain. MT provides clearly the worst approximation where the transfer function is not even matched anymore for higher frequencies. The SOMDDPA+StrInt(equi.) method yields the overall best relative approximation error due to the interpolation points equally distributed over the frequency range of interest. The methods with $\mathcal{H}_\infty$-greedy interpolation also nicely match the transfer function except for high frequencies, where they begin to diverge. Figure 4.4 illustrates the approximation in the time domain with one selected example entry of the output vector in Figure 4.4a and the pointwise relative errors of the complete output signal in Figure 4.4b. All methods are performing equally well except for MT, which is several orders of magnitude worse than the rest.

Table 4.2: MORscores for the artificial fishtail example with reduced orders from 1 to 10, and the percentage of stable reduced-order models.

Method	$\mathcal{H}_\infty$	L_2	L_∞	Stab. ratio
SOMDDPA	0.2490	0.2191	0.2192	1.0000
SOMDDPA+StrInt(equi.)	0.2461	0.2095	0.2133	1.0000
SOMDDPA+StrInt($\mathcal{H}_\infty$)	0.2447	0.2011	0.2050	1.0000
MT	0.2043	0.1657	0.1651	1.0000
SOBT(p)	0.2537	0.2631	0.2649	0.9000
SOBT(pm)	0.2447	0.2125	0.2123	0.8000
SOBT(pv)	0.2540	0.2441	0.2460	0.9000
SOBT(vp)	0.2461	0.2457	0.2468	1.0000
SOBT(vpm)	0.2352	0.2397	0.2399	1.0000
SOBT(v)	0.2548	0.2677	0.2721	1.0000
SOBT(fv)	0.2103	0.1890	0.1933	1.0000
SOBT(so)	0.2553	0.2671	0.2709	1.0000
StrInt(equi.)	0.0954	0.0921	0.0926	1.0000
StrInt($\mathcal{H}_\infty$)	0.2018	0.1747	0.1803	1.0000
StrInt(IRKA)	0.2005	0.1796	0.1843	1.0000
StrInt(avg.)	0.2262	0.2013	0.2041	1.0000

4.1.5.2 Artificial fishtail model

As second numerical example, the artificial fishtail model from Section 1.3.2 is considered. The example has $n_2 = 779\,232$ states, $m = 1$ input and $p = 3$ position outputs. As in the previous example, Rayleigh damping (4.4) is used to model the internal behavior of the system, with $E = \alpha M + \beta K$, where $\alpha = 10^{-4}$ and $\beta = 2 \cdot 10^{-4}$. Therefore, the artificial fishtail model also belongs to the class of modally damped mechanical systems (4.1).

The MORscores of all compared methods are shown in Table 4.2. The maximum reduced order for the comparison was chosen to be 10, because in [174] this was chosen as reasonable large approximation order, and the SOMDDPA implementation is only capable of computing 12 pole pairs before stagnating in large clusters of very weakly observable/controllable eigenvalues. The number of dominant pole pairs to reside in the reduced-order model in the basis enrichment methods was set to be 4 for the same practical reasons as in the previous example. For MT, the eigenvalue computations in MATLAB using `eigs` turned out to be difficult due to a cluster of weakly observable/controllable eigenvalues close to the imaginary axis. To get similar results to [168, 174], the 50 smallest eigenvalues of (3.8) were computed using `eigs` and all values corresponding to the occurring eigenvalue cluster were removed for MT. For the time domain simulation

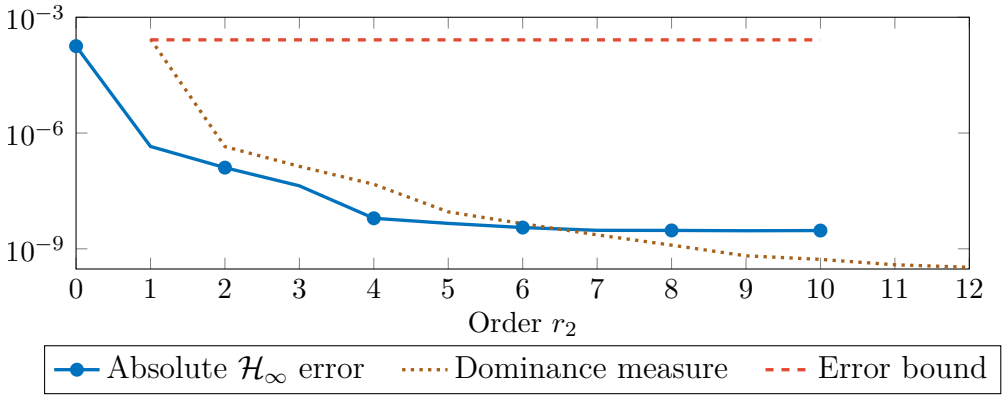


Figure 4.5: Comparison of dominance measure, $\mathcal{H}_\infty$-error bound (4.16) and absolute $\mathcal{H}_\infty$-error of SOMDDPA for the artificial fishtail example.

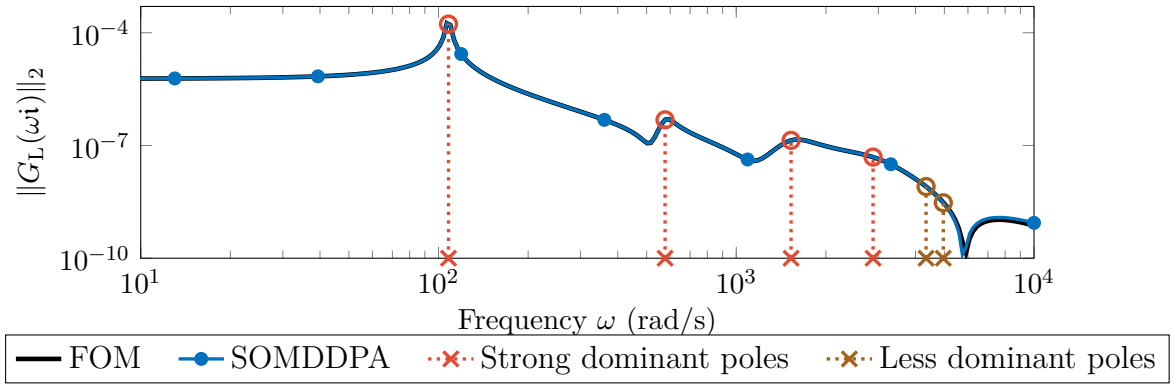


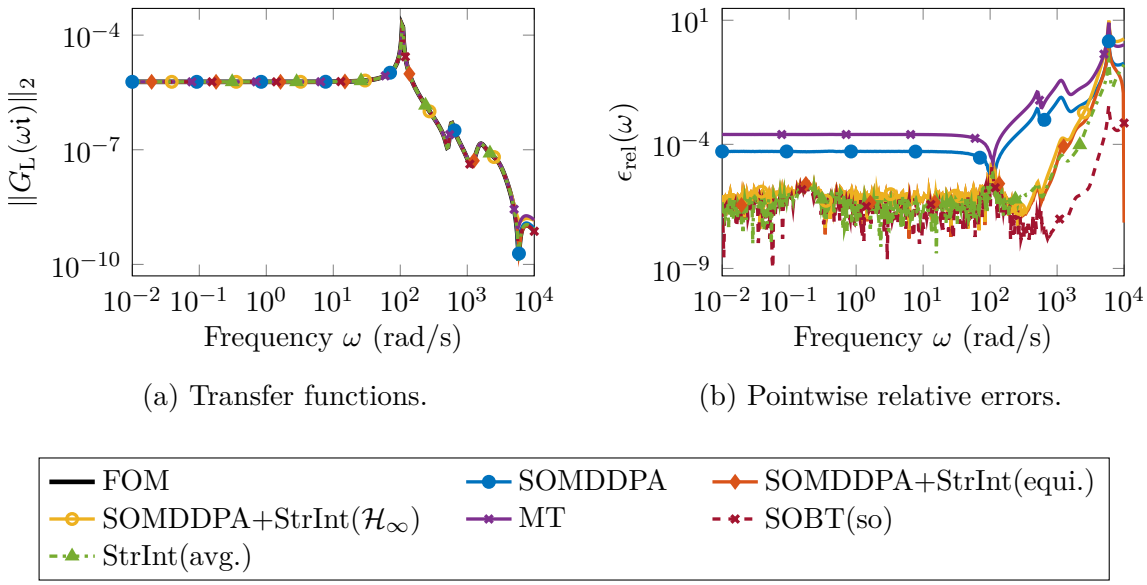
Figure 4.6: Projection of complex dominant poles onto the frequency axis and relation to the transfer function behavior for the artificial fishtail example.

in the interval $[0, 2]$ s, the input is chosen as piecewise constant white noise signal

$$u(t) = 5000 \cdot \eta(t_j), \quad \text{for } t_j \leq t < t_{j+1},$$

with $j = 0, \ldots, 99$, equidistant time steps $t_j = j \cdot \frac{2}{99}$ and presampled Gaussian white noise $\eta(t)$.

Comparing the MORscores in Table 4.2 reveals the SOMDDPA approaches to be good approximation methods that perform better than the classical MT, all interpolation-based methods and also some second-order balancing methods. In contrast to the previous example, the basis enrichment approach is not capable to improve the approximation quality compared to SOMDDPA. Figures 4.5 and 4.6 are used to give more inside about the dominant pole approach. Figure 4.5 shows, as for the previous example, the $\mathcal{H}_\infty$-error to behave very similar to the dominance measure from Definition 4.1. But for the artificial fishtail, the error bound (4.16) does not provide any information due to the early



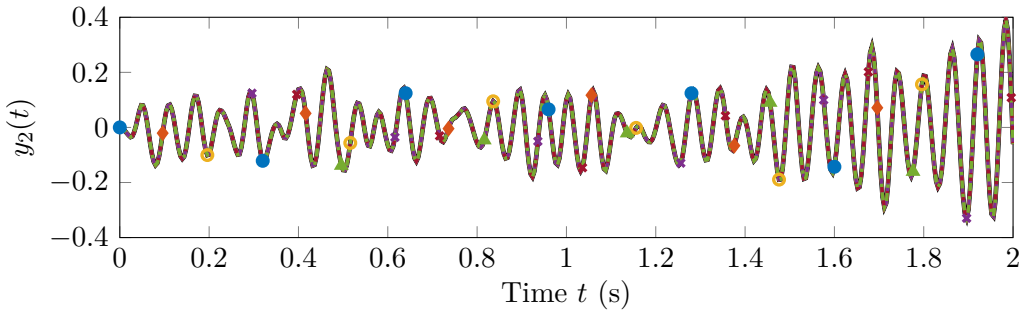
(a) Transfer functions.

(b) Pointwise relative errors.

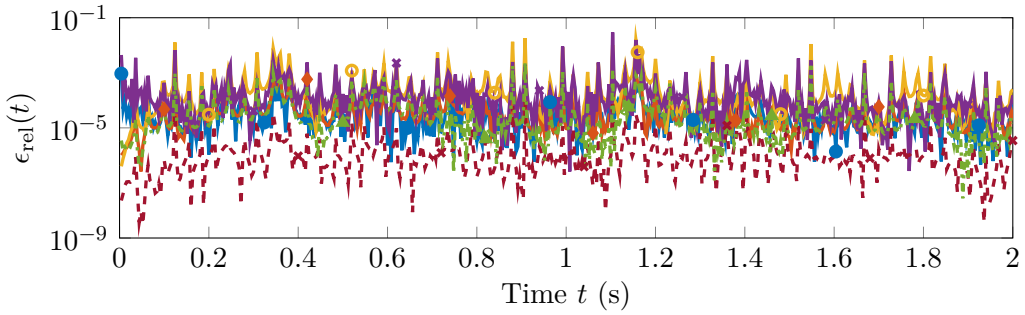
Figure 4.7: Frequency domain results for the artificial fishtail example.

stagnation of the dominance measure and the very large full-order state-space dimension. The complex dominant pole pairs are shown in Figure 4.6. The figure only shows 6 pairs since the rest of the pole pairs are real-valued. The 4 as "strong" denoted pole pairs are the most dominant ones and those which are chosen to reside in the reduced-order model in the basis enrichment methods. As recognized in the previous example, the most dominant poles exactly capture the peaks and their surrounding behavior of the transfer function very well.

For a more detailed comparison, the reduced order $r_2 = 10$ is chosen. For clarity in the upcoming plots, only selected reduction methods are chosen. SOBT(*so*) is used as representative of the second-order balancing methods and from the interpolation-based approaches StrInt(avg.) is taken. The approximation results in the frequency domain are shown in Figure 4.7. From the modal truncation methods, the SOMDDPA+StrInt($\mathcal{H}_\infty$) performs best and MT worst. MT clearly diverges from the original transfer function for frequencies close to 10^4 rad/s. Best performing is the SOBT(*so*) approach. A general problem in the approximation of the transfer function seems to be the sink close to 10^4 rad/s, which is best captured by SOMDDPA+StrInt($\mathcal{H}_\infty$) and SOBT(*so*). The time simulation results can be seen in Figure 4.8. The first and third entries of the system's output describe the fishtail movement in the non-horizontal directions, for which the original system's output is nearly zero. Therefore, Figure 4.8a only shows the second output entry giving the horizontal flapping movement of the fishtail. All chosen model reduction methods seem to capture the behavior of the original system in Figure 4.8a. Looking at the pointwise relative output errors for the complete system's



(a) Second output entry $y_2(t)$ of the time simulation.



(b) Pointwise relative errors of the complete output.

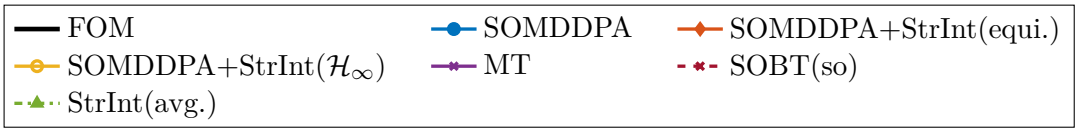


Figure 4.8: Time domain results for the artificial fishtail example.

output in Figure 4.8b reveals similar results to the frequency domain observations. MT performs again worst and SOBT(*so*) best, where also SOMDDPA and StrInt(avg.) yield comparably good approximations.

4.1.6 Conclusions

In this section, the idea of modal truncation via dominant poles was reconsidered for a special subclass of mechanical systems, namely those with modal damping. By using the special structure of the underlying quadratic eigenvalue problem, a structured pole-residue form was attained leading to the definition of dominant pole pairs for modally damped systems. An appropriate numerical procedure was developed to compute dominant pole pairs efficiently in a structure-preserving fashion, based on classical techniques

from first-order dominant pole algorithms. Two types of bounds for the absolute $\mathcal{H}_\infty$-approximation error were developed. While only being of limited practical use, these bounds imply a good approximation behavior of the method in cases of a fast decay of the dominance measure. Motivated by the observation that modal truncation quickly reaches its limits of approximation possibilities, a structure-preserving expansion of the constructed model reduction basis was suggested as refinement procedure using structured interpolation. In two numerical examples, the newly developed dominant pole algorithms for modally damped systems were compared to a variety of established structure-preserving model reduction methods and turned out to be very competitive alternatives in terms of approximation quality in time and frequency domains.

4.2 Second-order frequency- and time-limited balanced truncation methods

While most structure-preserving model reduction approaches, as well as the methods from the previous section (Section 4.1), aim for a globally sufficient approximation, this is not always necessary. In the presence of practical applications, often only local approximations of the original system's behavior in frequency or time domain are of actual interest, i.e., an approximation is only needed for a specified time or frequency range due to physical restrictions.

A class of approaches that can be used in the frequency domain to derive reduced-order models with locally good approximations is structured interpolation (Section 3.3.4). Interpolation-based methods usually provide good approximations in the surroundings of the chosen interpolation points. But this might not be sufficient for the approximation of a whole frequency region leading to larger numbers of interpolation points and, therefore, larger reduced-order models needed for the approximation. In case of first-order systems, the limited balanced truncation methods (cf. Sections 3.4.1 and 3.4.2) are suitable alternatives concerning local approximations in both frequency and time domains. Compared to the Krylov subspace approaches, these methods usually lead to a more uniform error behavior of the approximation in the ranges of interest. Therefore, one can expect smaller reduced-order models with the required approximation quality than using interpolation methods or global approximations. A first attempt of extending the limited balanced truncation approaches to second-order systems (2.17) was done in [107] for the frequency-limited case and, in the same fashion, in [108] for the time-limited case. While these references give a general idea, they are still incomplete concerning the concept of second-order balanced truncation methods and their application to the large-scale sparse system case. Also, they contain a general misconception about the problem of stability preservation in reduced-order second-order systems.

In the following, a full extension of the limited balanced truncation approaches from first- to second-order systems of the form (2.17) is presented, followed by proposed alternative methods for the problem of stability preservation and discussions on how to handle the large-scale sparse system case and numerical difficulties in computations. This section is based on the results published in [54, 57] and also partially available in [26, 27, 168].

4.2.1 Structured frequency-limited approach

The generalization of the frequency-limited balanced truncation method for second-order systems has been discussed in [107] for the position (p) and position-velocity (pv) balancing from [159] (cf. Table 3.1). A generalization to more second-order balanced truncation approaches can be done by the following observation: The block partitioning

of the Gramians (3.35) into position and velocity parts is given by

$$\begin{aligned} P_{\infty,\mathrm{p}} &= \begin{bmatrix} I_{n_2} & 0 \end{bmatrix} \mathsf{P}_\infty \begin{bmatrix} I_{n_2} \\ 0 \end{bmatrix}, & P_{\infty,\mathrm{v}} &= \begin{bmatrix} 0 & I_{n_2} \end{bmatrix} \mathsf{P}_\infty \begin{bmatrix} 0 \\ I_{n_2} \end{bmatrix}, \\ J_{\mathrm{fc}}^{\mathsf{T}} Q_{\infty,\mathrm{p}} J_{\mathrm{fc}} &= \begin{bmatrix} I_{n_2} & 0 \end{bmatrix} \mathsf{E}^{\mathsf{T}} \mathsf{Q}_\infty \mathsf{E} \begin{bmatrix} I_{n_2} \\ 0 \end{bmatrix}, & M^{\mathsf{T}} Q_{\infty,\mathrm{v}} M &= \begin{bmatrix} 0 & I_{n_2} \end{bmatrix} \mathsf{E}^{\mathsf{T}} \mathsf{Q}_\infty \mathsf{E} \begin{bmatrix} 0 \\ I_{n_2} \end{bmatrix}, \end{aligned} \tag{4.21}$$

for the infinite first- and second-order Gramians. Therefore, the extension of the existing second-order balanced truncation methods to the frequency-limited approach follows the replacement of the infinite first-order Gramians P_∞ and $\mathsf{E}^{\mathsf{T}}\mathsf{Q}_\infty\mathsf{E}$ in (4.21) by the first-order frequency-limited Gramians P_Ω and $\mathsf{E}^{\mathsf{T}}\mathsf{Q}_\Omega\mathsf{E}$ from (3.30), using the same first-order realization (2.18). Applying (4.21), the frequency-limited second-order Gramians are defined to be

$$\begin{aligned} P_{\Omega,\mathrm{p}} &:= \begin{bmatrix} I_{n_2} & 0 \end{bmatrix} \mathsf{P}_\Omega \begin{bmatrix} I_{n_2} \\ 0 \end{bmatrix}, & P_{\Omega,\mathrm{v}} &:= \begin{bmatrix} 0 & I_{n_2} \end{bmatrix} \mathsf{P}_\Omega \begin{bmatrix} 0 \\ I_{n_2} \end{bmatrix}, \\ J_{\mathrm{fc}}^{\mathsf{T}} Q_{\Omega,\mathrm{p}} J_{\mathrm{fc}} &:= \begin{bmatrix} I_{n_2} & 0 \end{bmatrix} \mathsf{E}^{\mathsf{T}} \mathsf{Q}_\Omega \mathsf{E} \begin{bmatrix} I_{n_2} \\ 0 \end{bmatrix}, & M^{\mathsf{T}} Q_{\Omega,\mathrm{v}} M &:= \begin{bmatrix} 0 & I_{n_2} \end{bmatrix} \mathsf{E}^{\mathsf{T}} \mathsf{Q}_\Omega \mathsf{E} \begin{bmatrix} 0 \\ I_{n_2} \end{bmatrix}, \end{aligned} \tag{4.22}$$

or, equivalently,

$$\mathsf{P}_\Omega = \begin{bmatrix} P_{\Omega,\mathrm{p}} & P_{\Omega,12} \\ P_{\Omega,12}^{\mathsf{T}} & P_{\Omega,\mathrm{v}} \end{bmatrix}, \quad \text{and} \quad \mathsf{E}^{\mathsf{T}}\mathsf{Q}_\Omega\mathsf{E} = \begin{bmatrix} J_{\mathrm{fc}}^{\mathsf{T}} Q_{\Omega,\mathrm{p}} J_{\mathrm{fc}} & J_{\mathrm{fc}}^{\mathsf{T}} Q_{\Omega,12} M \\ M^{\mathsf{T}} Q_{\Omega,12}^{\mathsf{T}} J_{\mathrm{fc}} & M^{\mathsf{T}} Q_{\Omega,\mathrm{v}} M \end{bmatrix},$$

with $P_{\Omega,\mathrm{p}}$, $P_{\Omega,\mathrm{v}}$ the *frequency-limited position and velocity controllability Gramians*, and $J_{\mathrm{fc}}^{\mathsf{T}} Q_{\Omega,\mathrm{p}} J_{\mathrm{fc}}$, $M^{\mathsf{T}} Q_{\Omega,\mathrm{v}} M$ the *frequency-limited position and velocity observability Gramians*. Remember that the matrices P_Ω and Q_Ω are given by (3.31) using the first companion form realization (2.18). As for the infinite second-order Gramians, one can observe that the frequency-limited position and velocity Gramians are symmetric positive semi-definite.

According to [90, 107, 159], the corresponding frequency-limited singular values are defined as follows.

Definition 4.5 (Second-order frequency-limited characteristic values):
Consider the second-order system (2.17) with the first-order realization (2.18) and the frequency range of interest $\Omega = -\Omega \subset \mathbb{R}$.

1. The positive square roots of the eigenvalues of $P_{\Omega,\mathrm{p}} J_{\mathrm{fc}}^{\mathsf{T}} Q_{\Omega,\mathrm{p}} J_{\mathrm{fc}}$ are the *frequency-limited position singular values* of (2.17).

2. The positive square roots of the eigenvalues of $P_{\Omega,\mathrm{p}} M^{\mathsf{T}} Q_{\Omega,\mathrm{v}} M$ are the *frequency-limited position-velocity singular values* of (2.17).

3. The positive square roots of the eigenvalues of $P_{\Omega,\mathrm{v}} J_{\mathrm{fc}}^{\mathsf{T}} Q_{\Omega,\mathrm{p}} J_{\mathrm{fc}}$ are the *frequency-limited velocity-position singular values* of (2.17).

Algorithm 4.3: Second-order frequency-limited balanced truncation square-root method.

Input: System matrices M, E, K, B_u, C_p, C_v from (2.17), frequency range of interest Ω.

Output: Matrices of the reduced-order system $\widehat{M}$, $\widehat{E}$, $\widehat{K}$, $\widehat{B}_\mathrm{u}$ $\widehat{C}_\mathrm{p}$, $\widehat{C}_\mathrm{v}$.

1 Compute Cholesky factorizations $\mathsf{P}_\Omega = \mathsf{R}_\Omega\mathsf{R}_\Omega^\mathsf{T}$, $\mathsf{Q}_\Omega = \mathsf{L}_\Omega\mathsf{L}_\Omega^\mathsf{T}$ of the solutions of the first-order frequency-limited Lyapunov equations (3.31), where the first companion form realization (2.18) is used.

2 Follow the Steps 2–4 in Algorithm 3.4.

4. The positive square roots of the eigenvalues of $P_{\Omega,\mathrm{v}}M^\mathsf{T}Q_{\Omega,\mathrm{v}}M$ are the *frequency-limited velocity singular values* of (2.17). ◊

Following the ideas of the first-order frequency-limited approach as well as the second-order balanced truncation method, the characteristic values in Definition 4.5 can be seen as a measure for the influence of the corresponding states to the input-to-output behavior of the system in the frequency range of interest. Currently, there is no supporting energy interpretation as for the classical first-order balanced truncation method available. But in practical implementations, the decay of the values in Definition 4.5 can be used to adaptively determine the reduced order.

Together with (4.22) and Definition 4.5, the resulting *second-order frequency-limited balanced truncation (SOFLBT)* square-root method is summarized in Algorithm 4.3.

Remark 4.6 (Stability issues of SOFLBT):
The SOFLBT method is in general not stability preserving. The same goes for the suggested approach in [107], which does neither necessarily yield a one-sided projection as claimed by the authors, nor may produce stable reduced-order second-order systems. Nevertheless, the general idea of the technique in [107] as well as a modified approach that are potentially advantageous in terms of stability preservation are discussed in Section 4.2.3. ◊

4.2.2 Structured time-limited approach

The idea of the second-order time-limited balanced truncation was first mentioned in [108]. Similarly to the frequency-limited case, the authors only considered two particular cases of the second-order balancing formulae. As in the previous section, the idea for the extension of the time-limited balanced truncation to second-order systems is to make use of writing the second-order Gramians as truncation of the first-order Gramians (4.21). Consequently, the infinite first-order Gramians in (4.21) are this time replaced by the first-order time-limited Gramians from (3.33) to define the second-order time-limited

Gramians

$$P_{\Theta,\mathrm{p}} := \begin{bmatrix} I_{n_2} & 0 \end{bmatrix} \mathsf{P}_\Theta \begin{bmatrix} I_{n_2} \\ 0 \end{bmatrix}, \qquad P_{\Theta,\mathrm{v}} := \begin{bmatrix} 0 & I_{n_2} \end{bmatrix} \mathsf{P}_\Theta \begin{bmatrix} 0 \\ I_{n_2} \end{bmatrix},$$
$$J_{\mathrm{fc}}^{\mathsf{T}} Q_{\Theta,\mathrm{p}} J_{\mathrm{fc}} := \begin{bmatrix} I_{n_2} & 0 \end{bmatrix} \mathsf{E}^{\mathsf{T}} \mathsf{Q}_\Theta \mathsf{E} \begin{bmatrix} I_{n_2} \\ 0 \end{bmatrix}, \quad M^{\mathsf{T}} Q_{\Theta,\mathrm{v}} M := \begin{bmatrix} 0 & I_{n_2} \end{bmatrix} \mathsf{E}^{\mathsf{T}} \mathsf{Q}_\Theta \mathsf{E} \begin{bmatrix} 0 \\ I_{n_2} \end{bmatrix},$$

or, equivalently,

$$\mathsf{P}_\Theta = \begin{bmatrix} P_{\Theta,\mathrm{p}} & P_{\Theta,12} \\ P_{\Theta,12}^{\mathsf{T}} & P_{\Theta,\mathrm{v}} \end{bmatrix}, \quad \text{and} \quad \mathsf{E}^{\mathsf{T}} \mathsf{Q}_\Theta \mathsf{E} = \begin{bmatrix} J_{\mathrm{fc}}^{\mathsf{T}} Q_{\Theta,\mathrm{p}} J_{\mathrm{fc}} & J_{\mathrm{fc}}^{\mathsf{T}} Q_{\Theta,12} M \\ M^{\mathsf{T}} Q_{\Theta,12}^{\mathsf{T}} J_{\mathrm{fc}} & M^{\mathsf{T}} Q_{\Theta,\mathrm{v}} M \end{bmatrix},$$

using the first companion form realization (2.18). Then, $P_{\Theta,\mathrm{p}}$ and $P_{\Theta,\mathrm{v}}$ are the *time-limited position and velocity controllability Gramians*, and $J_{\mathrm{fc}}^{\mathsf{T}} Q_{\Theta,\mathrm{p}} J_{\mathrm{fc}}$ and $M^{\mathsf{T}} Q_{\Theta,\mathrm{v}} M$ are the *time-limited position and velocity observability Gramians*. The two matrices P_Θ and Q_Θ are given via the time-limited dual Lyapunov equations (3.34) using the first companion form realization (2.18). Inherited from the first-order Gramians, also the second-order time-limited Gramians are all symmetric positive semi-definite. As pendant to the frequency-limited characteristic values from Definition 4.5, the following definition states the time-limited case.

Definition 4.7 (Second-order time-limited characteristic values):
Consider the second-order system (2.17) with the first-order realization (2.18) and the time range of interest $\Theta = [t_0, t_\mathrm{f}]$, $0 \leq t_0 < t_\mathrm{f}$.

1. The positive square roots of the eigenvalues of $P_{\Theta,\mathrm{p}} J_{\mathrm{fc}}^{\mathsf{T}} Q_{\Theta,\mathrm{p}} J_{\mathrm{fc}}$ are the *time-limited position singular values* of (2.17).

2. The positive square roots of the eigenvalues of $P_{\Theta,\mathrm{p}} M^{\mathsf{T}} Q_{\Theta,\mathrm{v}} M$ are the *time-limited position-velocity singular values* of (2.17).

3. The positive square roots of the eigenvalues of $P_{\Theta,\mathrm{v}} J_{\mathrm{fc}}^{\mathsf{T}} Q_{\Theta,\mathrm{p}} J_{\mathrm{fc}}$ are the *time-limited velocity-position singular values* of (2.17).

4. The positive square roots of the eigenvalues of $P_{\Theta,\mathrm{v}} M^{\mathsf{T}} Q_{\Theta,\mathrm{v}} M$ are the *time-limited velocity singular values* of (2.17). ◊

Similarly to the frequency-limited case, there is no energy interpretation for the characteristic values in Definition 4.7. But they are used in practical implementations as heuristics to determine the reduced order of the approximation.

As before, the resulting *second-order time-limited balanced truncation (SOTLBT)* methods can be obtained by replacing the infinite Gramians in the second-order balanced truncation method (Algorithm 3.4). This is summarized in Algorithm 4.4.

Algorithm 4.4: Second-order time-limited balanced truncation square-root method.

Input: System matrices M, E, K, B_{u}, C_{p}, C_{v} from (2.17), time range of interest Θ.

Output: Matrices of the reduced-order system $\widehat{M}$, $\widehat{E}$, $\widehat{K}$, $\widehat{B}_{\mathrm{u}}$ $\widehat{C}_{\mathrm{p}}$, $\widehat{C}_{\mathrm{v}}$.

1 Compute Cholesky factorizations $\mathsf{P}_\Theta = \mathsf{R}_\Theta \mathsf{R}_\Theta^{\mathsf{T}}$, $\mathsf{Q}_\Theta = \mathsf{L}_\Theta \mathsf{L}_\Theta^{\mathsf{T}}$ of the solutions of the first-order time-limited Lyapunov equations (3.34), where the first companion form realization (2.18) is used.

2 Follow the Steps 2–4 in Algorithm 3.4.

Remark 4.8 (Stability issues of SOTLBT):
In principle, stability preservation is no important property for time-limited model reduction methods. These techniques are supposed to approximate the system's behavior in a limited time range and are allowed to behave unstable outside this range. In some cases, it is nevertheless desired to preserve the stability of the original system in the reduced-order model. But as in the first-order case [130], there is no guarantee of stability preservation for the SOTLBT method. Also, the approach suggested in [108] is not capable to guarantee this. Another method that is potentially beneficial in terms of stability preservation and the idea from [108] are further discussed in the next section.◊

4.2.3 Mixed and modified Gramian methods

A drawback of the frequency- and time-limited balanced truncation methods in the first-order system case is the loss of stability preservation [90]. This holds as well for the second-order limited balanced truncation methods. Some modifications are known in the first-order system case to regain this property. However, these approaches cannot guarantee the preservation of stability for general second-order systems, since the original second-order balanced truncation method does not guarantee stability preservation in most cases [159]. But these modifications have the potential to produce a stable second-order reduced-order model in cases, in which the limited approaches failed to do so.

The first approach mentioned is the *mixed Gramian technique*. Therefore, one of the limited Gramians is replaced by their infinite counterpart such that the balanced truncation method is performed with one limited and one infinite Gramian [106, 117]. This idea directly translates to the second-order system case. One of the first-order Gramians in Algorithms 4.3 and 4.4 is replaced by the corresponding infinite first-order Gramian. This approach is suggested in [107, 108] but follows the misconception that the second-order balanced truncation is like the classical first-order version able to preserve stability. The mixed Gramian approach is in general not capable of preserving stability but might potentially work in some cases where the limited methods result in unstable

systems. In general, it is not clear which of the limited Gramians should be replaced in the approach to yield a good approximation in the limited time or frequency ranges. However, a good heuristic is the singular value decay of the limited Gramians since a faster decay indicates a better approximation using a smaller reduced order, i.e., the limited Gramian with the slower singular value decay could be replaced by the infinite Gramian.

A different technique, proposed in [102], are the modified Gramians. These replace the indefinite right-hand sides

$$\begin{aligned} \mathsf{B}_\Omega\mathsf{B}^\mathsf{T} + \mathsf{B}\mathsf{B}_\Omega^\mathsf{T} &= \widetilde{\mathsf{B}}\begin{bmatrix} 0 & I_m \\ I_m & 0 \end{bmatrix}\widetilde{\mathsf{B}}^\mathsf{T}, & \mathsf{C}_\Omega^\mathsf{T}\mathsf{C} + \mathsf{C}^\mathsf{T}\mathsf{C}_\Omega &= \widetilde{\mathsf{C}}^\mathsf{T}\begin{bmatrix} 0 & I_p \\ I_p & 0 \end{bmatrix}\widetilde{\mathsf{C}}, \\ \mathsf{B}_{t_0}\mathsf{B}_{t_0}^\mathsf{T} - \mathsf{B}_{t_\mathrm{f}}\mathsf{B}_{t_\mathrm{f}}^\mathsf{T} &= \breve{\mathsf{B}}\begin{bmatrix} I_m & 0 \\ 0 & -I_m \end{bmatrix}\breve{\mathsf{B}}^\mathsf{T}, & \mathsf{C}_{t_0}^\mathsf{T}\mathsf{C}_{t_0} - \mathsf{C}_{t_\mathrm{f}}^\mathsf{T}\mathsf{C}_{t_\mathrm{f}} &= \breve{\mathsf{C}}^\mathsf{T}\begin{bmatrix} I_p & 0 \\ 0 & -I_p \end{bmatrix}\breve{\mathsf{C}}, \end{aligned} \tag{4.23}$$

with $\widetilde{\mathsf{B}} = \begin{bmatrix} \mathsf{B}_\Omega & \mathsf{B} \end{bmatrix}$, $\widetilde{\mathsf{C}}^\mathsf{T} = \begin{bmatrix} \mathsf{C}_\Omega^\mathsf{T} & \mathsf{C}^\mathsf{T} \end{bmatrix}$, $\breve{\mathsf{B}} = \begin{bmatrix} \mathsf{B}_{t_0} & \mathsf{B}_{t_\mathrm{f}} \end{bmatrix}$ and $\breve{\mathsf{C}}^\mathsf{T} = \begin{bmatrix} \mathsf{C}_{t_0}^\mathsf{T} & \mathsf{C}_{t_\mathrm{f}}^\mathsf{T} \end{bmatrix}$, by definite right-hand sides. Using eigenvalue decompositions, the right-hand sides (4.23) can be rewritten as

$$\begin{aligned} \mathsf{B}_\Omega\mathsf{B}^\mathsf{T} + \mathsf{B}\mathsf{B}_\Omega^\mathsf{T} &= U_{\mathsf{B},\Omega}S_{\mathsf{B},\Omega}U_{\mathsf{B},\Omega}^\mathsf{T}, & \mathsf{C}_\Omega^\mathsf{T}\mathsf{C} + \mathsf{C}^\mathsf{T}\mathsf{C}_\Omega &= U_{\mathsf{C},\Omega}S_{\mathsf{C},\Omega}U_{\mathsf{C},\Omega}^\mathsf{T}, \\ \mathsf{B}_{t_0}\mathsf{B}_{t_0}^\mathsf{T} - \mathsf{B}_{t_\mathrm{f}}\mathsf{B}_{t_\mathrm{f}}^\mathsf{T} &= U_{\mathsf{B},\Theta}S_{\mathsf{B},\Theta}U_{\mathsf{B},\Theta}^\mathsf{T}, & \mathsf{C}_{t_0}^\mathsf{T}\mathsf{C}_{t_0} - \mathsf{C}_{t_\mathrm{f}}^\mathsf{T}\mathsf{C}_{t_\mathrm{f}} &= U_{\mathsf{C},\Theta}S_{\mathsf{C},\Theta}U_{\mathsf{C},\Theta}^\mathsf{T}, \end{aligned}$$

where $U_{\mathsf{B},\Omega}$, $U_{\mathsf{C},\Omega}$, $U_{\mathsf{B},\Theta}$, $U_{\mathsf{C},\Theta}$ are orthogonal matrices and

$$\begin{aligned} S_{\mathsf{B},\Omega} &= \operatorname{diag}(\eta_1^\mathsf{B}, \ldots, \eta_{2m}^\mathsf{B}, 0, \ldots, 0), & S_{\mathsf{C},\Omega} &= \operatorname{diag}(\eta_1^\mathsf{C}, \ldots, \eta_{2p}^\mathsf{C}, 0, \ldots, 0), \\ S_{\mathsf{B},\Theta} &= \operatorname{diag}(\mu_1^\mathsf{B}, \ldots, \mu_{2m}^\mathsf{B}, 0, \ldots, 0), & S_{\mathsf{C},\Theta} &= \operatorname{diag}(\mu_1^\mathsf{C}, \ldots, \mu_{2p}^\mathsf{C}, 0, \ldots, 0). \end{aligned}$$

Let $U_{\mathsf{B},\Omega,1}$, $U_{\mathsf{C},\Omega,1}$, $U_{\mathsf{B},\Theta,1}$, $U_{\mathsf{C},\Theta,1}$ be the parts of the orthogonal matrices, which correspond to the (potentially) non-zero eigenvalues. The *modified frequency- and time-limited Gramians* are then given via the solutions of the following Lyapunov equations

$$\begin{aligned} \mathsf{A}\mathsf{P}_{\Omega,\mathrm{mod}}\mathsf{E}^\mathsf{T} + \mathsf{E}\mathsf{P}_{\Omega,\mathrm{mod}}\mathsf{A}^\mathsf{T} + \mathsf{B}_{\Omega,\mathrm{mod}}\mathsf{B}_{\Omega,\mathrm{mod}}^\mathsf{T} &= 0, \\ \mathsf{A}^\mathsf{T}\mathsf{Q}_{\Omega,\mathrm{mod}}\mathsf{E} + \mathsf{E}^\mathsf{T}\mathsf{Q}_{\Omega,\mathrm{mod}}\mathsf{A} + \mathsf{C}_{\Omega,\mathrm{mod}}^\mathsf{T}\mathsf{C}_{\Omega,\mathrm{mod}} &= 0, \\ \mathsf{A}\mathsf{P}_{\Theta,\mathrm{mod}}\mathsf{E}^\mathsf{T} + \mathsf{E}\mathsf{P}_{\Theta,\mathrm{mod}}\mathsf{A}^\mathsf{T} + \mathsf{B}_{\Theta,\mathrm{mod}}\mathsf{B}_{\Theta,\mathrm{mod}}^\mathsf{T} &= 0, \\ \mathsf{A}^\mathsf{T}\mathsf{Q}_{\Theta,\mathrm{mod}}\mathsf{E} + \mathsf{E}^\mathsf{T}\mathsf{Q}_{\Theta,\mathrm{mod}}\mathsf{A} + \mathsf{C}_{\Theta,\mathrm{mod}}^\mathsf{T}\mathsf{C}_{\Theta,\mathrm{mod}} &= 0, \end{aligned} \tag{4.24}$$

with the definite right-hand sides

$$\begin{aligned} \mathsf{B}_{\Omega,\mathrm{mod}} &= U_{\mathsf{B},\Omega,1}\operatorname{diag}(|\eta_1^\mathsf{B}|^{\frac{1}{2}}, \ldots, |\eta_{2m}^\mathsf{B}|^{\frac{1}{2}}), & \mathsf{C}_{\Omega,\mathrm{mod}} &= \operatorname{diag}(|\eta_1^\mathsf{C}|^{\frac{1}{2}}, \ldots, |\eta_{2p}^\mathsf{C}|^{\frac{1}{2}})U_{\mathsf{C},\Omega,1}^\mathsf{T}, \\ \mathsf{B}_{\Theta,\mathrm{mod}} &= U_{\mathsf{B},\Theta,1}\operatorname{diag}(|\mu_1^\mathsf{B}|^{\frac{1}{2}}, \ldots, |\mu_{2m}^\mathsf{B}|^{\frac{1}{2}}), & \mathsf{C}_{\Theta,\mathrm{mod}} &= \operatorname{diag}(|\mu_1^\mathsf{C}|^{\frac{1}{2}}, \ldots, |\mu_{2p}^\mathsf{C}|^{\frac{1}{2}})U_{\mathsf{C},\Theta,1}^\mathsf{T}. \end{aligned} \tag{4.25}$$

Using these modified Gramians for the limited balanced truncation methods preserves the stability in reduced-order models in the first-order system case. Also, the modified

frequency-limited balanced truncation yields a (global) $\mathcal{H}_\infty$-error bound for first-order systems [47]. Note that for the solution of (4.24), still the matrix functions from the frequency- and time-limited Lyapunov equations (3.31) and (3.34) are needed to compute the new right-hand sides (4.25). For second-order versions of the modified Gramian methods, only the solutions of the frequency- and time-limited Lyapunov equations in Algorithms 4.3 and 4.4 need to be replaced. As for the mixed Gramian methods, the modified Gramians are not guaranteed to be stability preserving in the second-order system case but have the potential to produce stable reduced-order models when the fully limited methods failed to do so.

4.2.4 Numerical methods for the large-scale sparse systems case

In this section, numerical methods for applying Algorithms 4.3 and 4.4 to large-scale sparse second-order systems are discussed.

4.2.4.1 Matrix equation solvers for large-scale systems

A substantial part of the numerical effort in the computations of the second-order frequency- and time-limited balanced truncation methods goes into the solution of the arising matrix equations (3.31) and (3.34). In general, it was shown for the first-order case, that the singular values of the frequency- and time-limited Gramians are decaying possibly faster than for the infinite Gramians; see, e.g., [47] for the frequency-limited case. That leads to the natural approximation of the Gramians by low-rank factors, e.g.,

$$\mathsf{P}_\Omega \approx Z_{\mathsf{R}_\Omega} Z_{\mathsf{R}_\Omega}^\mathsf{T}, \quad \mathsf{P}_\Theta \approx Z_{\mathsf{R}_\Theta} Z_{\mathsf{R}_\Theta}^\mathsf{T}, \tag{4.26}$$

where $Z_{\mathsf{R}_\Omega} \in \mathbb{R}^{n_1 \times \ell_1}$, $Z_{\mathsf{R}_\Theta} \in \mathbb{R}^{n_1 \times \ell_2}$ and $\ell_1, \ell_2 \ll n_1$. These low-rank factors then replace the Cholesky factors in Algorithms 4.3 and 4.4.

The following three paragraphs will give a short inside into existing approaches for the solution of such large-scale sparse matrix equations and corresponding implementations.

Quadrature-based methods A natural approach based on the frequency and time domain integral representations of the limited Gramians (3.30) and (3.33) is the use of numerical integration formulae. As used for example in [107, 117], the low-rank factors of the Gramians can be computed by rewriting the full Gramians using quadrature formulae, for example, in the frequency-limited case

$$\begin{aligned}\mathsf{P}_\Omega &= \frac{1}{2\pi} \int_\Omega (\omega \mathsf{i}\mathsf{E} - \mathsf{A})^{-1} \mathsf{B}\mathsf{B}^\mathsf{T} (-\omega \mathsf{i}\mathsf{E} - \mathsf{A})^{-\mathsf{T}} \mathrm{d}\omega \\ &\approx \frac{1}{2\pi} \sum_{k=1}^{\ell} \gamma_k \Big((\omega_k \mathsf{i}\mathsf{E} - \mathsf{A})^{-1} \mathsf{B}\mathsf{B}^\mathsf{T} (-\omega_k \mathsf{i}\mathsf{E} - \mathsf{A})^{-\mathsf{T}} + (-\omega_k \mathsf{i}\mathsf{E} - \mathsf{A})^{-1} \mathsf{B}\mathsf{B}^\mathsf{T} (\omega_k \mathsf{i}\mathsf{E} - \mathsf{A})^{-\mathsf{T}} \Big),\end{aligned}$$

where γ_k are the weights and ω_k the evaluation points of an appropriate quadrature rule. This expression can be rewritten for the low-rank factors into

$$Z_{\mathsf{R}_\Omega} = \begin{bmatrix} \operatorname{Re}(\mathsf{B}_1) & \operatorname{Im}(\mathsf{B}_1) & \dots & \operatorname{Re}(\mathsf{B}_\ell) & \operatorname{Im}(\mathsf{B}_\ell) \end{bmatrix},$$

where $\mathsf{B}_k = \sqrt{\frac{\gamma_k}{\pi}}(\omega_k \mathsf{i}\mathsf{E} - \mathsf{A})^{-1}\mathsf{B}$. Note that this approach becomes impractical considering the time-limited case, since there, for each step of the quadrature rule, an approximation of the matrix exponential is needed. The empirical Gramians can be seen as a related approach, which uses simulations of the system to compute the time domain representation of the Gramians [110, 131].

A different approach was suggested in [47], which writes the right-hand sides of the frequency-limited Lyapunov equations (3.31) as integral expressions. In this way, the right-hand sides are first approximated and afterwards the large-scale matrix equations are solved, using one of the approaches in the following paragraphs. In principle, it is also possible to approximate the right-hand sides with matrix functions in (3.31) and (3.34) using the general quadrature approach from [109]. Currently, there is no stable, available implementation of quadrature-based matrix equation solvers for the frequency- and time-limited Lyapunov equations to be known. Therefore, the upcoming approaches will be rather used than the quadrature-based methods.

Low-rank ADI method The *low-rank alternating direction implicit (LR-ADI) method* [49, 136] is a well-established procedure for the solution of large-scale sparse Lyapunov equations via low-rank approximations. Originally developed for the Lyapunov equations corresponding to the infinite Gramians (3.28), the LR-ADI produces low-rank approximations of the form $Z_{\mathsf{R}_\infty,j} = \begin{bmatrix} Z_{\mathsf{R}_\infty,j-1} & \hat{\alpha}_j V_j \end{bmatrix}$ using the iteration scheme

$$V_j = (\mathsf{A} + \alpha_j \mathsf{E})^{-1} W_{j-1}, \quad W_j = W_{j-1} - 2\operatorname{Re}(\alpha_j) V_j,$$

with $\hat{\alpha}_j = \sqrt{-2\operatorname{Re}\alpha_j}$, $W_0 = \mathsf{B}$ and shifts $\alpha_j \in \mathbb{C}$; see [45–47, 129] for more details on this method.

The right-hand sides of the limited Lyapunov equations (3.31) and (3.34) can be rewritten in terms of LDL^T-factorizations as in (4.23). An extension of the LR-ADI method for LDL^T-factored right-hand sides is available by applying the same factorization type to the solution of the Lyapunov equations [132], e.g., in case of the frequency-limited controllability Gramian

$$\mathsf{P}_\Omega \approx Z_{\mathsf{R}_\Omega} Y_{\mathsf{R}_\Omega} Z_{\mathsf{R}_\Omega}^\mathsf{T}, \tag{4.27}$$

with low-rank factor $Z_{\mathsf{R}_\Omega} \in \mathbb{R}^{n_1 \times \ell_1}$ and symmetric center term $Y_{\mathsf{R}_\Omega} \in \mathbb{R}^{\ell_1 \times \ell_1}$. Since the limited Gramians are positive semi-definite, the three-term factorization in (4.27) can be reduced after converged iteration to a classical ZZ^T-type low-rank factorization (4.26).

For using the LR-ADI method to solve the large-scale matrix equations (3.31) and (3.34), an approximation of the matrix functions in the right-hand sides is needed beforehand. This could be done by methods from the previous or the next paragraph. It is noted in [47], that the information used for the approximation of the matrix functions cannot be re-used in the LR-ADI method. A stable implementation of the LR-ADI method in low-rank ZZ^T- and LDL^T-formats is available in [167].

Projection methods A class of methods that can be used to approximate the matrix functions in the right-hand sides of the limited Lyapunov equations, as well as to solve the large-scale matrix equations at the same time, are projection-based solvers. Thereby, low-dimensional subspaces $\operatorname{span}(V_k)$ are used to obtain the low-rank solutions of the large-scale matrix equations as solutions of projected small matrix equations. For example in case of (3.31), the solution to the first Lyapunov equation is given by $\mathsf{P}_\Omega \approx V_k \check{\mathsf{P}}_\Omega V_k^T$, where $\check{\mathsf{P}}_\Omega$ is the solution of the projected Lyapunov equation

$$\mathsf{T}_k \check{\mathsf{P}}_\Omega + \check{\mathsf{P}}_\Omega \mathsf{T}_k^T + \check{\mathsf{B}}_\Omega \check{\mathsf{B}}^T + \check{\mathsf{B}} \check{\mathsf{B}}_\Omega^T = 0, \tag{4.28}$$

where $\mathsf{T}_k = V_k^T \mathsf{E}^{-1} \mathsf{A} V_k$, $\check{\mathsf{B}}_\Omega = V_k^T \mathsf{E}^{-1} \mathsf{B}_\Omega$ and $\check{\mathsf{B}} = V_k^T \mathsf{E}^{-1} \mathsf{B}$ are the projected matrices from the large-scale Lyapunov equation (3.31). The equation (4.28) is now small and dense, and can be solved using established dense solvers. As one can observe, this method gives also the opportunity to approximate the matrix function in the right-hand side by the same low-dimensional subspace $\operatorname{span}(V_k)$. The projected right-hand side can then be computed using dense computation methods [109].

Usually, the low-dimensional subspace $\operatorname{span}(V_k)$ is constructed as standard [118], extended [176] or rational Krylov subspace [78], all of which can be efficiently computed for large-scale sparse systems. The implementation of the limited balanced truncation methods for second-order systems in [58] is also based on rational Krylov subspaces. The underlying theoretical algorithm and further details can be found in [47, Algorithm 4.1].

A drawback of the projection-based approach, especially for second-order systems, is that the projected system matrices T_k are not necessarily Hurwitz, i.e., they might have eigenvalues with nonnegative real parts. This can occur even if the original first- or second-order systems are asymptotically stable. In fact, quality and performance of the projection-based solvers strongly depend on the chosen first-order realization. Concerning second-order systems, projection methods are generally failing for at least one of the companion form realizations (2.18) and (2.19) due the occurring block structure. Therefore, in [57] it is suggested to use the strictly dissipative realization of second-order systems (2.22) from [151] for such computations. The advantages of this realization are that E is symmetric positive definite and $\mathsf{A} + \mathsf{A}^T$ symmetric negative definite in case of mechanical systems (M, E, K symmetric positive definite), and the same realization can be used for both dual Lyapunov equations without running into problems because of the block structure in the matrices. Following that, projection methods can preserve

the eigenvalue structure in the projected matrices T_k if the computations are made on the corresponding standard state-space realization, obtained by a symmetric state-space transformation. Using the Cholesky factorization $\mathsf{E} = \mathsf{L}\mathsf{L}^\mathsf{T}$, the projection methods should work implicitly on a realization of the form

$$\begin{aligned}\dot{\tilde{\mathsf{x}}}(t) &= \mathsf{L}^{-1}\mathsf{A}\mathsf{L}^{-\mathsf{T}}\tilde{\mathsf{x}}(t) + \mathsf{L}^{-1}\mathsf{B}u(t),\\ y(t) &= \mathsf{C}\mathsf{L}^{-\mathsf{T}}\tilde{\mathsf{x}}(t).\end{aligned}$$

By changing the first-order realization to (2.22), the computed solutions of the matrix equations change compared to the definition of the Gramians in the second-order balanced truncation methods. Consider for illustration the case of infinite Gramians. Given the two solutions $\widetilde{\mathsf{P}}_\infty$ and $\widetilde{\mathsf{Q}}_\infty$ of the Lyapunov equations (3.28) using the strictly dissipative realization (2.22), and let P_∞ and Q_∞ be the solutions of (3.28) with the first companion form realization (2.18). Then it holds

$$\mathsf{P}_\infty = T_{\text{fc2sd}}^\mathsf{T}\widetilde{\mathsf{P}}_\infty T_{\text{fc2sd}} = \widetilde{\mathsf{P}}_\infty \quad \text{and} \quad \mathsf{Q}_\infty = Z_{\text{fc2sd}}\widetilde{\mathsf{Q}}_\infty Z_{\text{fc2sd}}^\mathsf{T}, \tag{4.29}$$

with the transformation matrices from (2.23). The same transformation (4.29) can be used analogously to compute the solutions of the limited Lyapunov equations (3.31) and (3.34) with low-rank Gramian factors using the strictly dissipative realization (2.22).

4.2.4.2 Numerical stabilization and acceleration by second-order α-shifts

So far, it was always assumed that the second-order system (2.17) is asymptotically stable. In practice, the eigenvalues of $\lambda^2 M + \lambda E + K$ can be very close to the imaginary axis such that they behave numerically unstable, or they could be on the imaginary axis, e.g., in the case of marginal stability. This makes the usage of balancing-related model reduction methods and matrix equation solvers very difficult. A strategy to overcome these problems has been proposed in [86]. Therein, a frequency domain shift was used to move the spectrum of the pencil $\lambda\mathsf{E} - \mathsf{A}$, which had eigenvalues at zero, away from the imaginary axis to compute the system Gramians. This approach cannot be used exactly the same for the first-order realizations (2.18), (2.19), and (2.22) of second-order systems since it destroys the block structure used in the second-order balanced truncation methods as well as the block structure which is exploited in the numerical computations. Therefore, the concept of α-shifts needs to be transferred to second-order systems.

Remember the Laplace transformed second-order system (2.25) with the initial conditions $x_{\text{p},0} = x_{\text{v},0} = 0$. Now, let the Laplace variable be given by $s = \rho + \alpha$, with a shifted Laplace variable $\rho \in \mathbb{C}$ and a real, positive shift $\alpha \in \mathbb{R}_{>0}$. Then, the two equations

in (2.25) can be rewritten in terms of the shifted Laplace variable ρ such that

$$\begin{aligned}\Big((\rho+\alpha)^2M+(\rho+\alpha)E+K\Big)X(s) &= (\rho^2M+2\alpha\rho M+\alpha^2M+\rho E+\alpha E+K)X(s)\\ &= \Big(\rho^2M+\rho(E+2\alpha M)+(K+\alpha E+\alpha^2M)\Big)X(s)\\ &= (\rho^2M+\rho\widetilde{E}+\widetilde{K})X(s)\\ &= B_{\mathrm{u}}U(s)\end{aligned}$$

holds for the state equation, with $\widetilde{E}=E+2\alpha M$ and $\widetilde{K}=K+\alpha E+\alpha^2M$. For the output equation, it holds

$$\begin{aligned}Y(s) &= \Big((\rho+\alpha)C_{\mathrm{v}}+C_{\mathrm{p}}\Big)X(s)\\ &= \Big(\rho C_{\mathrm{v}}+(C_{\mathrm{p}}+\alpha C_{\mathrm{v}})\Big)X(s)\\ &= (\rho C_{\mathrm{v}}+\widetilde{C}_{\mathrm{p}})X(s),\end{aligned}$$

with $\widetilde{C}_{\mathrm{p}}=C_{\mathrm{p}}+\alpha C_{\mathrm{v}}$. The new system described by $(M,\widetilde{E},\widetilde{K},B_{\mathrm{u}},\widetilde{C}_{\mathrm{p}},C_{\mathrm{v}})$ has its spectrum shifted to the left by the constant α. This system is now used for the computation of the truncation matrices $W,V\in\mathbb{C}^{n_2\times r_2}$ for model reduction by projection (3.4). Then, the matrices of the reduced-order system $(\widehat{M},\widehat{\widetilde{E}},\widehat{\widetilde{K}},\widehat{B}_{\mathrm{u}},\widehat{\widetilde{C}}_{\mathrm{p}},\widehat{C}_{\mathrm{v}})$ yield the following additional relations

$$\widehat{\widetilde{E}}=\widehat{E}+2\alpha\widehat{M},\quad \widehat{\widetilde{K}}=\widehat{K}+\alpha\widehat{E}+\alpha^2\widehat{M},\quad \widehat{\widetilde{C}}_{\mathrm{p}}=\widehat{C}_{\mathrm{p}}+\alpha\widehat{C}_{\mathrm{v}},$$

where $\widehat{E}=W^{\mathsf{T}}EV$, $\widehat{K}=W^{\mathsf{T}}KV$ and $\widehat{C}_{\mathrm{p}}=C_{\mathrm{p}}V$ are the transformed matrices of the non-shifted second-order system. Assuming the reduced-order model to be written in frequency domain via the shifted Laplace variable ρ, it can be transformed back to a reduced second-order system using the original Laplace variable s. Using the substitution $\rho=s-\alpha$, the following two relations hold

$$\rho^2\widehat{M}+\rho\widehat{\widetilde{E}}+\widehat{\widetilde{K}}=s^2\widehat{M}+s\widehat{E}+\widehat{K}\quad\text{and}\quad \rho\widehat{C}_{\mathrm{v}}+\widehat{\widetilde{C}}_{\mathrm{p}}=s\widehat{C}_{\mathrm{v}}+\widehat{C}_{\mathrm{p}}.$$

The back-substitution gives the final reduced-order model to be $(\widehat{M},\widehat{E},\widehat{K},\widehat{B}_{\mathrm{u}},\widehat{C}_{\mathrm{p}},\widehat{C}_{\mathrm{v}})$. The α-shift strategy can be interpreted as a structured perturbation in the frequency domain during the computations. Experiments have shown that such an approach works fine for α small enough. It has to be noted that there are no theoretical results on the influence of the chosen α concerning the quality of the reduced-order model or properties like stability preservation and error bounds.

Remark 4.9 (Convergence behavior of numerical methods):
The α-shift approach can also be used to improve the behavior of numerical methods. In large-scale sparse matrix equation solvers, shifted linear systems with matrices of the

form $(\sigma^2 M + \sigma\widetilde{E} + \widetilde{K})$ need to be solved. Applying α-shifts can improve the conditioning of these systems since eigenvalues with smaller real parts are stronger influenced by the used shift compared to eigenvalues with larger real parts. Also, it can improve the convergence behavior of numerical methods by pushing the spectrum of the matrix pencil $\lambda^2 M + \lambda\widetilde{E} + \widetilde{K}$ further into the left open half-plane and away from the imaginary axis.$\Diamond$

The α-shift approach was used in [57] to apply the limited second-order balanced truncation methods in a numerical example with a system that has eigenvalues in zero. This technique will not further be investigated in the upcoming numerical experiments.

4.2.4.3 Two-step hybrid methods

The idea of two-step (or hybrid) model reduction methods has been used for quite some time in different applications [79, 135, 187]. In general, two-step methods are based on the division of the model reduction process into two phases. In the first step, a pre-reduction is computed by an efficient numerical procedure, which yields a very accurate approximation of the system's behavior. The model resulting from the pre-reduction is usually of medium-scale dimensions, to which the second reduction step using a more sophisticated model reduction method is applied. This procedure has the advantage that there is no necessity to solve complicated problems such as matrix equations in the large-scale sparse setting. Instead, one can use dense computation methods on the pre-reduced system usually avoiding the typical numerical problems as bad convergence behavior or the restriction to only using sparse operations.

In order to have an efficient structure-preserving pre-reduction method, the suggested approach is structured interpolation by rational Krylov subspaces (Sections 3.3.3 and 3.3.4). The pre-reduction is then computed via (3.4) with truncation matrices based on Proposition 3.2. As a small note here, large-scale sparse matrix equation solvers based on the solution of shifted linear systems (all methods in Section 4.2.4.1) are in fact equivalent to a two-step solution procedure using rational Krylov subspaces; see [187]. In general, the choice of interpolation points is crucial for the quality of the pre-reduced model. While there are strategies for an adaptive or optimal choice of these, it is usually enough to use as much sampling points as possible to be complex conjugate pairs on the imaginary axis, since the corresponding computations are rather cheap. A different problem that can occur in two-step methods is stability preservation in the pre-reduced model. In general, interpolation methods only preserve stability in special cases but not in general and might give an unstable pre-reduced model. However, this will not be further discussed here, since in the upcoming numerical experiments only mechanical systems are considered for which a one-sided projection ($W = V$) is enough to preserve stability.

Remark 4.10 (Pre-reduction in the frequency-limited case):
For the pre-reduction via interpolation for the frequency-limited balanced truncation method, a natural choice of interpolation points would be to sample locally in $\Omega\mathrm{i}$ instead of aiming for a global approximation. In this case, the resulting frequency-limited balanced truncation will very likely not give the same results as the large-scale approach that works with the original system matrices. This observation comes from the fact, that the frequency-limited balanced truncation still takes information about the complete system structure into account and the pre-reduced system can be completely different from the original one, while being rather accurate in the frequency region of interest. It is not known, which type of pre-reductions, local or global, performs better at the end.$\Diamond$

Due to the required accuracy of the pre-reduced model, its order can be still comparably large. Therefore, an efficient iterative solver for the Lyapunov equations appearing in the second reduction step is suggested. In general, the following stable Lyapunov equations are considered

$$\begin{aligned} \mathsf{A}X_1\mathsf{E}^\mathsf{T} + \mathsf{E}X_1\mathsf{A}^\mathsf{T} + \mathsf{B}\mathsf{Q}\mathsf{B}^\mathsf{T} &= 0, \\ \mathsf{A}^\mathsf{T}X_2\mathsf{E} + \mathsf{E}^\mathsf{T}X_2\mathsf{A} + \mathsf{C}^\mathsf{T}\mathsf{R}\mathsf{C} &= 0, \end{aligned} \tag{4.30}$$

with suitable $\mathsf{E}, \mathsf{A}, \mathsf{B}, \mathsf{C}$ as in (2.8) and symmetric (possibly indefinite) matrices $\mathsf{Q} \in \mathbb{R}^{m\times m}$ and $\mathsf{R} \in \mathbb{R}^{p\times p}$. The solutions of (4.30) can then be factored in the same way as the right-hand sides, i.e., $X_1 = Z_1Y_1Z_1^\mathsf{T}$ and $X_2 = Z_2Y_2Z_2^\mathsf{T}$, with Y_1 and Y_2 symmetric matrices. For efficiently computing the solutions of (4.30), the dual sign function iteration method from [33] is extended to the LDL^T-factorization of the solutions. As a result, a sign function iteration that solves both Lyapunov equations with symmetric indefinite right-hand sides (4.30) at the same time is presented in Algorithm 4.5.

An implementation of Algorithm 4.5 as well as dense versions of the second-order frequency- and time-limited balanced truncation methods can be found in [55].

Remark 4.11 (Compression of solution factors):
In Step 4 of Algorithm 4.5, the memory requirements as well as the number of operations for the next iteration step are doubling due to the concatenation of the solution factors. It is recommended to do LDL^T column and row compressions in that step to keep the size of the factors reasonably small. For example, consider the solution factors corresponding to X_1 in the k-th iteration step, i.e., the product $B_{k+1}Q_{k+1}B_{k+1}^\mathsf{T}$. Computing a QR decomposition followed by an eigenvalue decomposition such that

$$B_{k+1} = VR \quad \text{and} \quad RQ_{k+1}R^\mathsf{T} = U\Sigma U^\mathsf{T}$$

hold, with V, U orthogonal matrices and Σ a diagonal matrix with the eigenvalues, allows the approximation of the iteration factors in the following way. Let Σ_1 contain the

Algorithm 4.5: LDL^{T}-factored sign function dual Lyapunov equation solver.

Input: A, B, C, E, Q, R from (4.30), convergence tolerance τ.

Output: Z_1, Y_1, Z_2, Y_2 – solution factors of (4.30).

1 Set $A_1 = \mathsf{A}$, $B_1 = \mathsf{B}$, $Q_1 = \mathsf{Q}$, $C_1 = \mathsf{C}$, $R_1 = \mathsf{R}$, $k = 1$.

2 **while** $\|A_k + \mathsf{E}\| > \tau \|\mathsf{E}\|$ **do**

3 Compute the scaling factor for convergence acceleration $c_k = \sqrt{\frac{\|A_k\|_F}{\|\mathsf{E}A_k^{-1}\mathsf{E}\|_F}}$.

4 Compute the next iterates of the solution factors

$$B_{k+1} = \begin{bmatrix} B_k & \mathsf{E}A_k^{-1}B_k \end{bmatrix}, \quad Q_{k+1} = \begin{bmatrix} \frac{1}{2c_k}Q_k & \\ & \frac{c_k}{2}Q_k \end{bmatrix},$$
$$C_{k+1} = \begin{bmatrix} C_k \\ A_k^{-1}\mathsf{E}C_k \end{bmatrix}, \qquad R_{k+1} = \begin{bmatrix} \frac{1}{2c_k}R_k & \\ & \frac{c_k}{2}R_k \end{bmatrix}.$$

5 Compute the next iteration matrix

$$A_{k+1} = \frac{1}{2c_k}A_k + \frac{c_k}{2}\mathsf{E}A_k^{-1}\mathsf{E}.$$

6 Set $k = k + 1$.

7 Construct the final solution factors

$$Z_1 = \frac{1}{\sqrt{2}}\mathsf{E}^{-1}B_k, \quad Y_1 = Q_k, \quad Z_2 = \frac{1}{\sqrt{2}}\mathsf{E}^{-\mathsf{T}}C_k^{\mathsf{T}}, \quad Y_2 = R_k.$$

eigenvalues from Σ with the largest absolute values and U_1 the corresponding orthogonal eigenvectors, then

$$B_{k+1}Q_{k+1}B_{k+1}^{\mathsf{T}} \approx (VU_1)\Sigma_1(VU_1)^{\mathsf{T}},$$

such that B_{k+1} can be replaced by (VU_1) and Q_{k+1} by Σ_1 in the following iteration step of Algorithm 4.5. ◊

4.2.5 Numerical experiments

Different numerical experiments employing the frequency- and time-limited second-order balanced truncation methods can be found in [26, 27, 57, 168]. In these publications, the limited approximation quality between the different balancing formulae is the main focus. In this section, the limited second-order balanced truncation methods will be compared in numerical experiments with their global pendants as well as with the mixed

and modified Gramian approaches from Section 4.2.3. The methods for the comparisons will be denoted as follows:

SOBT(*p*/*pm*/*pv*/*vp*/*vpm*/*v*/*fv*/*so*) is the second-order balanced truncation method (Section 3.4.3) with the balancing formulae from Table 3.1,

SOFLBT(*p*/*pm*/*pv*/*vp*/*vpm*/*v*/*fv*/*so*) is the second-order frequency-limited balanced truncation method (Section 4.2.1) with the balancing formulae from Table 3.1,

SOMFLBT(*p*/*pm*/*pv*/*vp*/*vpm*/*v*/*fv*/*so*) is the modified second-order frequency-limited balanced truncation method (Section 4.2.3) with the balancing formulae from Table 3.1,

SOFLBTC(*p*/*pm*/*pv*/*vp*/*vpm*/*v*/*fv*/*so*) is the mixed second-order frequency-limited balanced truncation method (Section 4.2.3) using the infinite controllability Gramians with the balancing formulae from Table 3.1,

SOFLBTO(*p*/*pm*/*pv*/*vp*/*vpm*/*v*/*fv*/*so*) is the mixed second-order frequency-limited balanced truncation method (Section 4.2.3) using the infinite observability Gramians with the balancing formulae from Table 3.1,

SOTLBT(*p*/*pm*/*pv*/*vp*/*vpm*/*v*/*fv*/*so*) is the second-order time-limited balanced truncation method (Section 4.2.2) with the balancing formulae from Table 3.1,

SOMTLBT(*p*/*pm*/*pv*/*vp*/*vpm*/*v*/*fv*/*so*) is the modified second-order time-limited balanced truncation method (Section 4.2.3) with the balancing formulae from Table 3.1,

SOTLBTC(*p*/*pm*/*pv*/*vp*/*vpm*/*v*/*fv*/*so*) is the mixed second-order time-limited balanced truncation method using the infinite controllability Gramians (Section 4.2.3) with the balancing formulae from Table 3.1,

SOTLBTO(*p*/*pm*/*pv*/*vp*/*vpm*/*v*/*fv*/*so*) is the mixed second-order time-limited balanced truncation method using the infinite observability Gramians (Section 4.2.3) with the balancing formulae from Table 3.1.

As the limited model reduction methods are supposed to approximate the system's behavior in restricted intervals, limited versions of the approximate norms (2.44)–(2.46) will be used to compute the MORscores. Therefore, the norms will be restricted to the approximation ranges of interest in either frequency or time domain. For the notation, superscripts Ω and Θ are added to the current norm notation such that L_2^Θ, L_∞^Θ, and $\mathcal{H}_\infty^\Omega$ denote the local relative errors. For more detailed discussions, some methods will be considered for fixed reduced orders in frequency and time domain. Therefore, the same pointwise relative errors (4.19) and (4.20) as in Section 4.1.5 will be used.

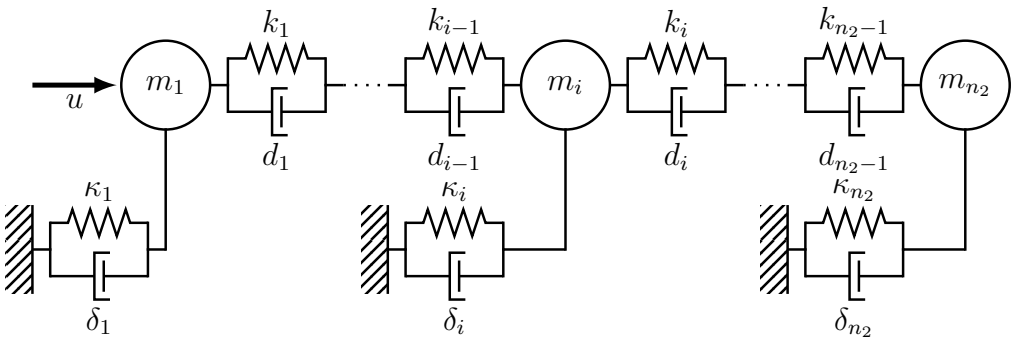


Figure 4.9: Sketch of the single chain oscillator example.

4.2.5.1 Single chain oscillator

The damped single chain oscillator benchmark was used in [142] with a holonomic constraint to test the first-order balanced truncation method for descriptor systems. As test example for the second-order frequency- and time-limited balanced truncation methods, the holonomic constraint was removed. The resulting damped mass-spring system can be seen in Figure 4.9. The system parameters are set exactly as in [142], with

$$\begin{aligned} k_1 = \ldots = k_{n_2-1} = \kappa_2 = \ldots = \kappa_{n_2-1} = 2, \quad \kappa_1 = \kappa_{n_2} = 4, \\ d_1 = \ldots = d_{n_2-1} = \delta_2 = \ldots = \delta_{n_2-1} = 5, \quad \kappa_1 = \kappa_{n_2} = 10, \end{aligned}$$

for stiffness and damping coefficients, and $m_1 = \ldots m_{n_2} = 100$ for the masses. The number of masses in the system is set to $n_2 = 10\,000$ for the following experiments. The input matrix is designed such that the first and last five masses are excited by the same input, and the outputs such that the summed displacement of the first three, eighth til tenth and the last three masses can be observed, i.e.,

$$B_{\mathrm{u}} = \begin{bmatrix} \mathbb{1}_5 \\ 0 \\ \vdots \\ 0 \\ \mathbb{1}_5 \end{bmatrix}, \quad C_{\mathrm{p}} = \begin{bmatrix} e_1 + e_2 + e_3 & e_8 + e_9 + e_{10} & e_{n_2-2} + e_{n_2-1} + e_{n_2} \end{bmatrix}^{\mathsf{T}},$$

where $\mathbb{1}_n$ is the vector of length n containing only ones and e_j is the j-th column of the n_2-dimensional identity matrix. In the experiments with the single chain oscillator, the computations were done directly on the large-scale sparse system using the projection-based matrix equation solvers from [58] and the LR-ADI method from [167].

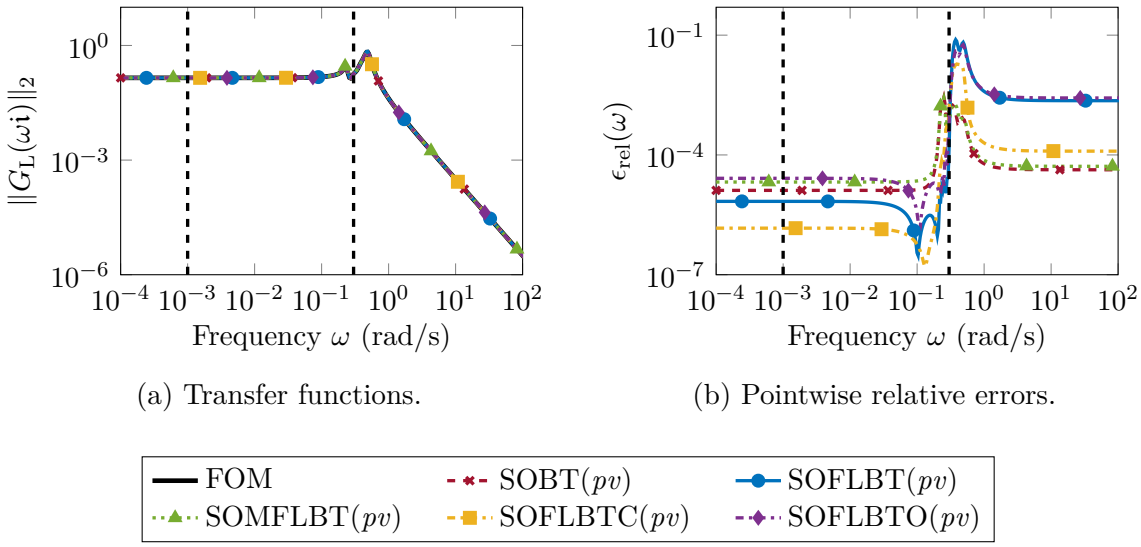
Frequency-limited methods The frequency-limited methods are considered with the frequency range of interest $[10^{-3}, 3 \cdot 10^{-1}]$ rad/s. The resulting MORscores of all computed

Table 4.3: MORscores of the classical and frequency-limited second-order balanced truncation for the single chain oscillator example with reduced orders from 1 to 40, and the percentage of stable reduced-order models.

Method	$\mathcal{H}_\infty$	$\mathcal{H}_\infty^\Omega$	Stab. ratio
SOBT(p)	0.2621	0.2430	1.0000
SOBT(pm)	0.2619	0.2422	1.0000
SOBT(pv)	0.2602	0.2480	1.0000
SOBT(vp)	0.2544	0.2333	0.9250
SOBT(vpm)	0.2595	0.2373	1.0000
SOBT(v)	0.2620	0.2422	1.0000
SOBT(fv)	0.1991	0.1867	1.0000
SOBT(so)	0.2623	0.2428	1.0000
SOFLBT(p)	0.0835	0.3912	0.9500
SOFLBT(pm)	0.0861	0.3949	0.9500
SOFLBT(pv)	0.0845	0.3971	1.0000
SOFLBT(vp)	0.0776	0.3905	0.9250
SOFLBT(vpm)	0.0814	0.3827	1.0000
SOFLBT(v)	0.0785	0.3896	0.9500
SOFLBT(fv)	0.0649	0.2742	1.0000
SOFLBT(so)	0.0799	0.3860	0.9750

model reduction methods can be found in Tables 4.3 and 4.4. First, compare the classical second-order balanced truncation and the frequency-limited variant in Table 4.3. The SOFLBT methods behave exactly as expected. Their global approximation behavior is very poor as indicated by the very small MORscores, but their local approximation quality is much better than that of the classical second-order balanced truncation since these MORscores are nearly twice as large. Concerning the amount of stable reduced-order models, the global method always produced stable models except for the *vp* formula. This is not true for the limited approach anymore. There, several formulae produced some unstable reduced-order models. On the other hand for the *vp* formula, the frequency-limited approach has exactly the same percentage of stable reduced-order models as the global approach.

The alternative techniques with potential stability preservation are shown in Table 4.4. The modified methods behave very similar to their global counterparts with only marginally better approximations in the frequency range of interest. Also, the stability ratio column of the SOMFLBT methods has exactly the same pattern as for SOBT, where for the *vp* formula even less stable reduced-order models were computed. On the



(a) Transfer functions.

(b) Pointwise relative errors.

Figure 4.10: Frequency domain results of the frequency-limited methods for the singlechain oscillator example.

other hand, the mixed Gramian methods seem to be very promising. Independent of the chosen Gramian to be exchanged by its infinite version, the local approximations yield good results. The version using the infinite observability Gramian performs better than using the infinite controllability Gramian, and is also very close to the performance of the fully limited methods. This observation coincides with the heuristic to keep the frequency-limited Gramian with smaller rank, since the frequency-limited controllability Gramian has rank 52 whereas the limited observability Gramian is of rank 144.

For a closer look at the frequency domain behavior of the reduced-order models, the reduced order $r_2 = 14$ was chosen. Comparing all the MORscores, the position-velocity balancing is overall very well performing and, therefore, chosen as representative for all different model reduction techniques. Transfer functions and pointwise relative approximation errors are shown in Figure 4.10. The frequency range of interest is depicted between the dashed vertical lines. The behavior of the methods for low frequencies is a bit different than one would expect from the discussion and MORscores before. Here, the mixed Gramian method using the infinite controllability Gramian performs better than the fully limited approach, and also SOBT(pv) has a smaller relative error than the modified and the other mixed Gramian method. This behavior changes close to the right border of the frequency range of interest. Here, the errors of SOBT(pv) and SOMFLBT(pv) shoot up due to the changing behavior of the transfer function, while the errors of the other limited methods increase at a slower rate.

Table 4.4: MORscores of the modified and mixed second-order frequency-limited balanced truncation for the single chain oscillator example with reduced orders from 1 to 40, and the percentage of stable reduced-order models.

Method	$\mathcal{H}_\infty$	$\mathcal{H}_\infty^\Omega$	Stab. ratio
SOMFLBT(p)	0.2530	0.2539	1.0000
SOMFLBT(pm)	0.2541	0.2530	1.0000
SOMFLBT(pv)	0.2443	0.2599	1.0000
SOMFLBT(vp)	0.2506	0.2454	0.9000
SOMFLBT(vpm)	0.2584	0.2489	1.0000
SOMFLBT(v)	0.2535	0.2530	1.0000
SOMFLBT(fv)	0.1902	0.1946	1.0000
SOMFLBT(so)	0.2534	0.2537	1.0000
SOFLBTC(p)	0.1832	0.3062	0.9500
SOFLBTC(pm)	0.1876	0.3093	1.0000
SOFLBTC(pv)	0.1802	0.3132	1.0000
SOFLBTC(vp)	0.1820	0.2957	0.9250
SOFLBTC(vpm)	0.1933	0.3048	1.0000
SOFLBTC(v)	0.1847	0.3092	1.0000
SOFLBTC(fv)	0.1605	0.1982	1.0000
SOFLBTC(so)	0.1846	0.3105	0.9750
SOFLBTO(p)	0.1086	0.3688	0.9500
SOFLBTO(pm)	0.1144	0.3727	1.0000
SOFLBTO(pv)	0.1096	0.3743	1.0000
SOFLBTO(vp)	0.1071	0.3746	1.0000
SOFLBTO(vpm)	0.1116	0.3745	1.0000
SOFLBTO(v)	0.1063	0.3765	1.0000
SOFLBTO(fv)	0.0676	0.2656	1.0000
SOFLBTO(so)	0.1064	0.3756	0.9500

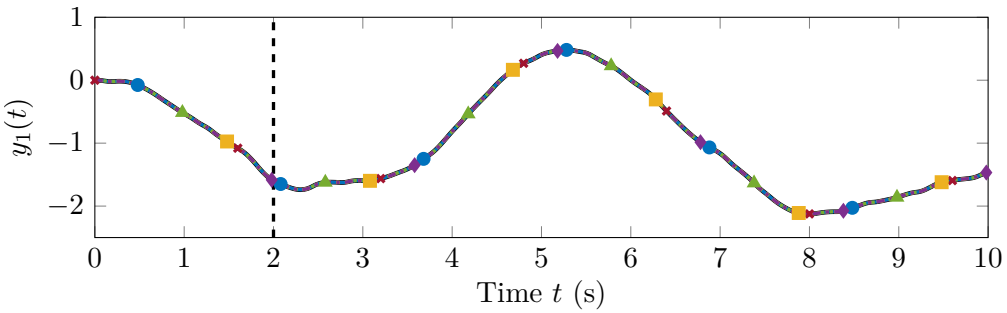
Table 4.5: MORscores of the classical and time-limited second-order balanced truncation for the single chain oscillator example with reduced orders from 1 to 40, and the percentage of stable reduced-order models.

Method	L_2	L_2^Θ	L_∞	L_∞^Θ	Stab. ratio
SOBT(p)	0.3568	0.3586	0.3567	0.3602	1.0000
SOBT(pm)	0.3640	0.3676	0.3634	0.3690	1.0000
SOBT(pv)	0.3456	0.3469	0.3454	0.3483	1.0000
SOBT(vp)	0.3425	0.3509	0.3418	0.3523	0.9250
SOBT(vpm)	0.3764	0.3853	0.3743	0.3859	1.0000
SOBT(v)	0.3633	0.3669	0.3632	0.3682	1.0000
SOBT(fv)	0.2733	0.2748	0.2755	0.2778	1.0000
SOBT(so)	0.3589	0.3606	0.3586	0.3620	1.0000
SOTLBT(p)	0.4556	0.8872	0.4288	0.8886	0.9500
SOTLBT(pm)	0.4616	0.8739	0.4409	0.8828	0.9750
SOTLBT(pv)	0.4612	0.8520	0.4391	0.8563	0.9750
SOTLBT(vp)	0.5283	0.8917	0.4985	0.8943	0.9750
SOTLBT(vpm)	0.5282	0.8684	0.4993	0.8752	0.9750
SOTLBT(v)	0.5281	0.8889	0.4982	0.8945	0.9750
SOTLBT(fv)	0.2588	0.6432	0.2366	0.6507	1.0000
SOTLBT(so)	0.4715	0.8698	0.4507	0.8740	0.9500

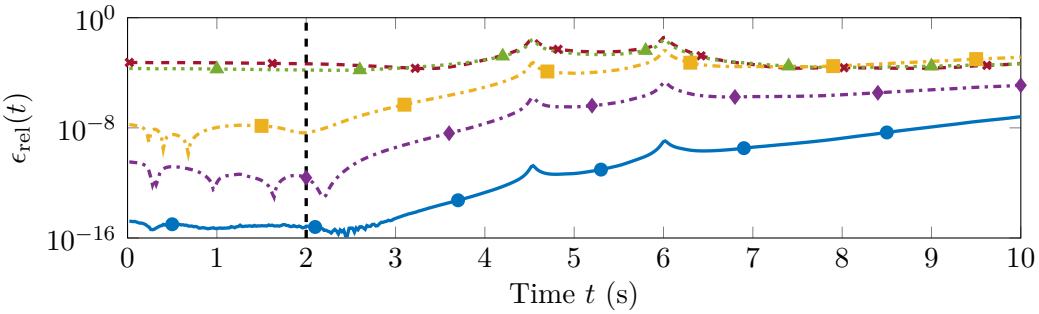
Time-limited methods Next, the time-limited approaches are considered. The time interval for the full simulation is set to be $[0, 10]$ s and the smaller time range for the limited model reduction is $[0, 2]$ s. The results in terms of MORscores are given in Tables 4.5 and 4.6. For the simulations, the input signal

$$u(t) = 100 \cdot \eta(t_j), \quad \text{for } t_j \leq t < t_{j+1},$$

was used, with $j = 0, \ldots, 99$, equidistant time steps $t_j = j \cdot \frac{10}{99}$ and presampled Gaussian white noise $\eta(t)$. Comparing the classical and unmodified time-limited methods in Table 4.5, the second-order time-limited balanced truncation methods have an overwhelmingly high MORscore in the time range of interest which is more than twice as large as the MORscores of the global methods. An interesting side effect that will be discussed later in more detail are the larger MORscores of the limited methods in the global norms. As mentioned in Remark 4.8, the time-limited balanced truncation is used to approximate the time domain behavior of the system only in a limited range and can be unstable otherwise. This effect is only indicated by the lower percentage of stable reduced-order models in case of SOTLBT.



(a) First output entry $y_1(t)$ of the time simulation.



(b) Pointwise relative errors of the complete output.

Figure 4.11: Time domain results of the time-limited methods for the single chain oscillator example.

Looking at Table 4.6 for the stabilization ideas, the modified and mixed Gramian approaches were often able to increase the number of stable reduced-order models. For nearly all methods, all computed reduced-order models were stable. Concerning the MORscores, similar relations as for the frequency-limited methods can be observed. The modified Gramian methods are only marginally better in the local approximation than SOBT and the mixed Gramian approaches perform still very well in the local approximation. The difference in the ranks of the time-limited Gramian factors is very small as the limited controllability Gramian has rank 10 and the limited observability Gramian rank 28 such that it is not surprising that both of the mixed Gramian approaches give compatible results. The MORscores reveal SOTLBTO to be usually better in the local approximation, while SOTLBTC gives results similar to SOBT and SOTLBT in the global norms.

Figure 4.11 shows the time-limited approaches with the *vpm* formula, choosing the reduced order $r_2 = 7$. The limited methods perform exactly as indicated by the MORscores with SOTLBT best followed by the mixed and then the modified Gramian methods. The interesting effect already seen in the global MORscores of Table 4.5 is visible here again. In the time simulation, the approximation quality of later time steps strongly depends on previous ones. The time-limited methods are very accurate in the beginning of the simulation and do not stop to approximate the full-order system right at the end of the considered time interval. Therefore, their errors are slowly diverging from the original system's simulation behavior after the considered time range of interest ended. At some point they will have larger simulation errors than SOBT as it can be already seen for SOTLBTC. But due to the length of the full time interval, they perform still better than the global approximations.

Table 4.6: MORscores of the modified and mixed second-order time-limited balanced truncation for the single chain oscillator example with reduced orders from 1 to 40, and the percentage of stable reduced-order models.

Method	L_2	L_2^Θ	L_∞	L_∞^Θ	Stab. ratio
SOMTLBT(p)	0.3578	0.3603	0.3573	0.3617	1.0000
SOMTLBT(pm)	0.3626	0.3672	0.3616	0.3680	1.0000
SOMTLBT(pv)	0.3501	0.3515	0.3496	0.3530	1.0000
SOMTLBT(vp)	0.3354	0.3421	0.3340	0.3438	0.9000
SOMTLBT(vpm)	0.3750	0.3900	0.3730	0.3917	1.0000
SOMTLBT(v)	0.3648	0.3688	0.3639	0.3700	1.0000
SOMTLBT(fv)	0.2739	0.2773	0.2763	0.2803	1.0000
SOMTLBT(so)	0.3596	0.3617	0.3590	0.3629	1.0000
SOTLBTC(p)	0.3813	0.5707	0.3693	0.5751	0.8500
SOTLBTC(pm)	0.4021	0.7027	0.3882	0.7033	1.0000
SOTLBTC(pv)	0.3882	0.5754	0.3752	0.5794	1.0000
SOTLBTC(vp)	0.3910	0.5815	0.3795	0.5857	0.7250
SOTLBTC(vpm)	0.4169	0.7172	0.4038	0.7206	1.0000
SOTLBTC(v)	0.4106	0.6114	0.3986	0.6154	1.0000
SOTLBTC(fv)	0.2616	0.2645	0.2630	0.2665	1.0000
SOTLBTC(so)	0.4004	0.6108	0.3881	0.6148	1.0000
SOTLBTO(p)	0.2583	0.6354	0.2422	0.6481	1.0000
SOTLBTO(pm)	0.2791	0.6596	0.2583	0.6667	1.0000
SOTLBTO(pv)	0.2757	0.6542	0.2547	0.6610	1.0000
SOTLBTO(vp)	0.3169	0.7029	0.2976	0.7095	1.0000
SOTLBTO(vpm)	0.3173	0.7086	0.2939	0.7174	1.0000
SOTLBTO(v)	0.3169	0.7029	0.2975	0.7095	1.0000
SOTLBTO(fv)	0.2590	0.6431	0.2367	0.6506	1.0000
SOTLBTO(so)	0.2774	0.6571	0.2562	0.6646	1.0000

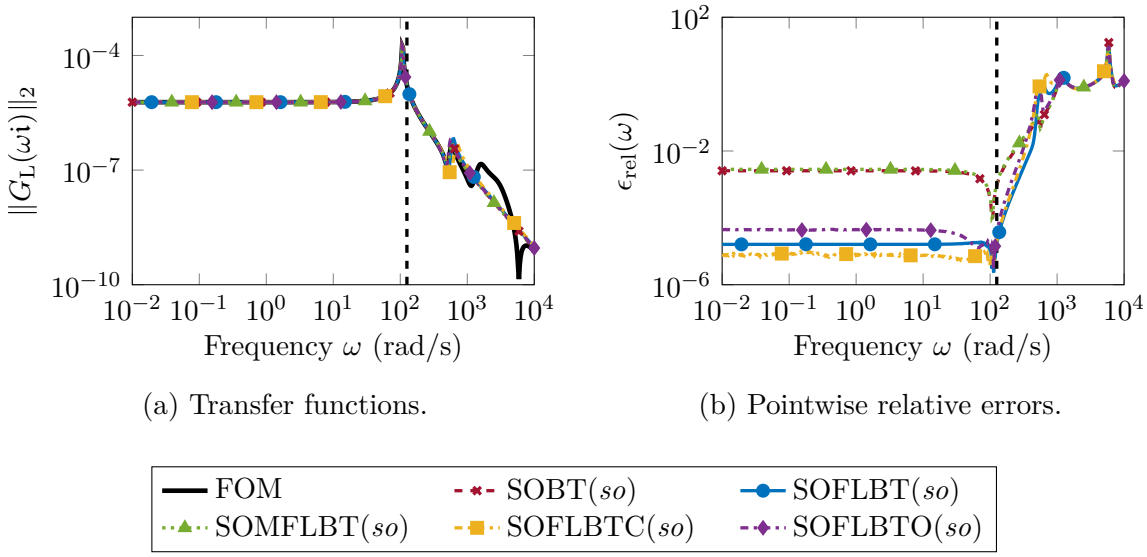
Table 4.7: MORscores of the (hybrid) classical and frequency-limited second-order balanced truncation for the artificial fishtail example with reduced orders from 1 to 10, and the percentage of stable reduced-order models.

Method	$\mathcal{H}_\infty$	$\mathcal{H}_\infty^\Omega$	Stab. ratio
SOBT(p)	0.2610	0.2770	0.9000
SOBT(pm)	0.1795	0.1795	0.5000
SOBT(pv)	0.2609	0.2772	0.9000
SOBT(vp)	0.2142	0.2674	0.6000
SOBT(vpm)	0.0884	0.0900	0.2000
SOBT(v)	0.2606	0.2778	1.0000
SOBT(fv)	0.2168	0.2338	1.0000
SOBT(so)	0.2610	0.2770	1.0000
SOFLBT(p)	0.1765	0.2876	0.6000
SOFLBT(pm)	0.1417	0.2521	0.2000
SOFLBT(pv)	0.1959	0.2876	0.8000
SOFLBT(vp)	0.2172	0.2816	0.9000
SOFLBT(vpm)	0.1206	0.1910	0.3000
SOFLBT(v)	0.2150	0.2832	0.7000
SOFLBT(fv)	0.1386	0.2404	1.0000
SOFLBT(so)	0.1876	0.2887	0.7000

4.2.5.2 Artificial fishtail model

As second numerical example, the artificial fishtail model from Section 1.3.2 is considered. For this example, the structure-preserving balanced truncation methods are applied as two-step approaches (cf. Section 4.2.4.3). A structured interpolation was computed as pre-reduction using 200 logarithmically equidistant interpolation points in complex conjugate pairs in the frequency range $[10^{-2}, 10^{4}]$ rad/s. Employing Parts (a) and (b) from Proposition 3.2, and basis concatenation to use only a one-sided projection results in a stable intermediate second-order system of order 1 600. The medium-scale dense implementations of the classical and limited second-order balanced truncation methods from [55] were then used. Results for the artificial fishtail model with the limited balanced truncation methods directly employed on the large-scale sparse system can be found in [57].

Frequency-limited methods From a practical point of view, the artificial fishtail cannot be operated at higher frequencies than 20 Hz. While it would make sense to consider



(a) Transfer functions.

(b) Pointwise relative errors.

Figure 4.12: Frequency domain results of the frequency-limited methods for the artificial fishtail example.

the frequency interval to begin at zero, this leads to unstable numerical behavior in computations due to the inversion of the mass or system matrix in the matrix logarithm. Therefore, the lower bound of the globally considered frequency range is taken leading the frequency range of interest to be $[10^{-2}, 2\pi \cdot 20]$ rad/s. The resulting MORscores of the applied methods can be found in Tables 4.7 and 4.8. First, one can observe the impact of the pre-reduction comparing the SOBT entries from Table 4.7 with those of Table 4.2. Beside small disturbances in the MORscores, there are more unstable reduced-order models in the two-step case. For the fully frequency-limited reduced-order models in Table 4.7, all MORscores in the limited norm are larger than for SOBT. This comes with the cost that less stable reduced-order models were computed by SOFLBT than by SOBT, except for the *vp* and *vpm* formulae.

Figure 4.12 shows the frequency-limited results in comparison with the global approaches using the *so* formula and the reduced order $r_2 = 2$. Here, the limited methods perform exceptionally well with several orders of magnitude better than SOBT, and SOFLBT and SOFLBTC as clear winners. The reason for the small difference in the MORscores is that SOFLBT and SOFLBTC already reached their smallest possible approximation accuracy in the frequency range of interest. Due to the bad conditioning of the example data, it is not possible to further reduce the error in this region. In fact, the rest of the methods also converge to this error level at latest with $r_2 = 4$ and stay there for all larger reduced-order models that were computed for the MORscores. The modified and mixed Gramian methods in Table 4.8 behave like SOBT and SOFLBT, respectively, but partially yield larger percentages of stable reduced-order models.

Table 4.8: MORscores of the (hybrid) modified and mixed second-order frequency-limited balanced truncation for the artificial fishtail example with reduced orders from 1 to 10, and the percentage of stable reduced-order models.

Method	$\mathcal{H}_\infty$	$\mathcal{H}_\infty^\Omega$	Stab. ratio
SOMFLBT(p)	0.2610	0.2770	0.9000
SOMFLBT(pm)	0.2234	0.2244	0.5000
SOMFLBT(pv)	0.2610	0.2771	0.9000
SOMFLBT(vp)	0.2175	0.2575	0.7000
SOMFLBT(vpm)	0.1240	0.1317	0.4000
SOMFLBT(v)	0.2606	0.2778	1.0000
SOMFLBT(fv)	0.2168	0.2336	1.0000
SOMFLBT(so)	0.2611	0.2770	0.9000
SOFLBTC(p)	0.2437	0.2883	0.7000
SOFLBTC(pm)	0.1783	0.2371	0.2000
SOFLBTC(pv)	0.2439	0.2888	0.8000
SOFLBTC(vp)	0.1773	0.2690	0.3000
SOFLBTC(vpm)	0.1050	0.1497	0.1000
SOFLBTC(v)	0.2192	0.2880	0.9000
SOFLBTC(fv)	0.2131	0.2356	1.0000
SOFLBTC(so)	0.2158	0.2880	0.6000
SOFLBTO(p)	0.2179	0.2888	0.6000
SOFLBTO(pm)	0.1488	0.2039	0.3000
SOFLBTO(pv)	0.2097	0.2886	0.8000
SOFLBTO(vp)	0.2467	0.2643	0.7000
SOFLBTO(vpm)	0.1374	0.1374	0.1000
SOFLBTO(v)	0.2484	0.2630	0.8000
SOFLBTO(fv)	0.1386	0.2334	1.0000
SOFLBTO(so)	0.2151	0.2860	0.6000

Table 4.9: MORscores of the (hybrid) classical and time-limited second-order balanced truncation for the artificial fishtail example with reduced orders from 1 to 10, and the percentage of stable reduced-order models.

Method	L_2	L_2^Θ	L_∞	L_∞^Θ	Stab. ratio
SOBT(p)	0.2538	0.2655	0.2571	0.2624	0.9000
SOBT(pm)	0.1467	0.1730	0.1473	0.1687	0.5000
SOBT(pv)	0.2364	0.2651	0.2391	0.2605	0.9000
SOBT(vp)	0.1750	0.1713	0.1817	0.1701	0.6000
SOBT(vpm)	0.0716	0.0946	0.0728	0.0918	0.2000
SOBT(v)	0.2538	0.2644	0.2578	0.2612	1.0000
SOBT(fv)	0.1888	0.1881	0.1934	0.1860	1.0000
SOBT(so)	0.2543	0.2678	0.2571	0.2624	1.0000
SOTLBT(p)	0.2533	0.2646	0.2568	0.2616	0.9000
SOTLBT(pm)	0.1178	0.1412	0.1180	0.1357	0.5000
SOTLBT(pv)	0.2535	0.2651	0.2564	0.2607	1.0000
SOTLBT(vp)	0.1656	0.1746	0.1679	0.1687	0.6000
SOTLBT(vpm)	0.0034	0.0064	0.0037	0.0066	0.1000
SOTLBT(v)	0.1933	0.1881	0.1955	0.1813	1.0000
SOTLBT(fv)	0.1812	0.1837	0.1841	0.1805	1.0000
SOTLBT(so)	0.2156	0.2210	0.2175	0.2166	0.9000

Time-limited methods To test the time-limited methods, the time interval of the full simulation is chosen as in Section 4.1.5.2 to be $[0, 2]$ s and the limited time interval for the reduction is set to be $[0, 0.5]$ s. For the simulations, the very same input signal as in Section 4.1.5.2 is used, namely

$$u(t) = 5000 \cdot \eta(t_j), \quad \text{for } t_j \leq t < t_{j+1},$$

with $j = 0, \ldots, 99$, equidistant time steps $t_j = j \cdot \frac{2}{99}$ and presampled Gaussian white noise $\eta(t)$. Tables 4.9 and 4.10 reveal the second-order time-limited balanced truncation methods to be at most as good as the global SOBT method in global and local approximation quality. This indicates that in fact the chosen time interval $[0, 0.5]$ s is already large enough to nearly recover the infinite Gramians. The very small MORscores in Tables 4.9 and 4.10 result from unstable time simulations occurring for some reduced-order models. Also, the time-limited methods are not always able to recover the behavior of SOBT. This can be explained by accumulation of numerical errors due to the bad conditioning of the original system combined with the pre-reduction step and the computation of the matrix exponential in the time-limited Lyapunov equations. Therefore, further investigations of

these numerical results are omitted here.

4.2.6 Conclusions

In this section, the limited balanced truncation approaches in time and frequency domains were combined with the second-order balanced truncation methods to create new structure-preserving model reduction approaches for linear second-order systems that intend to approximate the original system only in limited time and frequency ranges of interest. To provide alternative constructions of limited reduced-order models in those cases where stability could not be preserved, mixed and modified Gramian methods were considered. Solvers based on projection methods were the recommended tool to compute the solutions of the frequency- and time-limited Lyapunov equations. The strictly dissipative realization was the first-order realization of choice in computations with mechanical systems to preserve stability of projected system matrices in the matrix equation solvers. An extension of the α-shift theory from [86] was presented to accelerate numerical computations, improve conditioning of the underlying linear systems, and to handle systems with poles on the imaginary axis. The idea of two-step model reduction methods was outlined as an alternative to the use of large-scale matrix equation solvers. In two numerical examples, the different newly developed limited second-order balanced truncation methods were compared to their global counterparts. In the first example and the frequency-limited case of the second example, the new methods turned out to be very effective in local approximations. In the time-limited case of the artificial fishtail example, the time range of interest was not small enough to provide significant improvement of the local approximations. The suggested alternatives using mixed and modified Gramians have shown to be potentially more stability preserving than the fully limited methods. But in general, it was not possible to predict if a computed reduced-order model of a certain size would be stable or unstable. Also, in the comparison of the different available second-order balancing formulae from Table 3.1, no outstanding winner could be determined as for different examples also different formulae performed best.

Table 4.10: MORscores of the (hybrid) modified and mixed second-order time-limited balanced truncation for the artificial fishtail example with reduced orders from 1 to 10, and the percentage of stable reduced-order models.

Method	L_2	L_2^Θ	L_∞	L_∞^Θ	Stab. ratio
SOMTLBT(p)	0.2538	0.2655	0.2571	0.2624	1.0000
SOMTLBT(pm)	0.0988	0.1203	0.0992	0.1168	0.5000
SOMTLBT(pv)	0.2515	0.2609	0.2535	0.2557	1.0000
SOMTLBT(vp)	0.1705	0.1700	0.1766	0.1683	0.6000
SOMTLBT(vpm)	0.0477	0.0657	0.0491	0.0650	0.3000
SOMTLBT(v)	0.2537	0.2641	0.2578	0.2614	1.0000
SOMTLBT(fv)	0.1853	0.1861	0.1900	0.1847	1.0000
SOMTLBT(so)	0.2543	0.2678	0.2570	0.2625	1.0000
SOTLBTC(p)	0.2538	0.2654	0.2570	0.2624	0.9000
SOTLBTC(pm)	0.1410	0.1656	0.1428	0.1617	0.5000
SOTLBTC(pv)	0.2364	0.2650	0.2391	0.2603	0.9000
SOTLBTC(vp)	0.1850	0.1814	0.1921	0.1799	0.6000
SOTLBTC(vpm)	0.0072	0.0258	0.0082	0.0261	0.2000
SOTLBTC(v)	0.2537	0.2644	0.2578	0.2611	1.0000
SOTLBTC(fv)	0.1888	0.1881	0.1934	0.1860	1.0000
SOTLBTC(so)	0.2543	0.2679	0.2571	0.2625	1.0000
SOTLBTO(p)	0.2533	0.2646	0.2568	0.2616	0.9000
SOTLBTO(pm)	0.1269	0.1499	0.1272	0.1439	0.5000
SOTLBTO(pv)	0.2364	0.2651	0.2391	0.2605	0.9000
SOTLBTO(vp)	0.1727	0.1771	0.1747	0.1716	0.7000
SOTLBTO(vpm)	0.0387	0.0841	0.0392	0.0802	0.1000
SOTLBTO(v)	0.1935	0.1883	0.1956	0.1815	1.0000
SOTLBTO(fv)	0.1813	0.1838	0.1841	0.1805	1.0000
SOTLBTO(so)	0.2153	0.2211	0.2171	0.2166	0.9000

CHAPTER 5

STRUCTURED BILINEAR SYSTEMS

Contents

5.1 Introduction

Bilinear control systems like (2.27) are an important class of dynamical systems bridging between linear and nonlinear systems in theory and applications. They contain the multiplication of state and control variables, i.e., they are still linear in state and control separately but allow the modeling of nonlinear dynamics by the multiplication of both. Bilinear systems got a lot of attention in the last decades, as they appear naturally in the modeling of different physical phenomena, e.g., in the modeling of population, economical, thermal and mechanical dynamics [145,146], of electrical circuits [5], of plasma devices [150, 158], or of medical processes [171]. They can result from the approximation of general nonlinear systems employing the Carleman linearization process [68, 126], or appear in parameter control of PDEs [120, 124]. Recently, bilinear systems were considered as a generalizing framework in the modeling of linear stochastic [35] as well as parameter-varying systems [28, 32, 66].

Until now, bilinear systems were only considered with no further internal structure (2.27). There is a variety of model reduction methods available for the unstructured system case, for example, balanced truncation [5, 35, 116], interpolation of underlying multivariate transfer functions in the frequency domain [2, 10, 15, 65, 72, 81], Volterra series interpolation [29, 37, 85, 190] or even the construction of reduced-order bilinear systems from frequency data with the bilinear Loewner framework [12, 93]. However, in practice, as in the linear system case, also bilinear systems can inherit additional structures in the differential equations from underlying physical phenomena leading to *structured bilinear dynamical systems*. These systems come with two different concepts of structures that need to be preserved. On the one hand, there are the bilinear terms as special nonlinear structure and, on the other hand, the physically motivated internal structures of the differential equations. For example, in accordance with the main subject of this thesis, bilinear mechanical systems are given by

$$\begin{aligned} M\ddot{x}(t) + E\dot{x}(t) + Kx(t) &= \sum_{j=1}^{m} N_{\mathrm{p},j} x(t) u_j(t) + \sum_{j=1}^{m} N_{\mathrm{v},j} \dot{x}(t) u_j(t) + B_{\mathrm{u}} u(t), \\ y(t) &= C_{\mathrm{p}} x(t) + C_{\mathrm{v}} \dot{x}(t), \end{aligned} \tag{5.1}$$

with the classical second-order structure from the linear mechanical case (2.17) described by the matrices $M, E, K \in \mathbb{R}^{n_2 \times n_2}$, $B_{\mathrm{u}} \in \mathbb{R}^{n_2 \times m}$ and $C_{\mathrm{p}}, C_{\mathrm{v}} \in \mathbb{R}^{p \times n_2}$, and two types of bilinear terms with $N_{\mathrm{p},j}, N_{\mathrm{v},j} \in \mathbb{R}^{n_2 \times n_2}$, for $j = 1, \ldots, m$. While in principle one could rewrite (5.1) into a classical bilinear system (2.27) using the same idea as in the companion

form realizations of linear systems (2.18), (2.19), and (2.22), the original structure is completely lost in the model reduction process, which can lead to undesirable results in terms of accuracy, stability and physical interpretation. Moreover, other structured bilinear systems, e.g., such with internal time delays (see Section 5.2.4), cannot be represented by (2.27), which complicates the application of established model order reduction techniques. A structure-preserving reduced-order model for (5.1) that preserves the mechanical as well as the bilinear structure looks like follows:

$$\begin{aligned} \widehat{M}\ddot{\hat{x}}(t) + \widehat{E}\dot{\hat{x}}(t) + \widehat{K}\hat{x}(t) &= \sum_{j=1}^{m} \widehat{N}_{\mathrm{p},j}\hat{x}(t)u_j(t) + \sum_{j=1}^{m} \widehat{N}_{\mathrm{v},j}\dot{\hat{x}}(t)u_j(t) + \widehat{B}_{\mathrm{u}}u(t), \\ \hat{y}(t) &= \widehat{C}_{\mathrm{p}}\hat{x}(t) + \widehat{C}_{\mathrm{v}}\dot{\hat{x}}(t), \end{aligned} \tag{5.2}$$

with $\widehat{M}, \widehat{E}, \widehat{K}, \widehat{N}_{\mathrm{p},j}, \widehat{N}_{\mathrm{v},j} \in \mathbb{R}^{r_2 \times r_2}$, for $j = 1, \ldots, m$, $\widehat{B}_{\mathrm{u}} \in \mathbb{R}^{r_2 \times m}$ and $\widehat{C}_{\mathrm{p}}, \widehat{C}_{\mathrm{v}} \in \mathbb{R}^{p \times r_2}$, where $r_2 \ll n_2$.

In this chapter, a more general approach for model reduction of structured bilinear systems is established utilizing the ideas of structured transfer functions from [24] and subsystem interpolation for bilinear systems. Section 5.2 contains a generalization of the subsystem transfer functions of bilinear systems from Section 2.3.1 to the structured system case, for which in Sections 5.3 and 5.4 appropriate interpolation theory is developed. Section 5.5 contains an extension of the interpolation theory to parametric structured bilinear systems, and Section 5.6 adds the concept of tangential interpolation.

Parts of this introduction as well as Sections 5.2 to 5.4 are published in [43], and the extension to parametric systems in Section 5.5 is available in [42].

5.2 Structured bilinear systems and transfer functions

Main item for structured interpolation is the object to be interpolated, namely the multivariate transfer functions describing the dynamics of the bilinear control systems in the frequency domain. Therefore, the transfer functions (2.32) developed for the unstructured system case (2.27) will be generalized to structured bilinear systems in this section. These will be used afterwards to develop structured interpolation approaches.

5.2.1 From classical to structured bilinear systems

Inspired by different examples, the frequency domain description of linear dynamical systems got extended to the structured setting in [24]. Therein, the problem (3.17) is considered with two algebraic equations defining the system's state and output using arbitrary matrix-valued frequency-dependent functions. In case of bilinear systems, this approach can be combined with the Volterra series expansion, e.g., (2.30) and (2.31), from the unstructured case. With the two upcoming structured examples in Sections 5.2.3

and 5.2.4, it can be motivated that a suitable extension of (3.18) to the bilinear system case using regular subsystem transfer functions is given by

$$\begin{aligned}\mathcal{G}_{\mathrm{B},k}(s_1,\ldots,s_k) = \mathcal{C}(s_k)\mathcal{K}(s_k)^{-1}\left(\prod_{j=1}^{k-1}\left(I_{m^{j-1}}\otimes\mathcal{N}(s_{k-j})\right)\left(I_{m^j}\otimes\mathcal{K}(s_{k-j})^{-1}\right)\right)\\ \times\left(I_{m^{k-1}}\otimes\mathcal{B}(s_1)\right),\end{aligned} \tag{5.3}$$

for $k \geq 1$ and where $\mathcal{N}(s) = \begin{bmatrix}\mathcal{N}_1(s) & \ldots & \mathcal{N}_m(s)\end{bmatrix}$, with the matrix functions $\mathcal{C}\colon \mathbb{C} \to \mathbb{C}^{p\times n}$, $\mathcal{K}\colon \mathbb{C} \to \mathbb{C}^{n\times n}$, $\mathcal{B}\colon \mathbb{C} \to \mathbb{C}^{n\times m}$ and $\mathcal{N}_j\colon \mathbb{C} \to \mathbb{C}^{n\times n}$, for $j = 1,\ldots,m$, such that $\mathcal{G}_{\mathrm{B},k}\colon \mathbb{C}^k \to \mathbb{C}^{p\times m^k}$. The main differences to the transfer function formulation of structured linear systems (3.18) are the multivariate product structure, resulting from the subsystem idea of the Volterra series expansion, and the new matrix-valued function $\mathcal{N}(s) = \begin{bmatrix}\mathcal{N}_1(s) & \ldots & \mathcal{N}_m(s)\end{bmatrix}$ for the bilinear terms.

This general formulation includes transfer functions of classical bilinear systems (2.32) by choosing the matrix functions to be

$$\mathcal{C}(s) = \mathsf{C}, \quad \mathcal{K}(s) = s\mathsf{E} - \mathsf{A}, \quad \mathcal{N}(s) = \mathsf{N}, \quad \mathcal{B}(s) = \mathsf{B}.$$

Sections 5.2.3 and 5.2.4 will illustrate the derivation of two other structured examples, including the case of bilinear mechanical systems (5.1), which can be formulated in this general setting.

5.2.2 Structure-preserving model reduction by projection

In the linear case, projection-based model reduction methods (3.19) on the transfer function level are structure-preserving by nature; see Section 3.3.4.1. This idea can be extended to the bilinear system case. Given two basis matrices $W, V \in \mathbb{C}^{n\times r}$ of underlying projection spaces, the reduced-order bilinear system quantities are computed by

$$\widehat{\mathcal{C}}(s) = \mathcal{C}(s)V, \quad \widehat{\mathcal{K}}(s) = W^{\mathsf{H}}\mathcal{K}(s)V, \quad \widehat{\mathcal{B}}(s) = W^{\mathsf{H}}\mathcal{B}(s), \quad \widehat{\mathcal{N}}_j(s) = W^{\mathsf{H}}\mathcal{N}_j(s)V, \tag{5.4}$$

for $j = 1,\ldots,m$, and the concatenated reduced-order bilinear matrix function is

$$\widehat{\mathcal{N}}(s) = \begin{bmatrix}\widehat{\mathcal{N}}_1(s) & \ldots & \widehat{\mathcal{N}}_m(s)\end{bmatrix}.$$

The only difference to the linear setting (3.19) is the additional truncation of the bilinear terms. Utilizing the frequency-affine decomposition as in the linear case, e.g., in (3.21) and (3.22), the matrices defining time and frequency domain descriptions of the bilinear system can be extracted from (5.4). The corresponding structured regular subsystem transfer functions of the reduced-order bilinear systems are then given by

$$\begin{aligned}\widehat{\mathcal{G}}_{\mathrm{B},k}(s_1,\ldots,s_k) = \widehat{\mathcal{C}}(s_k)\widehat{\mathcal{K}}(s_k)^{-1}\left(\prod_{j=1}^{k-1}\left(I_{m^{j-1}}\otimes\widehat{\mathcal{N}}(s_{k-j})\right)\left(I_{m^j}\otimes\widehat{\mathcal{K}}(s_{k-j})^{-1}\right)\right)\\ \times\left(I_{m^{k-1}}\otimes\widehat{\mathcal{B}}(s_1)\right),\end{aligned} \tag{5.5}$$

for $k \geq 1$.

5.2.3 Bilinear second-order systems

As first example, the mechanical bilinear system (5.1) is revisited. Introducing the new state vector $\mathsf{x}(t)^\mathsf{T} = [x(t)^\mathsf{T}, \dot{x}(t)^\mathsf{T}]$, (5.1) can be rewritten in the first-order form (2.27). The resulting first-order bilinear system is then given by

$$\begin{aligned} \underbrace{\begin{bmatrix} J_{\mathrm{fc}} & 0 \\ 0 & M \end{bmatrix}}_{\mathsf{E}} \dot{\mathsf{x}}(t) &= \underbrace{\begin{bmatrix} 0 & J_{\mathrm{fc}} \\ -K & -E \end{bmatrix}}_{\mathsf{A}} \mathsf{x}(t) + \sum_{j=1}^{m} \underbrace{\begin{bmatrix} 0 & 0 \\ N_{\mathrm{p},j} & N_{\mathrm{v},j} \end{bmatrix}}_{\mathsf{N}_j} \mathsf{x}(t) u_j(t) + \underbrace{\begin{bmatrix} 0 \\ B_{\mathrm{u}} \end{bmatrix}}_{\mathsf{B}} u(t), \\ y(t) &= \underbrace{\begin{bmatrix} C_{\mathrm{p}} & C_{\mathrm{v}} \end{bmatrix}}_{\mathsf{C}} \mathsf{x}(t), \end{aligned} \tag{5.6}$$

with any invertible matrix $J_{\mathrm{fc}} \in \mathbb{R}^{n_2 \times n_2}$. For the realization (5.6), the frequency domain representation is given via the regular subsystem transfer functions (2.32). Inserting the matrices from (5.6), the occurring block structures can be used to reformulate the subsystem transfer functions in terms of the matrices defining (5.1). In general, it holds

$$\begin{aligned} (s\mathsf{E} - \mathsf{A})^{-1} &= \begin{bmatrix} sJ_{\mathrm{fc}} & -J_{\mathrm{fc}} \\ K & sM + E \end{bmatrix}^{-1} \\ &= \begin{bmatrix} \frac{1}{s} J_{\mathrm{fc}}^{-1} - \frac{1}{s}(s^2 M + sE + K)^{-1} K J_{\mathrm{fc}}^{-1} & (s^2 M + sE + K)^{-1} \\ -(s^2 M + sE + K)^{-1} K J_{\mathrm{fc}}^{-1} & s(s^2 M + sE + K)^{-1} \end{bmatrix}, \end{aligned}$$

for the frequency-dependent center terms and, by multiplication with the bilinear terms, it follows that

$$\mathsf{N}_j (s\mathsf{E} - \mathsf{A})^{-1} \mathsf{B} = \begin{bmatrix} 0 \\ (N_{\mathrm{p},j} + sN_{\mathrm{v},j})(s^2 M + sE + K)^{-1} B_{\mathrm{u}} \end{bmatrix}.$$

Consequently, the repeated multiplications of frequency-dependent terms describing the linear and bilinear dynamics in the k-th regular subsystem transfer function can be written as

$$\begin{aligned} &\left(\prod_{j=1}^{k-1} \left(I_{m^{j-1}} \otimes \mathsf{N} \right) \left(I_{m^j} \otimes (s_{k-j}\mathsf{E} - \mathsf{A})^{-1} \right) \right) (I_{m^{k-1}} \otimes \mathsf{B}) \\ &= \begin{bmatrix} 0 \\ \left(\prod_{j=1}^{k-1} \left(I_{m^{j-1}} \otimes (N_{\mathrm{p}} + s_{k-j} N_{\mathrm{v}}) \right) \left(I_{m^j} \otimes (s_{k-j}^2 M + s_{k-j} E + K)^{-1} \right) \right) (I_{m^{k-1}} \otimes B_{\mathrm{u}}) \end{bmatrix}, \end{aligned}$$

where the following concatenation of the bilinear terms from the second-order system (5.1) was used:

$$N_{\mathrm{p}} = \begin{bmatrix} N_{\mathrm{p},1} & \dots & N_{\mathrm{p},m} \end{bmatrix} \quad \text{and} \quad N_{\mathrm{v}} = \begin{bmatrix} N_{\mathrm{v},1} & \dots & N_{\mathrm{v},m} \end{bmatrix}$$

Multiplication with the one remaining frequency-dependent center term and the output matrix yields the regular transfer functions of (5.1) to be given by

$$\begin{aligned} G_{\mathrm{B},k}(s_1, \dots, s_k) = (C_{\mathrm{p}} + s_k C_{\mathrm{v}})(s_k^2 M + s_k E + K)^{-1} \\ \times \left(\prod_{j=1}^{k-1} \left(I_{m^{j-1}} \otimes (N_{\mathrm{p}} + s_{k-j} N_{\mathrm{v}}) \right) \right. \\ \left. \times \left(I_{m^j} \otimes (s_{k-j}^2 M + s_{k-j} E + K)^{-1} \right) \right) (I_{m^{k-1}} \otimes B_{\mathrm{u}}). \end{aligned} \tag{5.7}$$

In the setting of the general formulation of structured regular transfer functions (5.3), for (5.7) the matrix functions are set to be

$$\mathcal{C}(s) = C_{\mathrm{p}} + sC_{\mathrm{v}}, \quad \mathcal{K}(s) = s^2 M + sE + K, \quad \mathcal{N}(s) = N_{\mathrm{p}} + sN_{\mathrm{v}}, \quad \mathcal{B}(s) = B_{\mathrm{u}}. \tag{5.8}$$

Assume the truncation matrices W and V for projection-based model reduction (5.4) to be given. By (5.4), the reduced-order system quantities are computed via

$$\begin{aligned} \widehat{\mathcal{C}}(s) &= C_{\mathrm{p}} V + s(C_{\mathrm{v}} V), \\ \widehat{\mathcal{K}}(s) &= s^2 (W^{\mathsf{H}} M V) + s(W^{\mathsf{H}} E V) + (W^{\mathsf{H}} K V), \\ \widehat{\mathcal{N}}(s) &= (W^{\mathsf{H}} N_{\mathrm{p}} (I_m \otimes V)) + s(W^{\mathsf{H}} N_{\mathrm{v}} (I_m \otimes V)), \\ \widehat{\mathcal{B}}(s) &= W^{\mathsf{H}} B_{\mathrm{u}}. \end{aligned} \tag{5.9}$$

Since (5.9) has the same structure as the original system (5.8), the reduced-order model can be interpreted as a reduced-order second-order bilinear system of the form (5.2), where the reduced-order matrices are given in (5.9).

5.2.4 Bilinear time-delay systems

A second example for structured bilinear control systems is given by bilinear systems with an internal time delay, e.g.,

$$\begin{aligned} \mathsf{E}\dot{\mathsf{x}}(t) &= \mathsf{A}\mathsf{x}(t) + \mathsf{A}_{\mathsf{d}}\mathsf{x}(t-\tau) + \sum_{j=1}^{m} \mathsf{N}_j \mathsf{x}(t) u_j(t) + \mathsf{B}u(t), \\ y(t) &= \mathsf{C}\mathsf{x}(t), \end{aligned} \tag{5.10}$$

with $\mathsf{E}, \mathsf{A}_\mathsf{d}, \mathsf{N}_j \in \mathbb{R}^{n_1 \times n_1}$, for $j = 1, \ldots, m$, $\mathsf{B} \in \mathbb{R}^{n_1 \times m}$, $\mathsf{C} \in \mathbb{R}^{p \times n_1}$, and the delay $0 \leq \tau \in \mathbb{R}$. Systems like (5.10) were shown in [93] to have regular subsystem transfer functions of the form

$$\begin{aligned} \mathcal{G}_{\mathrm{B},k}(s_1, \ldots, s_k) = \mathsf{C}(s_k\mathsf{E} - \mathsf{A} - e^{-s_k\tau}\mathsf{A}_\mathsf{d})^{-1} \Bigg(\prod_{j=1}^{k-1} \left(I_{m^{j-1}} \otimes \mathsf{N} \right) \\ \times \left(I_{m^j} \otimes (s_{k-j}\mathsf{E} - \mathsf{A} - e^{-s_{k-j}\tau}\mathsf{A}_\mathsf{d})^{-1} \right) \Bigg) (I_{m^{k-1}} \otimes \mathsf{B}). \end{aligned} \tag{5.11}$$

As for the previous example, the regular transfer functions (5.11) of the time-delay system (5.10) can be written in the structured transfer function setting (5.3) using

$$\mathcal{C}(s) = \mathsf{C}, \quad \mathcal{K}(s) = s\mathsf{E} - \mathsf{A} - e^{-s\tau}\mathsf{A}_\mathsf{d}, \quad \mathcal{N}(s) = \mathsf{N}, \quad \text{and} \quad \mathcal{B}(s) = \mathsf{B}.$$

Once the model order reduction bases W and V are constructed, the resulting reduced-order model retains the time-delay structure of the original system as its system matrices are given by

$$\begin{aligned} \widehat{\mathcal{C}}(s) &= \mathsf{C}V, & \widehat{\mathcal{K}}(s) &= s(W^\mathsf{H}\mathsf{E}V) - (W^\mathsf{H}\mathsf{A}V) - e^{-s\tau}(W^\mathsf{H}\mathsf{A}_\mathsf{d}V), \\ \widehat{\mathcal{N}}(s) &= W^\mathsf{H}\mathsf{N}(I_m \otimes V), & \widehat{\mathcal{B}}(s) &= W^\mathsf{H}B_\mathrm{u}. \end{aligned}$$

5.3 Interpolation of single-input/single-output systems

A tremendous simplification of the structured subsystem transfer functions (5.3) appears in the SISO system case ($m = p = 1$), which will be considered in this section. Thereby, the bilinear part consists of, at most, a single term $\mathcal{N} = \mathcal{N}_1$ and the matrix functions $\mathcal{C}$ and $\mathcal{B}$ map frequency points onto either row or column vectors, respectively. In this setting, the Kronecker products in (5.3) simplify to classical matrix products and the regular subsystem transfer functions can be written as

$$\mathcal{G}_{\mathrm{B},k}(s_1, \ldots, s_k) = \mathcal{C}(s_k)\mathcal{K}(s_k)^{-1} \left(\prod_{j=1}^{k-1} \mathcal{N}(s_{k-j})\mathcal{K}(s_{k-j})^{-1} \right) \mathcal{B}(s_1), \tag{5.12}$$

for $k \geq 1$. In the remainder of this section, the theory for structure-preserving interpolation (the case of simple and higher-order (Hermite) interpolation) will be developed followed by numerical examples to illustrate the analysis.

5.3.1 Structured transfer function interpolation

The goal here is the construction of the model reduction bases W and V and, subsequently, the corresponding reduced-order structured bilinear systems via projection (5.4) such

that their leading regular subsystem transfer functions interpolate those of the original system:

$$\mathcal{G}_{\mathrm{B},k}(\sigma_1,\ldots,\sigma_k) = \widehat{\mathcal{G}}_{\mathrm{B},k}(\sigma_1,\ldots,\sigma_k),$$

for a sequence of selected interpolation points $\sigma_1,\ldots,\sigma_k \in \mathbb{C}$. The following two theorems answer the question of how the model reduction bases V and W can be constructed for the structured bilinear transfer function case similarly to the well-known results from the unstructured case, e.g., in [10]. Both theorems consider V and W independent of each other, or in other words, the interpolation conditions are satisfied only via V or W, no matter how the respective other matrix is chosen.

Theorem 5.1 (Bilinear interpolation via V):
Let $\mathcal{G}_{\mathrm{B}}$ be a bilinear SISO system, described by (5.12), and $\widehat{\mathcal{G}}_{\mathrm{B}}$ the reduced-order bilinear SISO system constructed by (5.4), with its subsystem transfer functions $\widehat{\mathcal{G}}_{\mathrm{B},k}$. Let $\sigma_1,\ldots,\sigma_k \in \mathbb{C}$ be interpolation points for which the matrix functions $\mathcal{C}$, $\mathcal{K}^{-1}$, $\mathcal{N}$, $\mathcal{B}$ and $\widehat{\mathcal{K}}^{-1}$ exist. Construct V using

$$\begin{aligned} v_1 &= \mathcal{K}(\sigma_1)^{-1}\mathcal{B}(\sigma_1), \\ v_j &= \mathcal{K}(\sigma_j)^{-1}\mathcal{N}(\sigma_{j-1})v_{j-1}, && 2 \le j \le k, \\ \operatorname{span}(V) &\supseteq \operatorname{span}\left(\begin{bmatrix} v_1 & \ldots & v_k \end{bmatrix}\right), \end{aligned}$$

and let W be an arbitrary full-rank truncation matrix of appropriate dimension. Then the subsystem transfer functions of $\widehat{\mathcal{G}}_{\mathrm{B}}$ interpolate those of $\mathcal{G}_{\mathrm{B}}$ in the following way:

$$\begin{aligned} \mathcal{G}_{\mathrm{B},1}(\sigma_1) &= \widehat{\mathcal{G}}_{\mathrm{B},1}(\sigma_1), \\ \mathcal{G}_{\mathrm{B},2}(\sigma_1,\sigma_2) &= \widehat{\mathcal{G}}_{\mathrm{B},2}(\sigma_1,\sigma_2), \\ &\vdots \\ \mathcal{G}_{\mathrm{B},k}(\sigma_1,\ldots,\sigma_k) &= \widehat{\mathcal{G}}_{\mathrm{B},k}(\sigma_1,\ldots,\sigma_k). \end{aligned}$$ ◊

Proof. As in the linear case, the main idea of this proof is the construction of appropriate projectors P_{V} of the form (3.24) onto $\operatorname{span}(V)$. Since the first subsystem transfer function corresponds to the linear case and is thereby given in Proposition 3.2, the second subsystem transfer function

$$\widehat{\mathcal{G}}_{\mathrm{B},2}(\sigma_1,\sigma_2) = \widehat{\mathcal{C}}(\sigma_2)\widehat{\mathcal{K}}(\sigma_2)^{-1}\widehat{\mathcal{N}}(\sigma_1)\widehat{\mathcal{K}}(\sigma_1)^{-1}\widehat{\mathcal{B}}(\sigma_1).$$

is considered next. With the projector (3.24), it holds

$$V\widehat{\mathcal{K}}(\sigma_1)^{-1}\widehat{\mathcal{B}}(\sigma_1) = V\widehat{\mathcal{K}}(\sigma_1)^{-1}W^{\mathsf{H}}\mathcal{B}(\sigma_1)$$

$$
\begin{aligned}
&= \underbrace{V\widehat{\mathcal{K}}(\sigma_1)^{-1}W^{\mathsf{H}}\mathcal{K}(\sigma_1)}_{=P_{\mathrm{V}}(\sigma_1)}\underbrace{\mathcal{K}(\sigma_1)^{-1}\mathcal{B}(\sigma_1)}_{=v_1} \\
&= P_{\mathrm{V}}(\sigma_1)v_1 \\
&= v_1,
\end{aligned}
$$

where the construction of V with $v_1 \in \operatorname{span}(V)$ and the resulting identity (3.26) by multiplying elements of $\operatorname{span}(V)$ with the projector P_{V} are used. Therefore, it holds

$$
\begin{aligned}
\widehat{\mathcal{G}}_{\mathrm{B},2}(\sigma_1,\sigma_2) &= \widehat{\mathcal{C}}(\sigma_2)\widehat{\mathcal{K}}(\sigma_2)^{-1}W^{\mathsf{H}}\mathcal{N}(\sigma_1)\mathcal{K}(\sigma_1)^{-1}\mathcal{B}(\sigma_1) \\
&= \mathcal{C}(\sigma_2)V\widehat{\mathcal{K}}(\sigma_2)^{-1}W^{\mathsf{H}}\mathcal{N}(\sigma_1)v_1.
\end{aligned}
$$

Analogously, for the remaining reduced-order terms, a second projector is constructed such that

$$
V\widehat{\mathcal{K}}(\sigma_2)^{-1}W^{\mathsf{H}}\mathcal{N}(\sigma_1)v_1 = \underbrace{V\widehat{\mathcal{K}}(\sigma_2)^{-1}W^{\mathsf{H}}\mathcal{K}(\sigma_2)}_{=P_{\mathrm{V}}(\sigma_2)}\underbrace{\mathcal{K}(\sigma_2)^{-1}\mathcal{N}(\sigma_1)v_1}_{=v_2} = P_{\mathrm{V}}(\sigma_2)v_2 = v_2
$$

holds, since by construction $v_2 \in \operatorname{span}(V)$. Expanding v_2 into the matrix functions yields the interpolation of the second subsystem transfer function

$$
\widehat{\mathcal{G}}_{\mathrm{B},2}(\sigma_1,\sigma_2) = \mathcal{C}(\sigma_2)\mathcal{K}(\sigma_2)^{-1}\mathcal{N}(\sigma_1)\mathcal{K}(\sigma_1)^{-1}\mathcal{B}(\sigma_1) = \mathcal{G}_{\mathrm{B},2}(\sigma_1,\sigma_2).
$$

Via induction over the transfer function index k and with the same construction arguments of the projectors (3.24) onto $\operatorname{span}(V)$ the theorem holds. □

The proof of Theorem 5.1 shows the recursive construction of the projection space to be necessary for the interpolation of higher-level regular transfer functions via projection. For example, putting v_1 into the projection space allows the interpolation of $\mathcal{G}_{\mathrm{B},1}(\sigma_1)$, but for the interpolation of $\mathcal{G}_{\mathrm{B},2}(\sigma_1,\sigma_2)$ having only $v_2 \in \operatorname{span}(V)$ is not enough. Both vectors are necessary to lie in the projection space, i.e., only from $v_1, v_2 \in \operatorname{span}(V)$ follows the interpolation of $\mathcal{G}_{\mathrm{B},2}(\sigma_1,\sigma_2)$. Consequently, aiming for the interpolation of the k-th subsystem transfer function directly yields the interpolation of all preceding transfer function levels.

Also, it should be noted that W was an arbitrary full-rank truncation matrix of suitable dimensions but with no additional constraints for the interpolation of (5.12). Theorem 5.2 will be the counterpart to Theorem 5.1 by only giving constraints for the left model reduction basis W, while V is now allowed to be arbitrary.

Theorem 5.2 (Bilinear interpolation via W):
Let $\mathcal{G}_{\mathrm{B}}$, $\widehat{\mathcal{G}}_{\mathrm{B}}$, and the interpolation points $\sigma_1,\ldots,\sigma_k \in \mathbb{C}$ be as in Theorem 5.1. Construct W using

$$
\begin{aligned}
w_1 &= \mathcal{K}(\sigma_k)^{-\mathsf{H}}\mathcal{C}(\sigma_k)^{\mathsf{H}}, \\
w_j &= \mathcal{K}(\sigma_{k-j+1})^{-\mathsf{H}}\mathcal{N}(\sigma_{k-j+1})^{\mathsf{H}}w_{j-1}, \qquad 2 \le j \le k, \\
\operatorname{span}(W) &\supseteq \operatorname{span}\left(\begin{bmatrix} w_1 & \ldots & w_k \end{bmatrix}\right),
\end{aligned}
$$

and let V be an arbitrary full-rank truncation matrix of appropriate dimension. Then the subsystem transfer functions of $\widehat{\mathcal{G}}_{\mathrm{B}}$ interpolate those of $\mathcal{G}_{\mathrm{B}}$ in the following way:

$$\begin{aligned}\mathcal{G}_{\mathrm{B},1}(\sigma_k) &= \widehat{\mathcal{G}}_{\mathrm{B},1}(\sigma_k),\\ \mathcal{G}_{\mathrm{B},2}(\sigma_{k-1},\sigma_k) &= \widehat{\mathcal{G}}_{\mathrm{B},2}(\sigma_{k-1},\sigma_k),\\ &\vdots\\ \mathcal{G}_{\mathrm{B},k}(\sigma_1,\ldots,\sigma_k) &= \widehat{\mathcal{G}}_{\mathrm{B},k}(\sigma_1,\ldots,\sigma_k).\end{aligned} \qquad \Diamond$$

Proof. The proof of this theorem follows analogously to the proof of Theorem 5.1 but now with the construction of the projector P_{W} from (3.25) onto $\operatorname{span}(W)$. For illustration and later reference, the proof is sketched nevertheless. As in the proof of Theorem 5.1, the reduced-order second subsystem transfer function is considered in the proposed interpolation points, i.e.,

$$\widehat{\mathcal{G}}_{\mathrm{B},2}(\sigma_{k-1},\sigma_k) = \widehat{\mathcal{C}}(\sigma_k)\widehat{\mathcal{K}}(\sigma_k)^{-1}\widehat{\mathcal{N}}(\sigma_{k-1})\widehat{\mathcal{K}}(\sigma_{k-1})^{-1}\widehat{\mathcal{B}}(\sigma_{k-1}).$$

In contrast to Theorem 5.1, the projector (3.25) is now used such that

$$\begin{aligned}W\widehat{\mathcal{K}}(\sigma_k)^{-\mathsf{H}}\widehat{\mathcal{C}}(\sigma_k)^{\mathsf{H}} &= W\widehat{\mathcal{K}}(\sigma_k)^{-\mathsf{H}}V^{\mathsf{H}}\mathcal{C}(\sigma_k)^{\mathsf{H}}\\ &= \underbrace{W\widehat{\mathcal{K}}(\sigma_k)^{-\mathsf{H}}V^{\mathsf{H}}\mathcal{K}(\sigma_k)^{\mathsf{H}}}_{=P_{\mathrm{W}}(\sigma_k)}\underbrace{\mathcal{K}(\sigma_k)^{-\mathsf{H}}\mathcal{C}(\sigma_k)^{\mathsf{H}}}_{=w_1}\\ &= P_{\mathrm{W}}(\sigma_k)w_1\\ &= w_1\end{aligned}$$

holds, since by construction $w_1 \in \operatorname{span}(W)$ and (3.26). This yields the reduced-order transfer function to satisfy

$$\widehat{\mathcal{G}}_{\mathrm{B},2}(\sigma_{k-1},\sigma_k) = w_1^{\mathsf{H}}\mathcal{N}(\sigma_{k-1})V\widehat{\mathcal{K}}(\sigma_{k-1})^{-1}W^{\mathsf{H}}\mathcal{B}(\sigma_{k-1}).$$

For the rest, again a projector like (3.25) is constructed as follows:

$$\begin{aligned}W\widehat{\mathcal{K}}(\sigma_{k-1})^{-\mathsf{H}}V^{\mathsf{H}}\mathcal{N}(\sigma_{k-1})^{\mathsf{H}}w_1 &= \underbrace{W\widehat{\mathcal{K}}(\sigma_{k-1})^{-\mathsf{H}}V^{\mathsf{H}}\mathcal{K}(\sigma_{k-1})^{\mathsf{H}}}_{=P_{\mathrm{W}}(\sigma_{k-1})}\underbrace{\mathcal{K}(\sigma_{k-1})^{-\mathsf{H}}\mathcal{N}(\sigma_{k-1})^{\mathsf{H}}w_1}_{=w_2}\\ &= P_{\mathrm{W}}(\sigma_{k-1})w_2\\ &= w_2,\end{aligned}$$

which results in the interpolation of the second subsystem transfer function

$$\widehat{\mathcal{G}}_{\mathrm{B},2}(\sigma_{k-1},\sigma_k) = w_2^{\mathsf{H}}\mathcal{B}(\sigma_{k-1}) = \mathcal{G}_{\mathrm{B},2}(\sigma_{k-1},\sigma_k).$$

The rest of the theorem follows via induction over the transfer function index k. □

The main difference between Theorems 5.1 and 5.2 is the order in which the interpolation points have to be used to end up in the same sequence for the k-th subsystem transfer function. Switching between the two interpolation schemes leads to a reverse ordering of the interpolation points for the intermediate transfer functions, which can easily be used to increase the number of matched interpolation conditions. The last theorem of this section states now the combination of Theorems 5.1 and 5.2 in the two-sided projection approach.

Theorem 5.3 (Bilinear interpolation by two-sided projection):
Let $\mathcal{G}_{\mathrm{B}}$ and $\widehat{\mathcal{G}}_{\mathrm{B}}$ be as in Theorem 5.1, let V be constructed as in Theorem 5.1 for a given sequence of interpolation points $\sigma_1, \ldots, \sigma_k \in \mathbb{C}$, and let W be constructed as in Theorem 5.2 for another sequence of interpolation points $\varsigma_1, \ldots, \varsigma_\theta \in \mathbb{C}$, for which the matrix functions $\mathcal{C}$, $\mathcal{K}^{-1}$, $\mathcal{N}$, $\mathcal{B}$ and $\widehat{\mathcal{K}}^{-1}$ exist. Then the regular subsystem transfer functions of $\widehat{\mathcal{G}}_{\mathrm{B}}$ interpolate those of $\mathcal{G}_{\mathrm{B}}$ in the following way:

$$\begin{aligned} \mathcal{G}_{\mathrm{B},1}(\sigma_1) &= \widehat{\mathcal{G}}_{\mathrm{B},1}(\sigma_1), \quad \ldots, \quad \mathcal{G}_{\mathrm{B},k}(\sigma_1, \ldots, \sigma_k) = \widehat{\mathcal{G}}_{\mathrm{B},k}(\sigma_1, \ldots, \sigma_k), \quad \text{and} \\ \mathcal{G}_{\mathrm{B},1}(\varsigma_\theta) &= \widehat{\mathcal{G}}_{\mathrm{B},1}(\varsigma_\theta), \quad \ldots, \quad \mathcal{G}_{\mathrm{B},\theta}(\varsigma_1, \ldots, \varsigma_\theta) = \widehat{\mathcal{G}}_{\mathrm{B},\theta}(\varsigma_1, \ldots, \varsigma_\theta), \end{aligned} \tag{5.13}$$

and, additionally,

$$\mathcal{G}_{\mathrm{B},q+\eta}(\sigma_1, \ldots, \sigma_q, \varsigma_{\theta-\eta+1}, \ldots, \varsigma_\theta) = \widehat{\mathcal{G}}_{\mathrm{B},q+\eta}(\sigma_1, \ldots, \sigma_q, \varsigma_{\theta-\eta+1}, \ldots, \varsigma_\theta), \tag{5.14}$$

for $1 \leq q \leq k$ and $1 \leq \eta \leq \theta$. ◊

Proof. The interpolation conditions in (5.13) are a reminder of the results in Theorems 5.1 and 5.2. Only the mixed interpolation conditions (5.14) involving both sequences of interpolation points are left to be proven. Therefore, a combination of the projectors P_{V} and P_{W} corresponding to the two truncation matrices V and W and their underlying projection spaces is needed. Let q and η be as in the theorem, the reduced-order $(q+\eta)$-th subsystem transfer function can be written as

$$\begin{aligned} &\widehat{\mathcal{G}}_{\mathrm{B},q+\eta}(\sigma_1, \ldots, \sigma_q, \varsigma_{\theta-\eta+1}, \ldots, \varsigma_\theta) \\ &= \widehat{\mathcal{C}}(\varsigma_\theta)\widehat{\mathcal{K}}(\varsigma_\theta)^{-1} \left(\prod_{j=1}^{\eta-1} \widehat{\mathcal{N}}(\varsigma_{\theta-j})\widehat{\mathcal{K}}(\varsigma_{\theta-j})^{-1} \right) \widehat{\mathcal{N}}(\sigma_q) \\ &\quad \times \left(\prod_{i=0}^{q-2} \widehat{\mathcal{K}}(\sigma_{q-i})^{-1}\widehat{\mathcal{N}}(\sigma_{q-i-1}) \right) \widehat{\mathcal{K}}(\sigma_1)^{-1}\widehat{\mathcal{B}}(\sigma_1) \\ &=: \hat{w}_\eta^{\mathsf{H}} \widehat{\mathcal{N}}(\sigma_q)\hat{v}_q \\ &= \hat{w}_\eta^{\mathsf{H}} W^{\mathsf{H}} \mathcal{N}(\sigma_q) V \hat{v}_q, \end{aligned}$$

where the vectors $\hat{w}_\eta^{\mathsf{H}}$ and $\hat{v}_q$ resemble the vectors from construction of the projection spaces $\operatorname{span}(W)$ and $\operatorname{span}(V)$ with the same subscripts but using the reduced-order

matrix functions. Following the proof of Theorem 5.1, it can be shown via induction that the identity

$$\begin{aligned} V\hat{v}_q &= V\left(\prod_{i=0}^{q-3}\widehat{\mathcal{K}}(\sigma_{q-i})^{-1}\widehat{\mathcal{N}}(\sigma_{q-i-1})\right)\widehat{\mathcal{K}}(\sigma_2)^{-1}\widehat{\mathcal{N}}(\sigma_1)\widehat{\mathcal{K}}(\sigma_1)^{-1}\widehat{\mathcal{B}}(\sigma_1) \\ &= V\left(\prod_{i=0}^{q-3}\widehat{\mathcal{K}}(\sigma_{q-i})^{-1}\widehat{\mathcal{N}}(\sigma_{q-i-1})\right)\widehat{\mathcal{K}}(\sigma_2)^{-1}W^{\mathsf{H}}\mathcal{N}(\sigma_1)P_{\mathrm{V}}(\sigma_1)v_1 \\ &\vdots \\ &= V\widehat{\mathcal{K}}(\sigma_q)^{-1}W^{\mathsf{H}}\mathcal{N}(\sigma_{q-1})v_{q-1} \\ &= P_{\mathrm{V}}(\sigma_q)v_q \\ &= v_q \end{aligned}$$

holds by construction of $\operatorname{span}(V)$, with v_q as the q-th constructed vector in Theorem 5.1. Analogously with the proof of Theorem 5.2, one can show that

$$W\hat{w}_\eta = w_\eta$$

has to hold by construction of $\operatorname{span}(W)$, with w_η as the η-th constructed vector in Theorem 5.2. With these two identities, the interpolation conditions follow

$$\begin{aligned} &\widehat{\mathcal{G}}_{\mathrm{B},q+\eta}(\sigma_1,\ldots,\sigma_q,\varsigma_{\theta-\eta+1},\ldots,\varsigma_\theta) \\ &= \hat{w}_\eta^{\mathsf{H}}W^{\mathsf{H}}\mathcal{N}(\sigma_q)V\hat{v}_q \\ &= w_\eta^{\mathsf{H}}\mathcal{N}(\sigma_q)v_q \\ &= \mathcal{C}(\varsigma_\theta)\mathcal{K}(\varsigma_\theta)^{-1}\left(\prod_{j=1}^{\eta-1}\mathcal{N}(\varsigma_{\theta-j})\mathcal{K}(\varsigma_{\theta-j})^{-1}\right)\mathcal{N}(\sigma_q) \\ &\quad\times\left(\prod_{i=0}^{q-2}\mathcal{K}(\sigma_{q-i})^{-1}\mathcal{N}(\sigma_{q-i-1})\right)\mathcal{K}(\sigma_1)^{-1}\mathcal{B}(\sigma_1) \\ &= \mathcal{G}_{\mathrm{B},q+\eta}(\sigma_1,\ldots,\sigma_q,\varsigma_{\theta-\eta+1},\ldots,\varsigma_\theta). \end{aligned}$$

□

With Theorem 5.3 it is now proven that higher-level transfer functions can be interpolated in an implicit way evaluating only parts of lower-level transfer functions and combining the resulting subspaces in a two-sided projection approach. In fact, by using Theorem 5.3, it is possible to interpolate transfer functions up to level $k+\theta$, while restricting the evaluation to only the k-th level for the right projection space and the η-th level for the left one. In the same setting, it is possible to match up to $k+\theta+k\cdot\theta$ interpolation conditions. These results are similar to the unstructured system case [10]. The special case of identical sequences of interpolation points will result in the interpolation of partial derivatives with respect to the function's frequency arguments. This will be discussed in the upcoming section regarding Hermite interpolation.

5.3.2 Matching Hermite interpolation conditions

In the linear case (Proposition 3.2), it is possible to interpolate higher-order derivatives of the transfer function in various ways. Similar results can be obtained for the multivariate transfer functions of bilinear systems considering partial derivatives with respect to the different frequency arguments. The following theorem states a Hermite interpolation extension of Theorem 5.1 via V only.

Theorem 5.4 (Bilinear Hermite interpolation via V):
Let $\mathcal{G}_{\mathrm{B}}$ be a bilinear SISO system, described by (5.12), and $\widehat{\mathcal{G}}_{\mathrm{B}}$ the reduced-order bilinear SISO system constructed by (5.4), with its subsystem transfer functions $\widehat{\mathcal{G}}_{\mathrm{B},k}$. Let $\sigma_1, \ldots, \sigma_k \in \mathbb{C}$ be interpolation points for which the matrix functions $\mathcal{C}$, $\mathcal{K}^{-1}$, $\mathcal{N}$, $\mathcal{B}$ and $\widehat{\mathcal{K}}^{-1}$ are complex differentiable, and $\ell_1, \ldots, \ell_k \in \mathbb{N}_0$ orders of partial derivatives to be matched in the k-th subsystem transfer function. Construct V using

$$
\begin{aligned}
v_{1,j_1} &= \partial_{s^{j_1}}(\mathcal{K}^{-1}\mathcal{B})(\sigma_1), && j_1 = 0, \ldots, \ell_1,\\
v_{2,j_2} &= \partial_{s^{j_2}}\mathcal{K}^{-1}(\sigma_2)\partial_{s^{\ell_1}}(\mathcal{N}\mathcal{K}^{-1}\mathcal{B})(\sigma_1), && j_2 = 0, \ldots, \ell_2,\\
&\vdots\\
v_{k,j_k} &= \partial_{s^{j_k}}\mathcal{K}^{-1}(\sigma_k)\left(\prod_{j=1}^{k-2}\partial_{s^{\ell_{k-j}}}(\mathcal{N}\mathcal{K}^{-1})(\sigma_{k-j})\right)\\
&\quad\times \partial_{s^{\ell_1}}(\mathcal{N}\mathcal{K}^{-1}\mathcal{B})(\sigma_1), && j_k = 0, \ldots, \ell_k,\\
\operatorname{span}(V) &\supseteq \operatorname{span}\left(\begin{bmatrix} v_{1,0} & \ldots & v_{k,\ell_k}\end{bmatrix}\right),
\end{aligned}
$$

and let W be an arbitrary full-rank truncation matrix of appropriate dimension. Then the subsystem transfer functions of $\widehat{\mathcal{G}}_{\mathrm{B}}$ interpolate those of $\mathcal{G}_{\mathrm{B}}$ in the following way:

$$
\begin{aligned}
\partial_{s_1^{j_1}}\mathcal{G}_{\mathrm{B},1}(\sigma_1) &= \partial_{s_1^{j_1}}\widehat{\mathcal{G}}_{\mathrm{B},1}(\sigma_1), && j_1 = 0, \ldots, \ell_1,\\
\partial_{s_1^{\ell_1}s_2^{j_2}}\mathcal{G}_{\mathrm{B},2}(\sigma_1,\sigma_2) &= \partial_{s_1^{\ell_1}s_2^{j_2}}\widehat{\mathcal{G}}_{\mathrm{B},2}(\sigma_1,\sigma_2), && j_2 = 0, \ldots, \ell_2,\\
&\vdots\\
\partial_{s_1^{\ell_1}\cdots s_{k-1}^{\ell_{k-1}}s_k^{j_k}}\mathcal{G}_{\mathrm{B},k}(\sigma_1,\ldots,\sigma_k) &= \partial_{s_1^{\ell_1}\cdots s_{k-1}^{\ell_{k-1}}s_k^{j_k}}\widehat{\mathcal{G}}_{\mathrm{B},k}(\sigma_1,\ldots,\sigma_k), && j_k = 0, \ldots, \ell_k. \qquad \lozenge
\end{aligned}
$$

Proof. As in case of classical interpolation (Theorem 5.1), the first subsystem transfer function corresponds to the linear case and, thereby, the interpolation results are available in Proposition 3.2. Also, the case of all derivative orders to be zero, $\ell_1 = \ldots = \ell_k = 0$ resembles Theorem 5.1. The first interpolation condition to be considered next is given for $k = 2$ and $j_2 = 0$. Using the product rule, the partial derivative with respect to the

first frequency argument only concerns the rightmost product of the bilinear, linear and input terms, which can in general be written as

$$\partial_{s^{\ell_1}}(\mathcal{N}\mathcal{K}^{-1}\mathcal{B})(\sigma_1) = \sum_{i=0}^{\ell_1} c_i \partial_{s^i}\mathcal{N}(\sigma_1)\partial_{s^{\ell_1-i}}(\mathcal{K}^{-1}\mathcal{B})(\sigma_1),$$

for some appropriate constants $c_i \in \mathbb{C}$, $i = 0, \ldots, \ell_1$. The reduced-order transfer function is then given by

$$\begin{aligned}
\partial_{s_1^{\ell_1}}\widehat{\mathcal{G}}_{\mathrm{B},2}(\sigma_1,\sigma_2) &= \widehat{\mathcal{C}}(\sigma_2)\widehat{\mathcal{K}}(\sigma_2)^{-1}\partial_{s^{\ell_1}}(\widehat{\mathcal{N}}\widehat{\mathcal{K}}^{-1}\widehat{\mathcal{B}})(\sigma_1) \\
&= \widehat{\mathcal{C}}(\sigma_2)\widehat{\mathcal{K}}(\sigma_2)^{-1}\left(\sum_{i=0}^{\ell_1} c_i \partial_{s^i}\widehat{\mathcal{N}}(\sigma_1)\partial_{s^{\ell_1-i}}(\widehat{\mathcal{K}}^{-1}\widehat{\mathcal{B}})(\sigma_1)\right) \\
&=: \widehat{\mathcal{C}}(\sigma_2)\widehat{\mathcal{K}}(\sigma_2)^{-1}W^{\mathsf{H}}\left(\sum_{i=0}^{\ell_1} c_i \partial_{s^i}\mathcal{N}(\sigma_1)V\hat{v}_{1,\ell_1-i}\right).
\end{aligned}$$

As in the proofs of the previous interpolation theorems, by construction of $\operatorname{span}(V)$, the identity

$$V\hat{v}_{1,\ell_1-i} = v_{1,\ell_1-i}$$

holds for all $0 \leq i \leq \ell_1$. This allows to further rewrite the reduced-order transfer function such that the interpolation condition holds:

$$\begin{aligned}
\partial_{s_1^{\ell_1}}\widehat{\mathcal{G}}_{\mathrm{B},2}(\sigma_1,\sigma_2) &= \widehat{\mathcal{C}}(\sigma_2)\widehat{\mathcal{K}}(\sigma_2)^{-1}W^{\mathsf{H}}\left(\sum_{i=0}^{\ell_1} c_i \partial_{s^i}\mathcal{N}(\sigma_1)v_{1,\ell_1-i}\right) \\
&= \widehat{\mathcal{C}}(\sigma_2)\widehat{\mathcal{K}}(\sigma_2)^{-1}W^{\mathsf{H}}\partial_{s_1^{\ell_1}}(\mathcal{N}\mathcal{K}^{-1}\mathcal{B})(\sigma_1) \\
&= \mathcal{C}(\sigma_2)P_{\mathrm{V}}(\sigma_2)v_{2,0} \\
&= \mathcal{C}(\sigma_2)v_{2,0} \\
&= \partial_{s_1^{\ell_1}}\mathcal{G}_{\mathrm{B},2}(\sigma_1,\sigma_2),
\end{aligned}$$

where $P_{\mathrm{V}}(\sigma_2)$ is the projector from (3.24) onto $\operatorname{span}(V)$. Using the same arguments, the rest of the theorem follows via induction over the partial derivative orders $j_2, \ldots, j_k$ and the transfer function index k. □

While for previous interpolation results, the subspaces were constructed in a recursive way by using previously computed evaluations in the next step, this is not (easily) possible in Theorem 5.4 due to $\mathcal{N}$ depending on the frequency argument of the terms right of it. Note that, in this sense, the construction in Theorem 5.4 becomes a recursive formula again in case of $\mathcal{N}$ being constant. Additionally, one can observe that for the interpolation

of the ℓ-th partial derivative, $\ell = \ell_1 + \ldots + \ell_k$, of the k-th subsystem transfer function $\mathcal{G}_{\mathrm{B},k}$ in the interpolation points $\sigma_1, \ldots, \sigma_k$, the maximal dimension of the projection space $\operatorname{span}(V)$ is given by $\ell + k$ if all constructed vectors are linear independent.

As before, it is possible to formulate the counterpart to Theorem 5.4 using the output term of the transfer function for the construction of W instead of V. In addition to reversing the order of interpolation points, the order of the partial derivatives needs to be reversed as well for Hermite interpolation.

Theorem 5.5 (Bilinear Hermite interpolation via W):
Let $\mathcal{G}_{\mathrm{B}}$, $\widehat{\mathcal{G}}_{\mathrm{B}}$, the interpolation points $\sigma_1, \ldots, \sigma_k \in \mathbb{C}$ and the orders of partial derivatives $\ell_1, \ldots, \ell_k \in \mathbb{N}_0$ be as in Theorem 5.4. Construct W using

$$\begin{aligned}
w_{1,j_k} &= \partial_{s^{j_k}}(\mathcal{K}^{-\mathsf{H}}\mathcal{C}^{\mathsf{H}})(\sigma_k), && j_k = 0, \ldots, \ell_k,\\
w_{2,j_{k-1}} &= \partial_{s^{j_{k-1}}}(\mathcal{K}^{-\mathsf{H}}\mathcal{N}^{\mathsf{H}})(\sigma_{k-1})w_{1,\ell_k}, && j_{k-1} = 0, \ldots, \ell_{k-1},\\
&\vdots\\
w_{k,j_1} &= \partial_{s^{j_1}}(\mathcal{K}^{-\mathsf{H}}\mathcal{N}^{\mathsf{H}})(\sigma_1)w_{k-1,\ell_2}, && j_1 = 0, \ldots, \ell_1,\\
\operatorname{span}(W) &\supseteq \operatorname{span}\left(\begin{bmatrix} w_{1,0} & \ldots & w_{k,\ell_k}\end{bmatrix}\right),
\end{aligned}$$

and let V be an arbitrary full-rank truncation matrix of appropriate dimension. Then the subsystem transfer functions of $\widehat{\mathcal{G}}_{\mathrm{B}}$ interpolate those of $\mathcal{G}_{\mathrm{B}}$ in the following way:

$$\begin{aligned}
\partial_{s_1^{j_k}}\mathcal{G}_{\mathrm{B},1}(\sigma_k) &= \partial_{s_1^{j_k}}\widehat{\mathcal{G}}_{\mathrm{B},1}(\sigma_k), && j_k = 0, \ldots, \ell_k,\\
\partial_{s_1^{j_{k-1}}s_2^{\ell_k}}\widehat{\mathcal{G}}_{\mathrm{B},2}(\sigma_{k-1}, \sigma_k) &= \partial_{s_1^{j_{k-1}}s_2^{\ell_k}}\widehat{\mathcal{G}}_{\mathrm{B},2}(\sigma_{k-1}, \sigma_k), && j_{k-1} = 0, \ldots, \ell_{k-1},\\
&\vdots\\
\partial_{s_1^{j_1}s_2^{\ell_2}\ldots s_k^{\ell_k}}\mathcal{G}_{\mathrm{B},k}(\sigma_1, \ldots, \sigma_k) &= \partial_{s_1^{j_1}s_2^{\ell_2}\ldots s_k^{\ell_k}}\widehat{\mathcal{G}}_{\mathrm{B},k}(\sigma_1, \ldots, \sigma_k), && j_1 = 0, \ldots, \ell_1. \qquad \lozenge
\end{aligned}$$

Proof. The proof follows directly using the projection arguments from the proofs of Theorems 5.2 and 5.4 with the construction of P_{W} from (3.25). □

It can be noted that Theorem 5.5, in contrast to Theorem 5.4, resembles the recursive structure from the classical interpolation of the subsystem transfer functions in Theorem 5.2. This results from the frequency dependency of the bilinear terms on the frequency argument from the right side but not from the left.

An interesting approach in the structured linear case is the implicit matching of Hermite interpolation conditions; see Proposition 3.2 Part (c). It is possible to avoid the evaluation of higher-order derivatives of the transfer function by using the two-sided projection approach in the same interpolation points for both projection spaces. Next, this idea is extended to the structured bilinear system case. The following result is a special case of Theorem 5.3 by using identical sets of interpolation points for the construction of V and W.

Theorem 5.6 (Implicit bilinear Hermite interpolation):
Let $\mathcal{G}_\mathrm{B}$ be a bilinear SISO system, described by (5.12), and $\widehat{\mathcal{G}}_\mathrm{B}$ the reduced-order bilinear SISO system constructed by (5.4), with its subsystem transfer functions $\widehat{\mathcal{G}}_{\mathrm{B},k}$. Let V and W be constructed as in Theorems 5.1 and 5.2, respectively, for the same sequence of interpolation points $\sigma_1, \ldots, \sigma_k \in \mathbb{C}$, for which the matrix functions $\mathcal{C}$, $\mathcal{K}^{-1}$, $\mathcal{N}$, $\mathcal{B}$ and $\widehat{\mathcal{K}}^{-1}$ are complex differentiable. Then the regular subsystem transfer functions of $\widehat{\mathcal{G}}_\mathrm{B}$ interpolate those of $\mathcal{G}_\mathrm{B}$ in the following way:

$$\begin{aligned}
\mathcal{G}_{\mathrm{B},1}(\sigma_1) &= \widehat{\mathcal{G}}_{\mathrm{B},1}(\sigma_1), \quad \ldots, \quad \mathcal{G}_{\mathrm{B},k-1}(\sigma_1, \ldots, \sigma_{k-1}) = \widehat{\mathcal{G}}_{\mathrm{B},k-1}(\sigma_1, \ldots, \sigma_{k-1}), \\
\mathcal{G}_{\mathrm{B},1}(\sigma_k) &= \widehat{\mathcal{G}}_{\mathrm{B},1}(\sigma_k), \quad \ldots, \quad \mathcal{G}_{\mathrm{B},k-1}(\sigma_2, \ldots, \sigma_k) = \widehat{\mathcal{G}}_{\mathrm{B},k-1}(\sigma_2, \ldots, \sigma_k),
\end{aligned}$$

and, additionally,

$$\begin{aligned}
\mathcal{G}_{\mathrm{B},k}(\sigma_1, \ldots, \sigma_k) &= \widehat{\mathcal{G}}_{\mathrm{B},k}(\sigma_1, \ldots, \sigma_k), \\
\nabla\mathcal{G}_{\mathrm{B},k}(\sigma_1, \ldots, \sigma_k) &= \nabla\widehat{\mathcal{G}}_{\mathrm{B},k}(\sigma_1, \ldots, \sigma_k), \\
\mathcal{G}_{\mathrm{B},q+\eta}(\sigma_1, \ldots, \sigma_q, \sigma_{k-\eta+1}, \ldots, \sigma_k) &= \widehat{\mathcal{G}}_{\mathrm{B},q+\eta}(\sigma_1, \ldots, \sigma_q, \sigma_{k-\eta+1}, \ldots, \sigma_k)
\end{aligned}$$

hold, for $1 \leq q, \eta \leq k$. ◊

Proof. The simple interpolation of the subsystem transfer functions without partial derivatives directly follows from Theorem 5.3 by using identical sets of interpolation points for V and W. The interpolation of the complete Jacobian $\nabla\mathcal{G}_{\mathrm{B},k}$ is left to be proven. As the case $k = 1$ is covered by Proposition 3.2, assume $k > 1$. Since the single entries of the Jacobi matrix are partial derivatives of the transfer function with respect to a single frequency argument each, this can be combined with the special structure of the subsystem transfer functions $\mathcal{G}_{\mathrm{B},k}$, where each matrix-valued function only depends on a single frequency argument. With this observation, three general cases of the differentiation of the matrix-valued functions can occur depending on the chosen differentiation variable:

$$\begin{aligned}
s_1: &\quad \partial_s(\mathcal{N}\mathcal{K}^{-1}\mathcal{B}) = (\partial_s\mathcal{N})\,\mathcal{K}^{-1}\mathcal{B} + \mathcal{N}\left(\partial_s(\mathcal{K}^{-1}\mathcal{B})\right), \\
s_j: &\quad \partial_s(\mathcal{N}\mathcal{K}^{-1}) = (\partial_s\mathcal{N})\,\mathcal{K}^{-1} + \mathcal{N}\left(\partial_s\mathcal{K}^{-1}\right), \qquad \text{for } 1 < j < k, \\
s_k: &\quad \partial_s(\mathcal{C}\mathcal{K}^{-1}) = (\partial_s\mathcal{C})\,\mathcal{K}^{-1} + \mathcal{C}\left(\partial_s\mathcal{K}^{-1}\right).
\end{aligned}$$

The resulting three cases of partial derivatives work analogously to each other, therefore, it is enough to proof one of them, which will be here the first entry of the Jacobian, i.e., $\partial_{s_1}\mathcal{G}_{\mathrm{B},k}$. The partial derivative of the terms of interest is extended further into

$$\partial_s(\mathcal{N}\mathcal{K}^{-1}\mathcal{B}) = (\partial_s\mathcal{N})\,\mathcal{K}^{-1}\mathcal{B} - \mathcal{N}\mathcal{K}^{-1}\,(\partial_s\mathcal{K})\,\mathcal{K}^{-1}\mathcal{B} + \mathcal{N}\mathcal{K}^{-1}\,(\partial_s\mathcal{B}),$$

which allows to write the complete partial derivative of the reduced-order transfer function to be

$$\begin{aligned}
&\partial_{s_1}\widehat{\mathcal{G}}_{\mathrm{B},k}(\sigma_1,\ldots,\sigma_k)\\
&= \widehat{\mathcal{C}}(\sigma_k)\widehat{\mathcal{K}}(\sigma_k)^{-1}\left(\prod_{j=1}^{k-2}\widehat{\mathcal{N}}(\sigma_{k-j})\widehat{\mathcal{K}}(\sigma_{k-j})^{-1}\right)\partial_s(\widehat{\mathcal{N}}\widehat{\mathcal{K}}^{-1}\widehat{\mathcal{B}})(\sigma_1)\\
&= \widehat{\mathcal{C}}(\sigma_k)\widehat{\mathcal{K}}(\sigma_k)^{-1}\left(\prod_{j=1}^{k-2}\widehat{\mathcal{N}}(\sigma_{k-j})\widehat{\mathcal{K}}(\sigma_{k-j})^{-1}\right)\\
&\quad\times\Big((\partial_s\widehat{\mathcal{N}})\widehat{\mathcal{K}}^{-1}\widehat{\mathcal{B}}-\widehat{\mathcal{N}}\widehat{\mathcal{K}}^{-1}(\partial_s\widehat{\mathcal{K}})\widehat{\mathcal{K}}^{-1}\widehat{\mathcal{B}}+\widehat{\mathcal{N}}\widehat{\mathcal{K}}^{-1}(\partial_s\widehat{\mathcal{B}})\Big)(\sigma_1)\\
&=: \hat{w}_{k-1}^{\mathsf{H}}(\partial_s\widehat{\mathcal{N}})(\sigma_1)\hat{v}_1-\hat{w}_k^{\mathsf{H}}(\partial_s\widehat{\mathcal{K}})(\sigma_1)\hat{v}_1+\hat{w}_k^{\mathsf{H}}(\partial_s\widehat{\mathcal{B}})(\sigma_1)\\
&= \hat{w}_{k-1}^{\mathsf{H}}W^{\mathsf{H}}(\partial_s\mathcal{N})(\sigma_1)V\hat{v}_1-\hat{w}_k^{\mathsf{H}}W^{\mathsf{H}}(\partial_s\mathcal{K})(\sigma_1)V\hat{v}_1+\hat{w}_k^{\mathsf{H}}W^{\mathsf{H}}(\partial_s\mathcal{B})(\sigma_1),
\end{aligned}$$

with $\hat{w}_{k-1}$, $\hat{w}_k$ and $\hat{v}_1$ vectors following the construction in Theorems 5.1 and 5.2 but with the reduced-order matrix functions. In fact, the same projectors P_{V} and P_{W} from (3.24) and (3.25) as in the proofs of Theorems 5.1 and 5.2 need to be constructed such that the following identities hold

$$V\hat{v}_1 = v_1, \quad W\hat{w}_{k-1} = w_{k-1}, \quad W\hat{w}_k = w_k, \tag{5.15}$$

using also the projection spaces $\operatorname{span}(V)$ and $\operatorname{span}(W)$. With (5.15), the formulation of the reduced-order transfer function yields the desired interpolation condition

$$\begin{aligned}
\partial_{s_1}\widehat{\mathcal{G}}_{\mathrm{B},k}(\sigma_1,\ldots,\sigma_k) &= w_{k-1}^{\mathsf{H}}(\partial_s\mathcal{N})(\sigma_1)v_1-w_k^{\mathsf{H}}(\partial_s\mathcal{K})(\sigma_1)v_1+w_k^{\mathsf{H}}(\partial_s\mathcal{B})(\sigma_1)\\
&= \mathcal{C}(\sigma_k)\mathcal{K}(\sigma_k)^{-1}\left(\prod_{j=1}^{k-2}\mathcal{N}(\sigma_{k-j})\mathcal{K}(\sigma_{k-j})^{-1}\right)\\
&\quad\times\Big((\partial_s\mathcal{N})\mathcal{K}^{-1}\mathcal{B}-\mathcal{N}\mathcal{K}^{-1}(\partial_s\mathcal{K})\mathcal{K}^{-1}\mathcal{B}+\mathcal{N}\mathcal{K}^{-1}(\partial_s\mathcal{B})\Big)(\sigma_1)\\
&= \mathcal{C}(\sigma_k)\mathcal{K}(\sigma_k)^{-1}\left(\prod_{j=1}^{k-2}\mathcal{N}(\sigma_{k-j})\mathcal{K}(\sigma_{k-j})^{-1}\right)\partial_s(\mathcal{N}\mathcal{K}^{-1}\mathcal{B})(\sigma_1)\\
&= \partial_{s_1}\mathcal{G}_{\mathrm{B},k}(\sigma_1,\ldots,\sigma_k).
\end{aligned}$$

As mentioned above, the same idea can be used for the other entries of the Jacobi matrix giving the proposed interpolation condition. □

As in the previous section, using two-sided projection allows to match interpolation conditions for a larger number of interpolation points and higher-level transfer functions. Following the results of Theorem 5.3 and using partial derivatives for the construction of the subspaces in the two-sided projection approach, it can be expected to match at least

$(k+\ell)+(\theta+\nu)+(k+\ell)\cdot(\theta+\nu)$ transfer function values and derivative evaluations, where k,ℓ relate to $\operatorname{span}(V)$ and θ,ν to $\operatorname{span}(W)$, and where $\ell=\ell_1+\ldots+\ell_k$ and $\nu=\nu_1+\ldots+\nu_\theta$ denote the orders of the partial derivatives and k,θ the maximum levels of the transfer functions to interpolate.

Theorem 5.7 (Bilinear Hermite interpolation by two-sided projection):
Let $\mathcal{G}_{\mathrm{B}}$ and $\widehat{\mathcal{G}}_{\mathrm{B}}$ be as in Theorem 5.4, let V be constructed as in Theorem 5.4 for a given sequence of interpolation points $\sigma_1,\ldots,\sigma_k\in\mathbb{C}$ and orders of partial derivatives $\ell_1,\ldots,\ell_k\in\mathbb{N}_0$, and let W be constructed as in Theorem 5.5 for another sequence of interpolation points $\varsigma_1,\ldots,\varsigma_\theta\in\mathbb{C}$ and orders of partial derivatives $\nu_1,\ldots,\nu_\theta\in\mathbb{N}_0$, for which the matrix functions $\mathcal{C},\mathcal{K}^{-1},\mathcal{N},\mathcal{B}$ and $\widehat{\mathcal{K}}^{-1}$ are complex differentiable. Then the regular subsystem transfer functions of $\widehat{\mathcal{G}}_{\mathrm{B}}$ interpolate those of $\mathcal{G}_{\mathrm{B}}$ in the following way:

$$\begin{aligned}
\partial_{s_1^{j_1}}\mathcal{G}_{\mathrm{B},1}(\sigma_1) &= \partial_{s_1^{j_1}}\widehat{\mathcal{G}}_{\mathrm{B},1}(\sigma_1), && j_1=0,\ldots,\ell_1,\\
&\vdots\\
\partial_{s_1^{\ell_1}\cdots s_{k-1}^{\ell_{k-1}}s_k^{j_k}}\mathcal{G}_{\mathrm{B},k}(\sigma_1,\ldots,\sigma_k) &= \partial_{s_1^{\ell_1}\cdots s_{k-1}^{\ell_{k-1}}s_k^{j_k}}\widehat{\mathcal{G}}_{\mathrm{B},k}(\sigma_1,\ldots,\sigma_k), && j_k=0,\ldots,\ell_k,\\
\partial_{s_1^{i_\theta}}\mathcal{G}_{\mathrm{B},1}(\varsigma_\theta) &= \partial_{s_1^{i_\theta}}\widehat{\mathcal{G}}_{\mathrm{B},1}(\varsigma_\theta), && i_\theta=0,\ldots,\nu_\theta,\\
&\vdots\\
\partial_{s_1^{i_1}s_2^{\nu_2}\cdots s_\theta^{\nu_\theta}}\mathcal{G}_{\mathrm{B},\theta}(\varsigma_1,\ldots,\varsigma_\theta) &= \partial_{s_1^{i_1}s_2^{\nu_2}\cdots s_\theta^{\nu_\theta}}\widehat{\mathcal{G}}_{\mathrm{B},\theta}(\varsigma_1,\ldots,\varsigma_\theta), && i_1=0,\ldots,\nu_1,
\end{aligned}$$

and, additionally,

$$\begin{aligned}
&\partial_{s_1^{\ell_1}\cdots s_{q-1}^{\ell_{q-1}}s_q^{j_q}s_{q+1}^{i_{\theta-\eta+1}}s_{q+2}^{\nu_{\theta-\eta+2}}\cdots s_{q+\eta}^{\nu_\theta}}\mathcal{G}_{\mathrm{B},q+\eta}(\sigma_1,\ldots,\sigma_q,\varsigma_{\theta-\eta+1},\ldots,\varsigma_\theta)\\
&=\partial_{s_1^{\ell_1}\cdots s_{q-1}^{\ell_{q-1}}s_q^{j_q}s_{q+1}^{i_{\theta-\eta+1}}s_{q+2}^{\nu_{\theta-\eta+2}}\cdots s_{q+\eta}^{\nu_\theta}}\widehat{\mathcal{G}}_{\mathrm{B},q+\eta}(\sigma_1,\ldots,\sigma_q,\varsigma_{\theta-\eta+1},\ldots,\varsigma_\theta)
\end{aligned}$$

holds for $j_q=0,\ldots,\ell_q$; $i_{\theta-\eta+1}=0,\ldots,\nu_{\theta-\eta+1}$; $1\le q\le k$ and $1\le\eta\le\theta$. ◊

Proof. The first part of the result just summarizes the theorems stating the one-sided projection approaches (Theorems 5.4 and 5.5), i.e., the mixed interpolation conditions involving both sequences of interpolation points are left to be proven. Let $j_q=0,\ldots,\ell_q$; $i_{\theta-\eta+1}=0,\ldots,\nu_{\theta-\eta+1}$; $2\le q\le k$ and $2\le\eta\le\theta$. The limit case of $q=1$ or $\eta=1$ has a deviating structure but can be treated analogously by taking also the input and output matrix functions $\mathcal{B}$ and $\mathcal{C}$ into account. The reduced-order transfer functions are then given by

$$\begin{aligned}
&\partial_{s_1^{\ell_1}\cdots s_{q-1}^{\ell_{q-1}}s_q^{j_q}s_{q+1}^{i_{\theta-\eta+1}}s_{q+2}^{\nu_{\theta-\eta+2}}\cdots s_{q+\eta}^{\nu_\theta}}\widehat{\mathcal{G}}_{\mathrm{B},q+\eta}(\sigma_1,\ldots,\sigma_q,\varsigma_{\theta-\eta+1},\ldots,\varsigma_\theta)\\
&=\partial_{s^{\nu_\theta}}(\widehat{\mathcal{C}}\widehat{\mathcal{K}}^{-1})(\varsigma_\theta)\cdots\partial_{s^{\nu_{\theta-\eta+2}}}(\widehat{\mathcal{N}}\widehat{\mathcal{K}}^{-1})(\varsigma_{\theta-\eta+2})\partial_{s^{i_{\theta-\eta+1}}}(\widehat{\mathcal{N}}\widehat{\mathcal{K}}^{-1})(\varsigma_{\theta-\eta+1})
\end{aligned}$$

$$\begin{aligned}
&\times \partial_{s^{j_q}}(\widehat{\mathcal{N}}\widehat{\mathcal{K}}^{-1})(\sigma_q)\partial_{s^{\ell_{q-1}}}(\widehat{\mathcal{N}}\widehat{\mathcal{K}}^{-1})(\sigma_{q-1})\cdots\partial_{s^{\ell_1}}(\widehat{\mathcal{N}}\widehat{\mathcal{K}}^{-1}\widehat{\mathcal{B}})(\sigma_1)\\
&=: \hat{w}^{\mathsf{H}}_{\eta,\nu_{\theta-\eta+1}}\partial_{s^{j_q}}(\widehat{\mathcal{N}}\widehat{\mathcal{K}}^{-1})(\sigma_q)\partial_{s^{\ell_{q-1}}}(\widehat{\mathcal{N}}\widehat{\mathcal{K}}^{-1})(\sigma_{q-1})\cdots\partial_{s^{\ell_1}}(\widehat{\mathcal{N}}\widehat{\mathcal{K}}^{-1}\widehat{\mathcal{B}})(\sigma_1).
\end{aligned}$$

Evaluating the partial derivative in the middle via the product rule yields

$$\partial_{s^{j_q}}(\widehat{\mathcal{N}}\widehat{\mathcal{K}}^{-1})(\sigma_q) = \sum_{\alpha=0}^{j_q} c_\alpha(\partial_{s^\alpha}\widehat{\mathcal{N}})(\sigma_q)(\partial_{s^{j_q-\alpha}}\widehat{\mathcal{K}}^{-1})(\sigma_q),$$

with appropriate constants $c_\alpha \in \mathbb{C}$. Therefore, it is possible to further rewrite the reduced-order transfer function into

$$\begin{aligned}
&\partial_{s_1^{\ell_1}\cdots s_{q-1}^{\ell_{q-1}} s_q^{j_q} s_{q+1}^{i_{\theta-\eta+1}} s_{q+2}^{\nu_{\theta-\eta+2}}\cdots s_{q+\eta}^{\nu_\theta}} \widehat{\mathcal{G}}_{\mathrm{B},q+\eta}(\sigma_1,\ldots,\sigma_q,\varsigma_{\theta-\eta+1},\ldots,\varsigma_\theta)\\
&=: \sum_{\alpha=0}^{j_q} c_\alpha \hat{w}^{\mathsf{H}}_{\eta,\nu_{\theta-\eta+1}}(\partial_{s^\alpha}\widehat{\mathcal{N}})(\sigma_q)\hat{v}_{q,j_q-\alpha}\\
&= \sum_{\alpha=0}^{j_q} c_\alpha \hat{w}^{\mathsf{H}}_{\eta,\nu_{\theta-\eta+1}} W^{\mathsf{H}}(\partial_{s^\alpha}\mathcal{N})(\sigma_q)V\hat{v}_{q,j_q-\alpha}.
\end{aligned}$$

As in previous proofs, the truncation matrices and underlying projection spaces are now used to show recursive identities for the constructed vectors using the projectors (3.24) and (3.25), i.e., it holds

$$V\hat{v}_{q,j_q-\alpha} = v_{q,j_q-\alpha}, \quad \text{and} \quad W\hat{w}_{\eta,\nu_{\theta-\eta+1}} = w_{\eta,\nu_{\theta-\eta+1}}, \quad \text{for all } 0 \le \alpha \le j_q.$$

Consequently, the mixed interpolation conditions in the theorem hold true:

$$\begin{aligned}
&\partial_{s_1^{\ell_1}\cdots s_{q-1}^{\ell_{q-1}} s_q^{j_q} s_{q+1}^{i_{\theta-\eta+1}} s_{q+2}^{\nu_{\theta-\eta+2}}\cdots s_{q+\eta}^{\nu_\theta}} \widehat{\mathcal{G}}_{\mathrm{B},q+\eta}(\sigma_1,\ldots,\sigma_q,\varsigma_{\theta-\eta+1},\ldots,\varsigma_\theta)\\
&= \sum_{\alpha=0}^{j_q} c_\alpha w^{\mathsf{H}}_{\eta,\nu_{\theta-\eta+1}}(\partial_{s^\alpha}\mathcal{N})(\sigma_q)v_{q,j_q-\alpha}\\
&= \partial_{s^{\nu_\theta}}(\mathcal{C}\mathcal{K}^{-1})(\varsigma_\theta)\cdots\partial_{s^{\nu_{\theta-\eta+2}}}(\mathcal{N}\mathcal{K}^{-1})(\varsigma_{\theta-\eta+2})\partial_{s^{i_{\theta-\eta+1}}}(\mathcal{N}\mathcal{K}^{-1})(\varsigma_{\theta-\eta+1})\\
&\quad\times \partial_{s^{j_q}}(\mathcal{N}\mathcal{K}^{-1})(\sigma_q)\partial_{s^{\ell_{q-1}}}(\mathcal{N}\mathcal{K}^{-1})(\sigma_{q-1})\cdots\partial_{s^{\ell_1}}(\mathcal{N}\mathcal{K}^{-1}\mathcal{B})(\sigma_1)\\
&= \partial_{s_1^{\ell_1}\cdots s_{q-1}^{\ell_{q-1}} s_q^{j_q} s_{q+1}^{i_{\theta-\eta+1}} s_{q+2}^{\nu_{\theta-\eta+2}}\cdots s_{q+\eta}^{\nu_\theta}} \mathcal{G}_{\mathrm{B},q+\eta}(\sigma_1,\ldots,\sigma_q,\varsigma_{\theta-\eta+1},\ldots,\varsigma_\theta). \qquad \square
\end{aligned}$$

For an easier understanding of Theorem 5.7, a small theoretical experiment is considered, where only the linear part is used choosing $k = \theta = 1$, with the interpolation points σ, ς and the orders of partial derivatives to be $\ell = \ell_1 = 2$ and $\nu = \nu_1 = 1$. Then, by the first part of Theorem 5.7, the interpolation of the following terms by means of $\operatorname{span}(V)$ is enforced:

$$\mathcal{G}_{\mathrm{B},1}(\sigma), \quad \partial_{s_1}\mathcal{G}_{\mathrm{B},1}(\sigma), \quad \partial_{s_1^2}\mathcal{G}_{\mathrm{B},1}(\sigma).$$

Similarly via $\operatorname{span}(W)$, the interpolation of

$$\mathcal{G}_{\mathrm{B},1}(\varsigma), \quad \partial_{s_1}\mathcal{G}_{\mathrm{B},1}(\varsigma)$$

is given. Using the two-sided projection approach, it is now possible to additionally match higher-level transfer functions and partial derivatives of these, namely

$$\begin{aligned} &\mathcal{G}_{\mathrm{B},2}(\sigma,\varsigma), \quad \partial_{s_1}\mathcal{G}_{\mathrm{B},2}(\sigma,\varsigma), \quad \partial_{s_2}\mathcal{G}_{\mathrm{B},2}(\sigma,\varsigma), \\ &\partial_{s_1^2}\mathcal{G}_{\mathrm{B},2}(\sigma,\varsigma), \quad \partial_{s_1 s_2}\mathcal{G}_{\mathrm{B},2}(\sigma,\varsigma), \quad \partial_{s_1^2 s_2}\mathcal{G}_{\mathrm{B},2}(\sigma,\varsigma). \end{aligned}$$

As already realized in Theorem 5.6, using the same set of interpolation points in the two-sided projection leads to additional interpolation of derivatives in an implicit way. This works analogously in combination with Theorem 5.7. The following corollary states this special case.

Corollary 5.8 (Implicit higher-order bilinear Hermite interpolation):
Assume $\mathcal{G}_{\mathrm{B}}$ and $\widehat{\mathcal{G}}_{\mathrm{B}}$ are constructed as in Theorem 5.7 for identical sets of interpolation points $\sigma_1 = \varsigma_1, \ldots, \sigma_k = \varsigma_k \in \mathbb{C}$ and matching orders of the partial derivatives $\ell_1 = \nu_1, \ldots, \ell_k = \nu_k \in \mathbb{N}_0$. Then additionally to the interpolation results of Theorem 5.7, it holds

$$\nabla\left(\partial_{s_1^{\ell_1}\cdots s_k^{\ell_k}}\mathcal{G}_{\mathrm{B},k}\right)(\sigma_1,\ldots,\sigma_k) = \nabla\left(\partial_{s_1^{\ell_1}\cdots s_k^{\ell_k}}\widehat{\mathcal{G}}_{\mathrm{B},k}\right)(\sigma_1,\ldots,\sigma_k). \qquad \lozenge$$

5.3.3 Numerical experiments

As illustration of the established interpolation theory for structured bilinear SISO systems, numerical experiments are performed for instances of the two example structures from Sections 5.2.3 and 5.2.4. Parts of the experiments will resemble the results published in [43]. Different variants of structured interpolation, following the suggestions for interpolation point selection in Section 3.3.4.2, will be compared to unstructured methods in the MORscore. Additionally to the approximate time domain measures (2.44) and (2.45) from Section 2.4.2, two frequency domain errors are computed based on the subsystem transfer functions of bilinear systems. For the first subsystem transfer functions, the classical approximate $\mathcal{H}_\infty/\mathcal{L}_\infty$-error (2.46) will be used and further denoted by $\mathcal{H}_\infty^{(1)}$, and for the second subsystem transfer functions, the $\mathcal{H}_\infty/\mathcal{L}_\infty$-error (2.46) is extended in the following sense

$$\|\mathcal{G} - \widehat{\mathcal{G}}\|_{\mathcal{H}_\infty/\mathcal{L}_\infty,2} \approx \max_{\omega_k,\omega_j} \|\mathcal{G}_{\mathrm{B},2}(\omega_k \mathbf{i}, \omega_j \mathbf{i}) - \widehat{\mathcal{G}}_{\mathrm{B},2}(\omega_k \mathbf{i}, \omega_j \mathbf{i})\|_2, \tag{5.16}$$

with discrete frequency evaluation points $\omega_k, \omega_j \in [\omega_{\min}, \omega_{\max}]$. The MORscores based on this error will be further denoted by $\mathcal{H}_\infty^{(2)}$. For additional illustration, a fixed reduced

order is selected for both examples and pointwise relative approximation errors are plotted. For the first subsystem transfer functions and the time domain simulations, (4.19) and (4.20) are used, respectively, and for the second subsystem transfer functions,

$$\epsilon_{\text{rel}}(\omega_1, \omega_2) := \frac{\|\mathcal{G}_{\text{B},2}(\omega_1 \mathbf{i}, \omega_2 \mathbf{i}) - \widehat{\mathcal{G}}_{\text{B},2}(\omega_1 \mathbf{i}, \omega_2 \mathbf{i})\|_2}{\|\mathcal{G}_{\text{B},2}(\omega_1 \mathbf{i}, \omega_2 \mathbf{i})\|_2} \tag{5.17}$$

is shown.

5.3.3.1 Bilinear mass-spring-damper system

The first example to be considered is an extension of the single chain oscillator from Section 4.2.5.1; see also [43, 142]. For the bilinearity, the springs are modeled to depend on the applied input force such that a displacement to the right increases the stiffness due to compression of the springs and to the left decreases it due to the appearing strain. The resulting bilinear mechanical system has the form

$$\begin{aligned} M\ddot{x}(t) + E\dot{x}(t) + Kx(t) &= N_{\text{p}}x(t)u(t) + B_{\text{u}}u(t), \\ y(t) &= C_{\text{p}}x(t), \end{aligned} \tag{5.18}$$

where M, E, K are chosen exactly as in Section 4.2.5.1, also with $n_2 = 10\,000$ masses. The bilinear term is constructed to be a scaled version of the stiffness matrix

$$N_{\text{p}} = -SKS,$$

where S is a diagonal matrix with linearly decaying entries `linspace(0.5,0,`n_2`)`. Input and output vectors are compressed versions of the inputs and outputs in Section 4.2.5.1, with

$$B_{\text{u}} = \begin{bmatrix} \mathbb{1}_5 \\ 0 \\ \vdots \\ 0 \\ \mathbb{1}_5 \end{bmatrix}, \quad C_{\text{p}} = \begin{pmatrix} e_1 + e_2 + e_3 + e_8 + e_9 + e_{10} + e_{n_2-2} + e_{n_2-1} + e_{n_2} \end{pmatrix}^{\mathsf{T}}.$$

The model reduction is performed with two general types of approaches: (i) the new structure-preserving bilinear interpolation, denoted by StrInt, (ii) the classical unstructured bilinear interpolation by converting (5.18) to first-order form (5.6), further on as FOInt. For both approaches, the first and second subsystem transfer functions are interpolated with the suggested interpolation point heuristics from the linear case in Section 3.3.4.2:

equi. denotes the simple choice of logarithmically equidistant interpolation points on the imaginary axis in complex conjugate pairs, where for both frequency arguments of the second subsystem transfer function the same interpolation point is chosen,

Table 5.1: MORscores for the bilinear mass-spring-damper example with reduced orders from 1 to 40.

Method	$\mathcal{H}_\infty^{(1)}$	$\mathcal{H}_\infty^{(2)}$	L_2	L_∞
StrInt(equi.)	0.1833	0.1365	0.1829	0.1841
StrInt($\mathcal{H}_\infty$)	0.2250	0.1625	0.1671	0.1660
StrInt(IRKA)	0.2285	0.1700	0.2030	0.2019
StrInt(avg.)	0.2738	0.2410	0.2478	0.2465
FOInt(equi.)	0.1097	0.0698	0.1109	0.1099
FOInt($\mathcal{H}_\infty$)	0.1713	0.0741	0.0893	0.0882
FOInt(IRKA)	0.1827	0.0819	0.0893	0.0889
FOInt(avg.)	0.1450	0.0890	0.0640	0.0645

$\mathcal{H}_\infty$ is an $\mathcal{H}_\infty$-greedy selection based on the first and second subsystem transfer function errors,

IRKA computes $\mathcal{H}_2$-optimal interpolation points via TF-IRKA and uses these for the first as well as second subsystem transfer function,

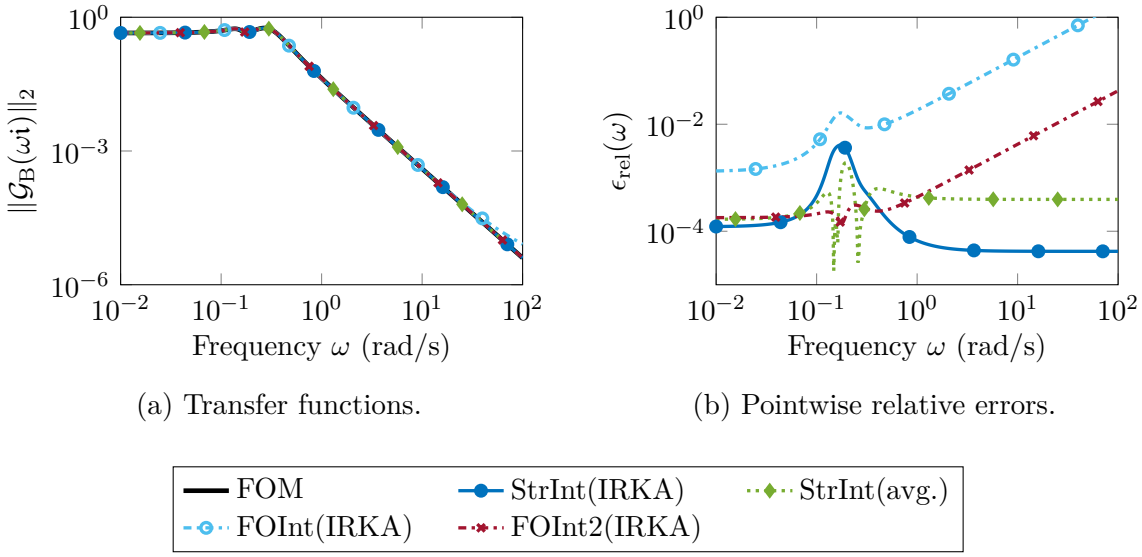
avg. is not an interpolation point selection but the averaged subspace approach from Remark 3.3 based on interpolation using samples from the first and second subsystem transfer functions in form of the input and output spaces.

To preserve even further mechanical properties of the bilinear mass-spring-damper system, only a one-sided projection is performed, i.e., in the three interpolation point selections, the reduced-order models are computed via Theorem 5.1 with $V = W$. The averaged subspace approach uses additionally Theorem 5.2 to compute both projection spaces, then concatenates the basis matrices into a single one and uses the pivoted QR decomposition to obtain a single truncation matrix of appropriate size for the one-sided projection.

The results in terms of the different MORscores for computing reduced-order models from order 1 to order 40 can be seen in Table 5.1. For the time domain MORscores, the systems have been simulated in the time interval $[0, 100]$ s using the input signal

$$u(t) = 10 \cdot \eta(t_j), \quad \text{for } t_j \leq t < t_{j+1}, \tag{5.19}$$

with $j = 0, \ldots, 99$, equidistant time steps $t_j = j \cdot \frac{100}{99}$ and presampled Gaussian white noise $\eta(t)$. In general, one can say that the structured interpolation performs exceptionally better than the unstructured approach in both frequency and time domains. The structured averaged subspace approach (StrInt(avg.)) is the best performing method of the full comparison, where StrInt(IRKA) is a strong competitor. This does not hold for



(a) Transfer functions.

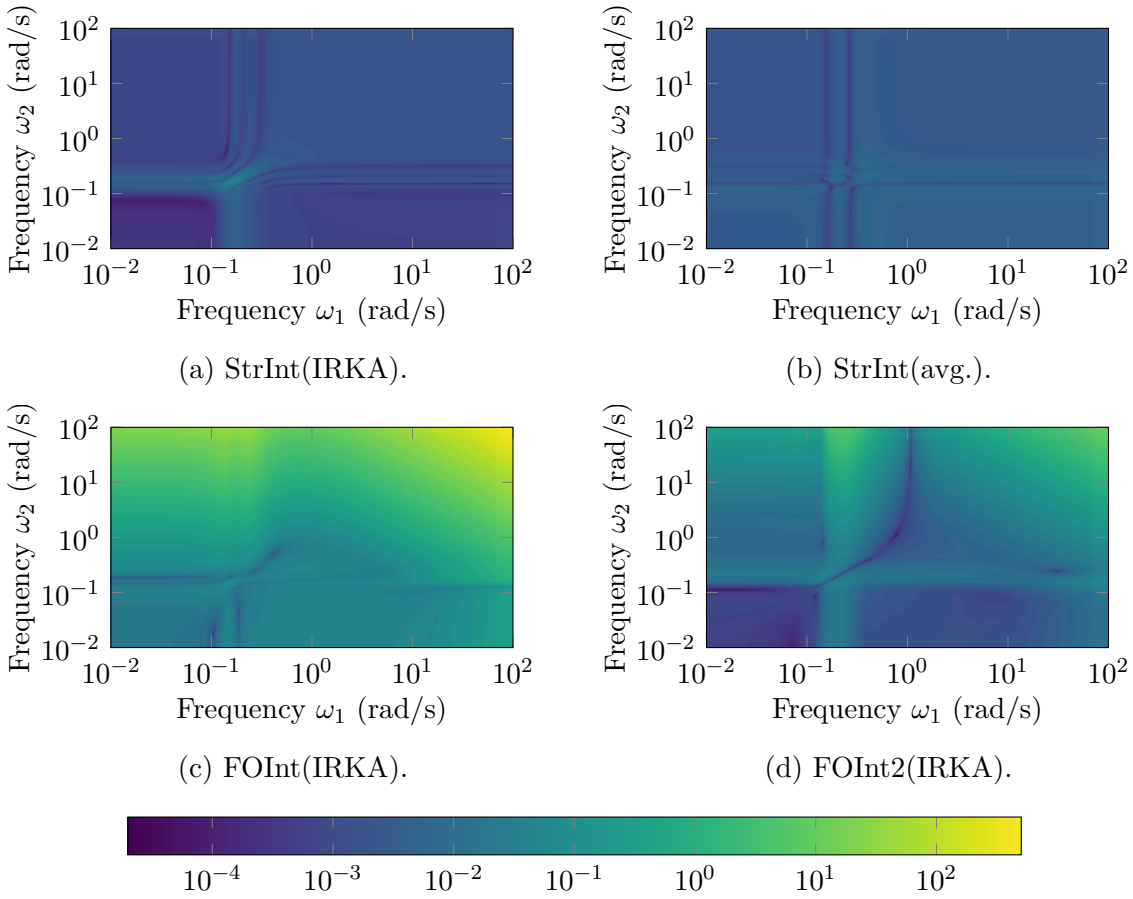
(b) Pointwise relative errors.

Figure 5.1: First subsystem transfer functions and approximation errors for the bilinear mass-spring-damper example.

the unstructured case, since except for the equidistant interpolation points, the other methods have a very small MORscore in the time domain measures. In frequency domain, from the unstructured methods the IRKA interpolation points perform best.

For a more detailed comparison of the methods, the reduced order is fixed to $r_2 = 12$. The methods with IRKA point selection are chosen and compared to StrInt(avg.) as the overall best performing method and an unstructured interpolation with IRKA points of double order FOInt2(IRKA). This additional comparison with FOInt2 is based on the observation that every bilinear mechanical system can be alternatively described by a first-order system of double order using, e.g., (5.6). The results for the first subsystem transfer functions are shown in Figure 5.1. Except for FOInt(IRKA), the other three methods behave very compatible to each other. The structured approaches have a slightly larger error in the middle of the frequency axis due to the changing behavior of the transfer function, which is nicely compensated in FOInt2(IRKA). But for increasing frequencies, the relative error of FOInt2(IRKA) is vastly growing while the structured approaches stay constantly small. The relative approximation errors of the second subsystem transfer functions are given in Figure 5.2. Here, FOInt2(IRKA) gives a comparably small error to the structured methods for small frequencies in both directions. Both structured approaches perform overall very well, where the errors of StrInt(IRKA) are usually smaller than for StrInt(avg.) except for a small region in the lower left area of the frequency plane.

Last, the time domain simulations for the chosen methods are shown in Figure 5.3. While all chosen methods perform stable for the given input signal (5.19), the approxi-



(a) StrInt(IRKA). (b) StrInt(avg.).

(c) FOInt(IRKA). (d) FOInt2(IRKA).

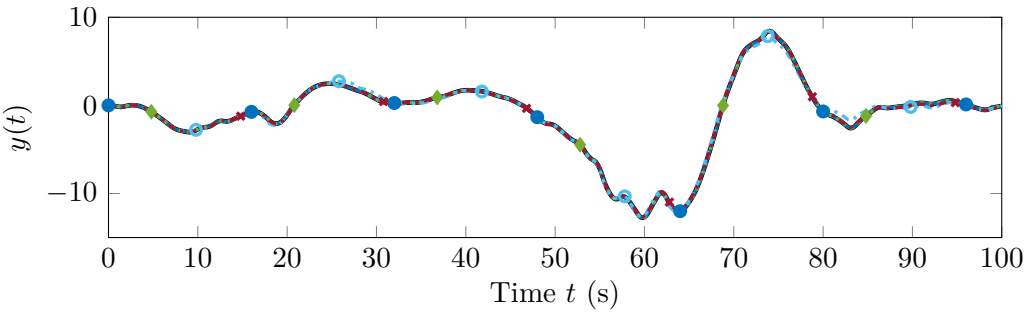
Figure 5.2: Relative approximation errors $\epsilon_{\text{rel}}(\omega_1, \omega_2)$ of the second subsystem transfer functions for the bilinear mass-spring-damper example.

mation quality shows significant differences. FOInt(IRKA) is not fully able to capture the behavior of the original system even in the eyeball-norm with visible deviations in Figure 5.3a. FOInt2(IRKA) and StrInt(avg.) are of comparable quality, but StrInt(IRKA) clearly performs best with exceptionally small relative errors in the beginning of the time simulation.

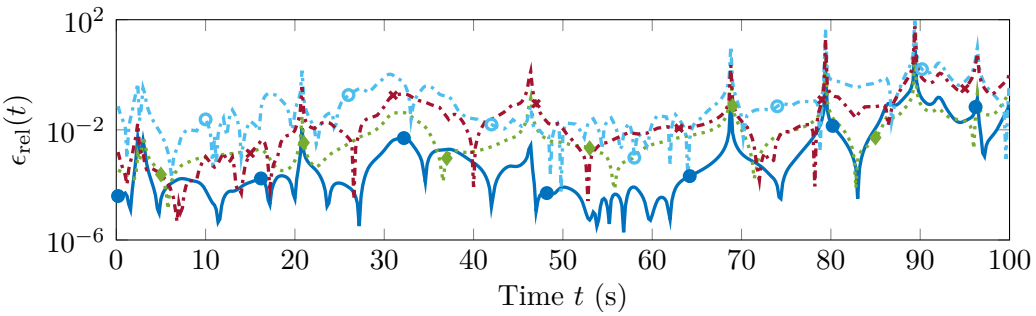
5.3.3.2 Time-delayed heated rod

As second numerical example, the bilinear time-delay system from [93] is considered. This example models a semi-discretized heated rod with distributed control and homogeneous Dirichlet boundary conditions, which is cooled by a delayed feedback and described by the partial differential equation

$$\partial_t v(\zeta, t) = \partial_{\zeta^2} v(\zeta, t) - 2\sin(\zeta)v(\zeta, t) + 2\sin(\zeta)v(\zeta, t-1) + u(t), \tag{5.20}$$



(a) Time simulation.



(b) Pointwise relative errors.

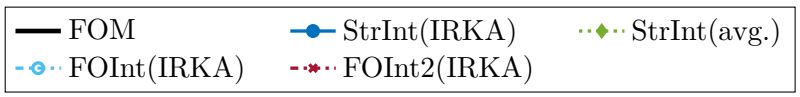


Figure 5.3: Time domain results for the bilinear mass-spring-damper example.

with $(\zeta, t) \in (0, \pi) \times (0, t_\mathrm{f})$ and boundary conditions $v(0, t) = v(\pi, t) = 0$ for $t \in [0, t_\mathrm{f}]$. A spatial discretization using centered finite differences results in a bilinear time-delay system of the form (5.10) with the time delay $\tau = 1$. For the experiments, $n = 5\,000$ is chosen.

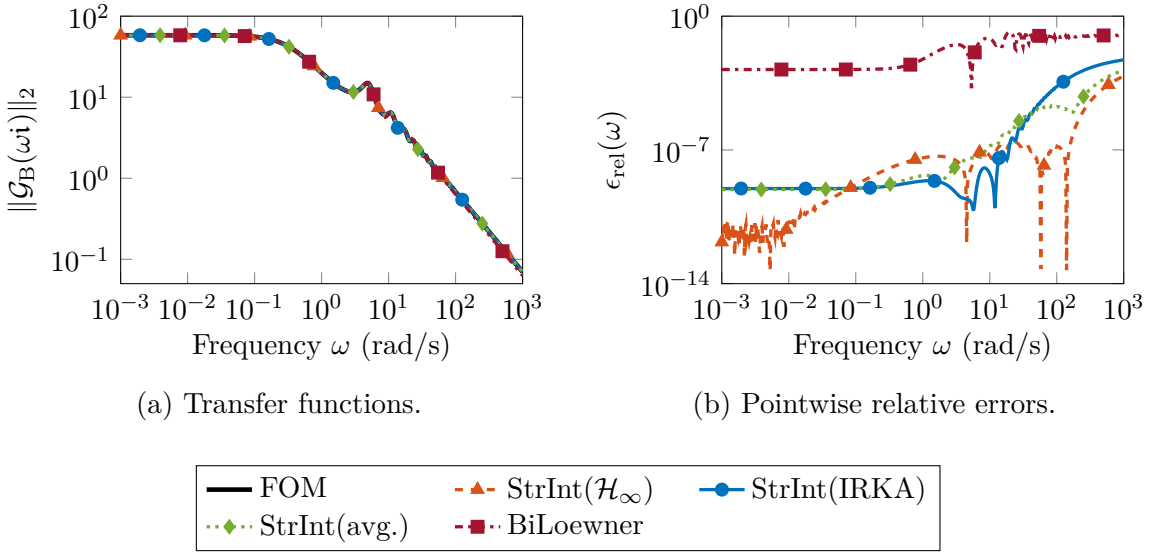
For the model reduction, the structured interpolation, StrInt, is used with the same choices of interpolation points (equi./$\mathcal{H}_\infty$/IRKA) as in the previous section, as well as the averaged subspace approach (avg.), for the first and second subsystem transfer functions. In this example, the two-sided projection approach is used based on Theorem 5.6. For comparison, the bilinear Loewner framework [12, 93], BiLoewner, is used to generate an unstructured bilinear system (2.27) without the time delay.

The resulting MORscores for reduced-order models from order 1 to order 30 are given in Table 5.2. For the simulations, the time interval $[0, 10]$ s is chosen with the input signal

$$u(t) = 0.05 \cdot \eta(t_j), \quad \text{for } t_j \leq t < t_{j+1}, \tag{5.21}$$

Table 5.2: MORscores for the time-delay example with reduced orders from 1 to 30.

Method	$\mathcal{H}_\infty^{(1)}$	$\mathcal{H}_\infty^{(2)}$	L_2	L_∞
StrInt(equi.)	0.4381	0.4932	0.4607	0.4472
StrInt($\mathcal{H}_\infty$)	0.4934	0.5075	0.4550	0.4474
StrInt(IRKA)	0.3850	0.4867	0.4656	0.4554
StrInt(avg.)	0.5092	0.5520	0.5213	0.5120
BiLoewner	0.1234	0.1188	0.0810	0.0663



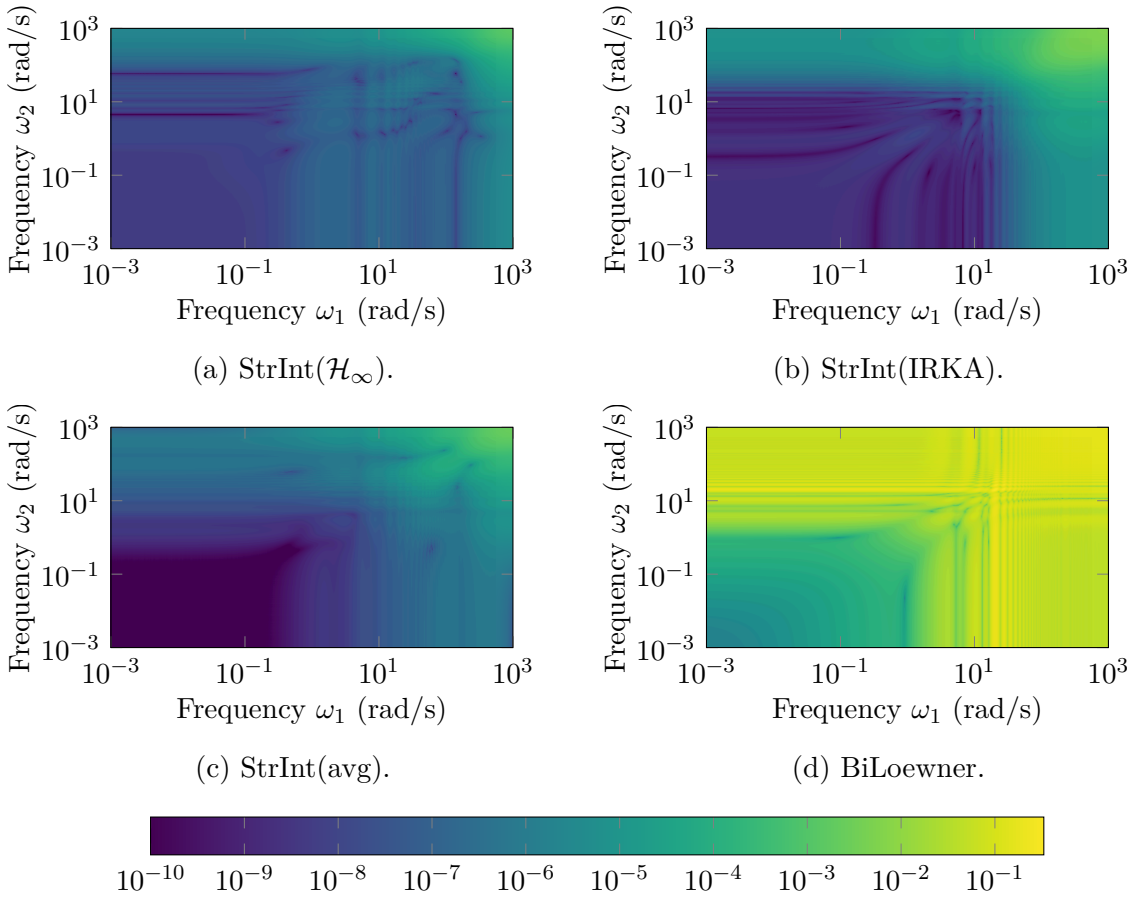
(a) Transfer functions.

(b) Pointwise relative errors.

Figure 5.4: First subsystem transfer functions and approximation errors for the time-delay example.

with $j = 0, \ldots, 9$, equidistant time steps $t_j = j \cdot \frac{10}{9}$ and presampled Gaussian white noise $\eta(t)$. The structured averaged subspace approach performs best, directly followed by the $\mathcal{H}_\infty$-greedy selection method StrInt($\mathcal{H}_\infty$). All structured methods perform pretty similar except for StrInt(IRKA) in the approximation of the first subsystem transfer function. Nevertheless, the structured approaches perform around 4 times better than the bilinear Loewner framework in frequency domain and between 6 and 8 times better in the time domain.

The large difference in the approximation quality becomes even clearer when considering a fixed reduced order, here $r_1 = 8$. In Figures 5.4 and 5.5, the frequency domain results are shown. BiLoewner fails in both figures to be compatible with the structured interpolation methods, which behave all very similar to StrInt($\mathcal{H}_\infty$), the clear winner. The same



(a) StrInt($\mathcal{H}_\infty$). (b) StrInt(IRKA).

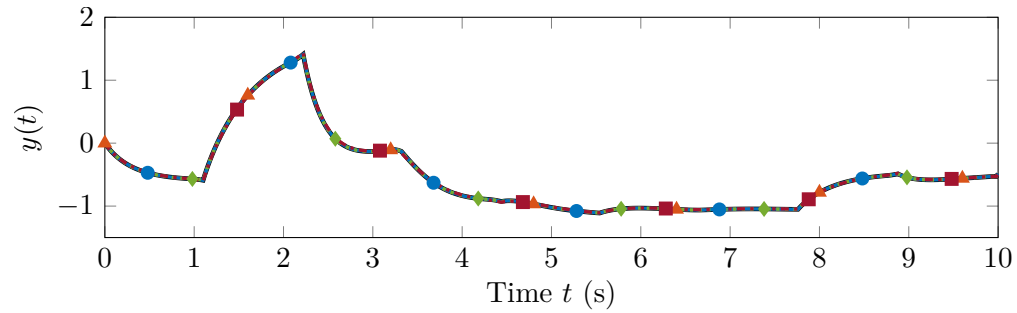
(c) StrInt(avg). (d) BiLoewner.

Figure 5.5: Relative approximation errors $\epsilon_{\text{rel}}(\omega_1, \omega_2)$ of the second subsystem transfer functions for the time-delay example.

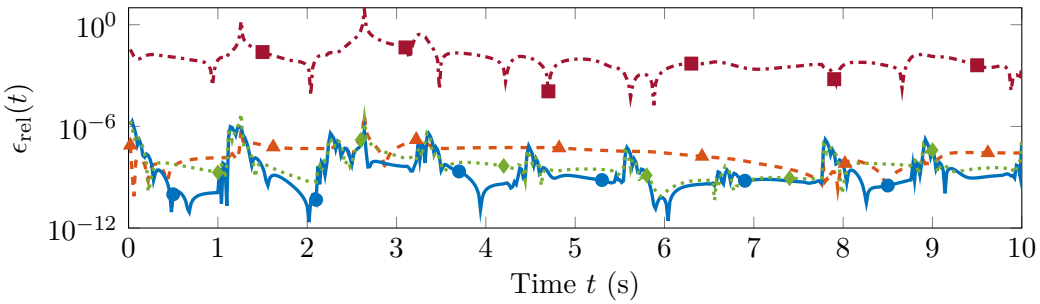
results can be seen in the time domain simulation in Figure 5.6. The bilinear Loewner framework performs several orders of magnitude worse than the structured interpolation methods. While the relative errors of the selected StrInt methods are in the same order of magnitude, their error behavior strongly differs. Thereby, StrInt($\mathcal{H}_\infty$) provides an overall very smooth and constant relative error, while StrInt(IRKA) provides a more spiky error that is sometimes smaller than StrInt($\mathcal{H}_\infty$) but also sometimes larger.

5.4 Matrix interpolation of multi-input/multi-output systems

In the previous section, the special case of SISO systems was treated to make use of the significantly simplified structure of the transfer function (5.12) with only a single bilinear



(a) Time simulation.



(b) Pointwise relative errors.

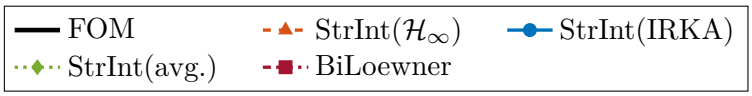


Figure 5.6: Time domain results for the time-delay example.

term. The more regularly occurring case in practice are MIMO systems, potentially involving several different bilinear terms. In principle, the generalization of the results in Section 5.3 to the MIMO case (5.3) is a straightforward procedure. However, one needs to realize first that for bilinear MIMO systems the quantities to be interpolated, i.e., the subsystem transfer functions, are matrix-valued with the column dimension increasing exponentially with the transfer function level. The main difference to the SISO system case in terms of formulae lies in the concatenation of the bilinear terms

$$\mathcal{N}(s) = \begin{bmatrix} \mathcal{N}_1(s) & \dots & \mathcal{N}_m(s) \end{bmatrix} \tag{5.22}$$

and the corresponding Kronecker products that produce different combinations of the linear and bilinear parts in the k-th subsystem transfer function; cf. (2.33). This section will only focus on matrix interpolation, i.e., the full matrix-valued structured subsystem transfer functions will be interpolated. The concept of tangential interpolation from the linear case [10, 89] is an efficient way to handle matrix-valued transfer functions by

restricting the interpolation to certain evaluation directions; see Section 3.3.2. There were attempts to generalize the definition of tangential interpolation to bilinear systems in [31, 160]. However, this topic will be discussed separately in more detail in Section 5.6.

For the results involving conditions on the left projection space $\operatorname{span}(W)$, a different concatenation of the bilinear terms than (5.22) is needed. Therefore, consider (5.22) to be the 1-mode matricization of the tensor-valued function $\boldsymbol{\mathcal{N}}\colon \mathbb{C} \to \mathbb{C}^{n\times n\times m}$ given by

$$\mathcal{N}(s) = \boldsymbol{\mathcal{N}}^{(1)}(s).$$

In the upcoming theory, the 2-mode matricization of this tensor function is needed, which is given by

$$\boldsymbol{\mathcal{N}}^{(2)}(s) = \begin{bmatrix} \mathcal{N}_1(s)^{\mathsf{T}} & \dots & \mathcal{N}_m(s)^{\mathsf{T}} \end{bmatrix}. \tag{5.23}$$

See Section 2.1.1 for more details on tensors and matricizations. The following theorem extends the results from Theorems 5.1 to 5.3 to structured bilinear MIMO systems.

Theorem 5.9 (Bilinear matrix interpolation):
Let $\mathcal{G}_{\mathrm{B}}$ be a bilinear system, described by (5.3), and $\widehat{\mathcal{G}}_{\mathrm{B}}$ the reduced-order bilinear system constructed by (5.4), with its subsystem transfer functions $\widehat{\mathcal{G}}_{\mathrm{B},k}$ in (5.5). Given sets of interpolation points $\sigma_1, \dots, \sigma_k \in \mathbb{C}$ and $\varsigma_1, \dots, \varsigma_\theta \in \mathbb{C}$, for which the matrix functions $\mathcal{C}$, $\mathcal{K}^{-1}$, $\mathcal{N}$, $\mathcal{B}$ and $\widehat{\mathcal{K}}^{-1}$ are defined, the following statements hold:

(a) If V is constructed as

$$\begin{aligned} V_1 &= \mathcal{K}(\sigma_1)^{-1}\mathcal{B}(\sigma_1), \\ V_j &= \mathcal{K}(\sigma_j)^{-1}\mathcal{N}(\sigma_{j-1})(I_m \otimes V_{j-1}), && 2 \le j \le k, \\ \operatorname{span}(V) &\supseteq \operatorname{span}\left(\begin{bmatrix} V_1 & \dots & V_k \end{bmatrix}\right), \end{aligned}$$

then the following interpolation conditions hold true:

$$\mathcal{G}_{\mathrm{B},j}(\sigma_1, \dots, \sigma_j) = \widehat{\mathcal{G}}_{\mathrm{B},j}(\sigma_1, \dots, \sigma_j),$$

for $j = 1, \dots, k$.

(b) If W is constructed as

$$\begin{aligned} W_1 &= \mathcal{K}(\varsigma_\theta)^{-\mathsf{H}}\mathcal{C}(\varsigma_\theta)^{\mathsf{H}}, \\ W_i &= \mathcal{K}(\varsigma_{\theta-i+1})^{-\mathsf{H}}\overline{\boldsymbol{\mathcal{N}}^{(2)}(\varsigma_{k-i+1})}(I_m \otimes W_{i-1}), && 2 \le i \le \theta, \\ \operatorname{span}(W) &\supseteq \operatorname{span}\left(\begin{bmatrix} W_1, \dots, W_\theta \end{bmatrix}\right), \end{aligned}$$

where $\boldsymbol{\mathcal{N}}^{(2)}$ is the 2-mode matricization of the tensor defined by $\boldsymbol{\mathcal{N}}^{(1)} = \mathcal{N}$ like in (5.23), then the following interpolation conditions hold true:

$$\mathcal{G}_{\mathrm{B},i}(\varsigma_{\theta-i+1}, \dots, \varsigma_\theta) = \widehat{\mathcal{G}}_{\mathrm{B},i}(\varsigma_{\theta-i+1}, \dots, \varsigma_\theta),$$

for $i = 1, \dots, \theta$.

(c) Let V be constructed as in Part (a) and W as in Part (b), then, additionally to the results in (a) and (b), the interpolation conditions

$$\mathcal{G}_{\mathrm{B},q+\eta}(\sigma_1,\ldots,\sigma_q,\varsigma_{\theta-\eta+1},\ldots,\varsigma_\theta) = \widehat{\mathcal{G}}_{\mathrm{B},q+\eta}(\sigma_1,\ldots,\sigma_q,\varsigma_{\theta-\eta+1},\ldots,\varsigma_\theta)$$

hold, for $1 \leq q \leq k$ and $1 \leq \eta \leq \theta$. ◊

Proof. As mentioned before, large parts of the proof directly follow from the ideas in the SISO case. First, consider Part (a). Let $\mathcal{G}_{\mathrm{B},j,\alpha}$, with $1 \leq j \leq k$ and $1 \leq \alpha \leq m^j$, denote a block entry of the transfer function $\mathcal{G}_{\mathrm{B},j}$ with evaluated Kronecker products, i.e.,

$$\mathcal{G}_{\mathrm{B},j,\alpha}(s_1,\ldots,s_j) := \mathcal{C}(s_j)\mathcal{K}(s_j)^{-1}\mathcal{N}_{\alpha_{j-1}}(s_{j-1})\mathcal{K}(s_{j-1})^{-1}\cdots\mathcal{N}_{\alpha_1}(s_1)\mathcal{K}(s_1)^{-1}\mathcal{B}(s_1), \quad (5.24)$$

where the indices $1 \leq \alpha_1,\ldots,\alpha_{j-1} \leq m$ denote any appropriate combination of the bilinear terms. In the way of Theorem 5.1, it is now possible to analogously construct the projector P_{V} from (3.24) onto $\operatorname{span}(V)$. Therefore, consider the reduced-order version of (5.24) in the interpolation points

$$\widehat{\mathcal{G}}_{\mathrm{B},j,\alpha}(\sigma_1,\ldots,\sigma_j) = \widehat{\mathcal{C}}(\sigma_j)\widehat{\mathcal{K}}(\sigma_j)^{-1}\widehat{\mathcal{N}}_{\alpha_{j-1}}(\sigma_{j-1})\widehat{\mathcal{K}}(\sigma_{j-1})^{-1}\cdots\widehat{\mathcal{N}}_{\alpha_1}(\sigma_1)\widehat{\mathcal{K}}(\sigma_1)^{-1}\widehat{\mathcal{B}}(\sigma_1).$$

Considering (5.24) and the entry of the reduced-order transfer function column-wise , it can be seen that the interpolation conditions hold exactly as before in a recursive way using the single columns of $V_1,\ldots,V_j$, i.e., it holds

$$\widehat{\mathcal{G}}_{\mathrm{B},j,\alpha}(\sigma_1,\ldots,\sigma_j) = \mathcal{G}_{\mathrm{B},j,\alpha}(\sigma_1,\ldots,\sigma_j)$$

for all $1 \leq j \leq k$ and $1 \leq \alpha \leq m^j$, giving the result in Part (a).

Parts (b) and (c) work analogously using Theorems 5.2 and 5.3, where in the construction of the matrices $W_1,\ldots,W_k$ the 2-mode matricization of the bilinear concatenation (5.23) is used as complex conjugate such that the single matrix-valued entries of (5.23) are the Hermitian transposed matrix functions of the original bilinear terms

$$\overline{\boldsymbol{\mathcal{N}}^{(2)}(\varsigma_{k-i+1})} = \begin{bmatrix}\mathcal{N}_1(s)^{\mathsf{H}} & \ldots & \mathcal{N}_m(s)^{\mathsf{H}}\end{bmatrix}.$$

This leads to an analogue to the proof of Part (a). □

In a similar fashion, an extension for the Hermite interpolation results in Theorems 5.4, 5.5 and 5.7 to the MIMO case is given below.

Theorem 5.10 (Bilinear Hermite matrix interpolation):
Let $\mathcal{G}_{\mathrm{B}}$ be a bilinear system, described by (5.3), and $\widehat{\mathcal{G}}_{\mathrm{B}}$ the reduced-order bilinear system constructed by (5.4), with its subsystem transfer functions $\widehat{\mathcal{G}}_{\mathrm{B},k}$ in (5.5). Given sets of interpolation points $\sigma_1,\ldots,\sigma_k \in \mathbb{C}$ and $\varsigma_1,\ldots,\varsigma_\theta \in \mathbb{C}$, for which the matrix functions $\mathcal{C}$, $\mathcal{K}^{-1}$, $\mathcal{N}$, $\mathcal{B}$ and $\widehat{\mathcal{K}}^{-1}$ are complex differentiable, and orders of partial derivatives $\ell_1,\ldots,\ell_k \in \mathbb{N}_0$ and $\nu_1,\ldots,\nu_\theta \in \mathbb{N}_0$, then the following statements hold:

(a) If V is constructed as

$$\begin{aligned}
V_{1,j_1} &= \partial_{s^{j_1}}(\mathcal{K}^{-1}\mathcal{B})(\sigma_1), && j_1 = 0,\dots,\ell_1,\\
V_{2,j_2} &= \partial_{s^{j_2}}\mathcal{K}^{-1}(\sigma_2)\partial_{s^{\ell_1}}\Big(\mathcal{N}(I_m\otimes\mathcal{K}^{-1}\mathcal{B})\Big)(\sigma_1), && j_2 = 0,\dots,\ell_2,\\
&\vdots\\
V_{k,j_k} &= \partial_{s^{j_k}}\mathcal{K}^{-1}(\sigma_k)\\
&\quad\times\left(\prod_{j=1}^{k-2}\partial_{s^{\ell_{k-j}}}\Big((I_{m^{j-1}}\otimes\mathcal{N})(I_{m^j}\otimes\mathcal{K})\Big)(\sigma_{k-j})\right)\\
&\quad\times\partial_{s^{\ell_1}}\Big((I_{m^{k-2}}\otimes\mathcal{N})(I_{m^{k-1}}\otimes\mathcal{K})(I_{m^{k-1}}\otimes\mathcal{B})\Big)(\sigma_1), && j_k = 0,\dots,\ell_k,\\
\operatorname{span}(V) &\supseteq \operatorname{span}\left(\begin{bmatrix}V_{1,0} & \dots & V_{k,\ell_k}\end{bmatrix}\right),
\end{aligned}$$

then the following interpolation conditions hold true:

$$\partial_{s_1^{\ell_1}\cdots s_{q-1}^{\ell_{q-1}}s_q^{j_q}}\mathcal{G}_{\mathrm{B},q}(\sigma_1,\dots,\sigma_q) = \partial_{s_1^{\ell_1}\cdots s_{q-1}^{\ell_{q-1}}s_q^{j_q}}\widehat{\mathcal{G}}_{\mathrm{B},q}(\sigma_1,\dots,\sigma_q),$$

for $q = 1,\dots,k$ and $j_q = 0,\dots\ell_q$.

(b) If W is constructed as

$$\begin{aligned}
W_{1,i_\theta} &= \partial_{s^{i_\theta}}(\mathcal{K}^{-\mathsf{H}}\mathcal{C}^{\mathsf{H}})(\varsigma_\theta), && i_\theta = 0,\dots,\nu_\theta,\\
W_{2,i_{\theta-1}} &= \partial_{s^{i_{\theta-1}}}(\mathcal{K}^{-\mathsf{H}}\overline{\boldsymbol{\mathcal{N}}^{(2)}})(\varsigma_{\theta-1})\Big(I_m\otimes W_{1,\nu_\theta}\Big), && i_{\theta-1} = 0,\dots,\nu_{\theta-1},\\
&\vdots\\
W_{\theta,i_1} &= \partial_{s^{i_1}}(\mathcal{K}^{-\mathsf{H}}\overline{\boldsymbol{\mathcal{N}}^{(2)}})(\varsigma_1)\Big(I_m\otimes W_{\theta-1,\nu_2}\Big), && i_1 = 0,\dots,\nu_1,\\
\operatorname{span}(W) &\supseteq \operatorname{span}\left(\begin{bmatrix}W_{1,0} & \dots & W_{\theta,\nu_\theta}\end{bmatrix}\right),
\end{aligned}$$

where $\boldsymbol{\mathcal{N}}^{(2)}$ is the 2-mode matricization of the tensor defined by $\boldsymbol{\mathcal{N}}^{(1)} = \mathcal{N}$ like in (5.23), then the following interpolation conditions hold true:

$$\partial_{s_1^{i_{\theta-\eta+1}}s_2^{\nu_{\theta-\eta+2}}\cdots s_\eta^{\nu_\theta}}\mathcal{G}_{\mathrm{B},\eta}(\varsigma_{\theta-\eta+1},\dots,\varsigma_\theta) = \partial_{s_1^{i_{\theta-\eta+1}}s_2^{\nu_{\theta-\eta+2}}\cdots s_\eta^{\nu_\theta}}\widehat{\mathcal{G}}_{\mathrm{B},\eta}(\varsigma_{\theta-\eta+1},\dots,\varsigma_\theta),$$

for $\eta = 1,\dots,\theta$ and $i_{\theta-\eta+1} = 0,\dots,\nu_{\theta-\eta+1}$.

(c) Let V be constructed as in Part (a) and W as in Part (b), then, additionally to the results in (a) and (b), the interpolation conditions

$$\begin{aligned}
&\partial_{s_1^{\ell_1}\cdots s_{q-1}^{\ell_{q-1}}s_q^{j_q}s_{q+1}^{i_{\theta-\eta+1}}s_{q+2}^{\nu_{\theta-\eta+2}}\cdots s_{q+\eta}^{\nu_\theta}}\mathcal{G}_{\mathrm{B},q+\eta}(\sigma_1,\dots,\sigma_q,\varsigma_{\theta-\eta+1},\dots,\varsigma_\theta)\\
&= \partial_{s_1^{\ell_1}\cdots s_{q-1}^{\ell_{q-1}}s_q^{j_q}s_{q+1}^{i_{\theta-\eta+1}}s_{q+2}^{\nu_{\theta-\eta+2}}\cdots s_{q+\eta}^{\nu_\theta}}\widehat{\mathcal{G}}_{\mathrm{B},q+\eta}(\sigma_1,\dots,\sigma_q,\varsigma_{\theta-\eta+1},\dots,\varsigma_\theta)
\end{aligned}$$

hold, for $j_q = 0,\dots,\ell_q$; $i_{\theta-\eta+1} = 0,\dots,\nu_{\theta-\eta+1}$; $1 \le q \le k$ and $1 \le \eta \le \theta$. ◊

Proof. The results follow directly by using the ideas from the proof of Theorem 5.9 for the MIMO case and the results from Theorems 5.4, 5.5 and 5.7 about structured Hermite interpolation. □

As in the SISO case, implicit interpolation of partial derivatives of the transfer functions by two-sided projection is possible by using identical sequences of interpolation points for the construction of left and right projection spaces in Theorems 5.9 and 5.10. This is summarized in the following corollary without additional proofs.

Corollary 5.11 (Implicit bilinear matrix interpolation):
Let $\mathcal{G}_{\mathrm{B}}$ be a bilinear system, described by (5.3), and $\widehat{\mathcal{G}}_{\mathrm{B}}$ the reduced-order bilinear system constructed by (5.4), with its subsystem transfer functions $\widehat{\mathcal{G}}_{\mathrm{B},k}$ in (5.5). Given a set of interpolation points $\sigma_1, \ldots, \sigma_k \in \mathbb{C}$, for which the matrix functions $\mathcal{C}$, $\mathcal{K}^{-1}$, $\mathcal{N}$, $\mathcal{B}$ and $\widehat{\mathcal{K}}^{-1}$ are complex differentiable, the following statements hold:

(a) Let V and W be constructed as in Theorem 5.9 Parts (a) and (b) for a matching sequence of interpolation points $\sigma_1 = \varsigma_1, \ldots, \sigma_k = \varsigma_k$, then, additionally to the results in Theorem 5.9, it holds

$$\nabla \mathcal{G}_{\mathrm{B},k}(\sigma_1, \ldots, \sigma_k) = \nabla \widehat{\mathcal{G}}_{\mathrm{B},k}(\sigma_1, \ldots, \sigma_k).$$

(b) Let V and W be constructed as in Theorem 5.10 Parts (a) and (b) for a matching sequence of interpolation points $\sigma_1 = \varsigma_1, \ldots, \sigma_k = \varsigma_k$ and matching orders of partial derivatives $\ell_1 = \nu_1, \ldots, \ell_k = \nu_k$, then, additionally to the results in Theorem 5.10, it holds

$$\nabla\left(\partial_{s_1^{\ell_1} \cdots s_k^{\ell_k}} \mathcal{G}_{\mathrm{B},k}\right)(\sigma_1, \ldots, \sigma_k) = \nabla\left(\partial_{s_1^{\ell_1} \cdots s_k^{\ell_k}} \widehat{\mathcal{G}}_{\mathrm{B},k}\right)(\sigma_1, \ldots, \sigma_k). \qquad \lozenge$$

For brevity and prevention of repetitions, numerical experiments for the matrix interpolation theory presented in this section are shown later in Section 5.6. The matrix interpolation approach is then compared to a newly developed framework for structured tangential interpolation of bilinear MIMO systems.

5.5 Extension to parametric structured bilinear systems

An important system class extension, when thinking about real-world applications, are bilinear time-invariant systems with additional parameter dependencies. As the system structure itself, parameter dependencies are usually modeled with a physical interpretation, allowing to use a similar mathematical descriptions for different system realizations, e.g., in case of material coefficients as parameters. Going back to the

motivating example of mechanical bilinear systems (5.1) from the introduction of this chapter, its parametric version can be written as

$$\begin{aligned} 0 &= M(\mu)\ddot{x}(t;\mu) + E(\mu)\dot{x}(t;\mu) + K(\mu)x(t;\mu) \\ &\quad - B_{\mathrm{u}}(\mu)u(t) - \sum_{j=1}^{m} N_{\mathrm{p},j}(\mu)x(t;\mu)u_j(t) - \sum_{j=1}^{m} N_{\mathrm{v},j}(\mu)\dot{x}(t;\mu)u_j(t), \\ y(t;\mu) &= C_{\mathrm{p}}(\mu)x(t;\mu) + C_{\mathrm{v}}(\mu)\dot{x}(t;\mu), \end{aligned} \tag{5.25}$$

where $M(\mu)$, $E(\mu)$, $K(\mu)$, $N_{\mathrm{p},j}(\mu)$, $N_{\mathrm{v},j}(\mu) \in \mathbb{R}^{n_2\times n_2}$, for $j = 1,\ldots,m$; $B_{\mathrm{u}}(\mu) \in \mathbb{R}^{n_2\times m}$ and $C_{\mathrm{p}}(\mu), C_{\mathrm{v}}(\mu) \in \mathbb{R}^{p\times n_2}$ are constant matrices; and $\mu \in \mathbb{M} \subset \mathbb{R}^d$ is the collection of the time-invariant parameters affecting the dynamics. The parameter μ may represent variations in, e.g., material properties or system geometry.

The aim of structure-preserving parametric model order reduction is in principle the same as in structure-preserving model reduction to construct a cheap-to-evaluate approximation of the input-to-output behavior of the original system by reducing the state-space dimension while additionally the internal system structure and even the parameter dependencies are preserved in the reduced-order model to retain the underlying physical structure and its interpretation. For example, for the system (5.25), the structure-preserving parametric reduced-order model will have the form

$$\begin{aligned} 0 &= \widehat{M}(\mu)\ddot{\hat{x}}(t;\mu) + \widehat{E}(\mu)\dot{\hat{x}}(t;\mu) + \widehat{K}(\mu)\hat{x}(t;\mu) \\ &\quad - \widehat{B}_{\mathrm{u}}(\mu)u(t) - \sum_{j=1}^{m} \widehat{N}_{\mathrm{p},j}(\mu)\hat{x}(t;\mu)u_j(t) - \sum_{j=1}^{m} \widehat{N}_{\mathrm{v},j}(\mu)\dot{\hat{x}}(t;\mu)u_j(t), \\ \hat{y}(t;\mu) &= \widehat{C}_{\mathrm{p}}(\mu)\hat{x}(t;\mu) + \widehat{C}_{\mathrm{v}}(\mu)\dot{\hat{x}}(t;\mu), \end{aligned}$$

with $\widehat{M}(\mu), \widehat{E}(\mu), \widehat{K}(\mu), \widehat{N}_{\mathrm{p},j}(\mu), \widehat{N}_{\mathrm{v},j}(\mu) \in \mathbb{R}^{r_2\times r_2}$, for $j = 1,\ldots,m$, $\widehat{B}_{\mathrm{u}}(\mu) \in \mathbb{R}^{r_2\times m}$, $\widehat{C}_{\mathrm{p}}(\mu), \widehat{C}_{\mathrm{v}}(\mu) \in \mathbb{R}^{p\times r_2}$, and $r_2 \ll n_2$.

For parametric unstructured (classical) bilinear systems, i.e., for systems of the form

$$\begin{aligned} \mathsf{E}(\mu)\dot{\mathsf{x}}(t;\mu) &= \mathsf{A}(\mu)\mathsf{x}(t;\mu) + \mathsf{B}(\mu)u(t) + \sum_{j=1}^{m} \mathsf{N}_j(\mu)\mathsf{x}(t;\mu)u_j(t), \\ y(t;\mu) &= \mathsf{C}(\mu)\mathsf{x}(t;\mu), \end{aligned} \tag{5.26}$$

an interpolatory parametric model reduction framework was developed in [160] by synthesizing the interpolation theory for parametric linear dynamical systems [10, 21] with the subsystem interpolation approaches for bilinear systems [10, 15, 65, 72, 81]. In a similar fashion, the structured interpolation theory from Sections 5.3 and 5.4 can be extended to the parametric system case. The following subsections describe the extension of the structured subsystem transfer functions (5.3) to parametric systems and, thereafter, subspace conditions to enforce transfer function interpolation in frequency and parameter points. Large parts of this section are published in [42].

5.5.1 Parametric structured subsystem transfer functions

Since the parameter $\mu \in \mathbb{M}$ is considered to be constant in time, the parametric system case resembles the non-parametric one for any chosen parameter configuration μ, i.e., the structured subsystem transfer functions (5.3) can be directly extended to the parametric setting:

$$\begin{aligned}\mathcal{G}_{\mathrm{B},k}(s_1,\ldots,s_k,\mu) = \mathcal{C}(s_k,\mu)\mathcal{K}(s_k,\mu)^{-1} \Bigg(\prod_{j=1}^{k-1} \Big(I_{m^{j-1}} \otimes \mathcal{N}(s_{k-j},\mu)\Big) \\ \times \Big(I_{m^j} \otimes \mathcal{K}(s_{k-j},\mu)^{-1}\Big)\Bigg) \Big(I_{m^{k-1}} \otimes \mathcal{B}(s_1,\mu)\Big),\end{aligned} \tag{5.27}$$

where the matrix-valued functions describing the different parts of the system dynamics are now multivariate with the additional dependency on the parameter configuration $\mu \in \mathbb{M}$. Therefore, in this and the following subsections belonging to Section 5.5, the matrix-valued functions are considered to be $\mathcal{C}\colon \mathbb{C}\times\mathbb{M} \to \mathbb{C}^{p\times n}$, $\mathcal{K}\colon \mathbb{C}\times\mathbb{M} \to \mathbb{C}^{n\times n}$, $\mathcal{B}\colon \mathbb{C}\times\mathbb{M} \to \mathbb{C}^{n\times m}$, $\mathcal{N}_j\colon \mathbb{C}\times\mathbb{M} \to \mathbb{C}^{n\times n}$, for $j = 1,\ldots,m$, with the column concatenation of the bilinear terms $\mathcal{N}(s,\mu) = \begin{bmatrix}\mathcal{N}_1(s,\mu) & \ldots & \mathcal{N}_m(s,\mu)\end{bmatrix}$, such that $\mathcal{G}_{\mathrm{B},k}\colon \mathbb{C}^k \times \mathbb{M} \to \mathbb{C}^{p\times m^k}$. For parametric bilinear first-order systems (5.26), the matrix functions are realized by

$$\mathcal{C}(s,\mu) = \mathsf{C}(\mu), \quad \mathcal{K}(s,\mu) = s\mathsf{E}(\mu) - \mathsf{A}(\mu), \quad \mathcal{B}(s,\mu) = \mathsf{B}(\mu), \quad \mathcal{N}_j(s,\mu) = \mathsf{N}_j(\mu),$$

and in case of parametric bilinear mechanical systems (5.25) by

$$\begin{aligned}\mathcal{C}(s,\mu) &= C_{\mathrm{p}}(\mu) + sC_{\mathrm{v}}(\mu), & \mathcal{K}(s,\mu) &= s^2M(\mu) + sE(\mu) + K(\mu),\\ \mathcal{B}(s,\mu) &= B_{\mathrm{u}}(\mu), & \mathcal{N}_j(s,\mu) &= N_{\mathrm{p},j}(\mu) + sN_{\mathrm{v},j}(\mu),\end{aligned}$$

with $j = 1,\ldots,m$.

The very same projection approach (5.4) as for non-parametric bilinear systems is used to compute the reduced-order matrix functions for parametric systems by

$$\begin{aligned}\widehat{\mathcal{C}}(s,\mu) &= \mathcal{C}(s,\mu)V, & \widehat{\mathcal{K}}(s,\mu) &= W^{\mathsf{H}}\mathcal{K}(s,\mu)V,\\ \widehat{\mathcal{B}}(s,\mu) &= W^{\mathsf{H}}\mathcal{B}(s,\mu), & \widehat{\mathcal{N}}_j(s,\mu) &= W^{\mathsf{H}}\mathcal{N}_j(s,\mu)V,\end{aligned} \tag{5.28}$$

with $j = 1,\ldots,m$ and constant truncation matrices $V, W \in \mathbb{C}^{n\times r}$. The reduced-order system is then described by

$$\begin{aligned}\widehat{\mathcal{G}}_{\mathrm{B},k}(s_1,\ldots,s_k,\mu) = \widehat{\mathcal{C}}(s_k,\mu)\widehat{\mathcal{K}}(s_k,\mu)^{-1} \Bigg(\prod_{j=1}^{k-1} \Big(I_{m^{j-1}} \otimes \widehat{\mathcal{N}}(s_{k-j},\mu)\Big) \\ \times \Big(I_{m^j} \otimes \widehat{\mathcal{K}}(s_{k-j},\mu)^{-1}\Big)\Bigg) \Big(I_{m^{k-1}} \otimes \widehat{\mathcal{B}}(s_1,\mu)\Big).\end{aligned} \tag{5.29}$$

Additionally to the internal system structure, also the exact parameter dependencies are preserved in the reduced-order system when using the projection framework (5.28). In general, the frequency-affine decomposition (3.22) can be directly extended to a parameter-and-frequency-affine decomposition such that

$$\mathcal{K}(s,\mu) = \sum_{j=1}^{n_\mathcal{K}} h_{\mathcal{K},j}(s,\mu)\mathcal{K}_j$$

holds, with frequency- and parameter-dependent scalar functions $h_{\mathcal{K},j}$ and a possibly different $n_\mathcal{K}$ than in (3.22). This leads to the same observation as in the non-parametric case that the reduction only affects the constant matrices $\mathcal{K}_j$ and, therefore, the original system structure and now also the parameter dependencies described by the scalar functions $h_{\mathcal{K},j}$ are preserved.

Remark 5.12 (Parametric bilinear SISO systems):
As used in Section 5.3, the case of SISO subsystem transfer functions simplifies significantly due to the vanishing of Kronecker products and only a single bilinear term present:

$$\mathcal{G}_{\mathrm{B},k}(s_1,\ldots,s_k,\mu) = \mathcal{C}(s_k,\mu)\mathcal{K}(s_k,\mu)^{-1}\left(\prod_{j=1}^{k-1}\mathcal{N}(s_{k-j},\mu)\mathcal{K}(s_{k-j},\mu)^{-1}\right)\mathcal{B}(s_1,\mu). \quad (5.30)$$

This also inherits the simplification of the conditions on projection spaces in the structured interpolation theory. Due to the similarity to Section 5.3 and the recovering of SISO results from matrix interpolation, those simplified results for parametric bilinear SISO systems are omitted here. ◊

5.5.2 Structured interpolation in frequency and parameter

In the setting of parametric structured subsystem transfer functions $\mathcal{G}_{\mathrm{B},k}$ in (5.27), the goal is to construct V and W such that the reduced transfer functions $\widehat{\mathcal{G}}_{\mathrm{B},k}$ in (5.29) satisfy

$$\mathcal{G}_{\mathrm{B},k}(\sigma_1,\ldots,\sigma_k,\hat{\mu}) = \widehat{\mathcal{G}}_{\mathrm{B},k}(\sigma_1,\ldots,\sigma_k,\hat{\mu}) \quad \text{and} \quad (5.31)$$

$$\nabla\mathcal{G}_{\mathrm{B},k}(\sigma_1,\ldots,\sigma_k,\hat{\mu}) = \nabla\widehat{\mathcal{G}}_{\mathrm{B},k}(\sigma_1,\ldots,\sigma_k,\hat{\mu}), \quad (5.32)$$

for given frequency interpolation points $\sigma_1, \ldots, \sigma_k \in \mathbb{C}$ and the parameter interpolation point $\hat{\mu} \in \mathbb{M}$. In (5.31), $\nabla\mathcal{G}_{\mathrm{B},k}$ denotes the complete Jacobi matrix with

$$\nabla\mathcal{G}_{\mathrm{B},k} = \begin{bmatrix}\partial_{s_1}\mathcal{G}_{\mathrm{B},k} & \ldots & \partial_{s_k}\mathcal{G}_{\mathrm{B},k} & \partial_{\mu_1}\mathcal{G}_{\mathrm{B},k} & \ldots & \partial_{\mu_d}\mathcal{G}_{\mathrm{B},k}\end{bmatrix},$$

involving not only the partial derivatives with respect to the frequency arguments as in the non-parametric case but also the parameter sensitivities. Having in mind that in the general MIMO system case the transfer functions are matrix-valued, the conditions in (5.31) and (5.32) enforce matrix interpolation. The following theorem extends the results of Theorem 5.9 to the parametric case.

Theorem 5.13 (Parametric bilinear matrix interpolation):
Let $\mathcal{G}_{\mathrm{B}}$ be a parametric bilinear system, with its structured subsystem transfer functions $\mathcal{G}_{\mathrm{B},k}$ in (5.27), and $\widehat{\mathcal{G}}_{\mathrm{B}}$ be the reduced-order parametric bilinear system, constructed by (5.28) with its subsystem transfer functions $\widehat{\mathcal{G}}_{\mathrm{B},k}$ in (5.29). Given sets of frequency interpolation points $\sigma_1, \ldots, \sigma_k \in \mathbb{C}$ and $\varsigma_1, \ldots, \varsigma_\theta \in \mathbb{C}$, and the parameter interpolation point $\hat{\mu} \in \mathbb{M}$ for which the matrix functions $\mathcal{C}$, $\mathcal{K}^{-1}$, $\mathcal{N}$, $\mathcal{B}$ and $\widehat{\mathcal{K}}^{-1}$ are defined, the following statements hold:

(a) If V is constructed as

$$\begin{aligned} V_1 &= \mathcal{K}(\sigma_1, \hat{\mu})^{-1}\mathcal{B}(\sigma_1, \hat{\mu}), \\ V_j &= \mathcal{K}(\sigma_j, \hat{\mu})^{-1}\mathcal{N}(\sigma_{j-1}, \hat{\mu})(I_m \otimes V_{j-1}), \qquad 2 \le j \le k, \\ \operatorname{span}(V) &\supseteq \operatorname{span}\left(\begin{bmatrix} V_1 & \ldots & V_k \end{bmatrix}\right), \end{aligned}$$

then the following interpolation conditions hold true:

$$\mathcal{G}_{\mathrm{B},j}(\sigma_1, \ldots, \sigma_j, \hat{\mu}) = \widehat{\mathcal{G}}_{\mathrm{B},j}(\sigma_1, \ldots, \sigma_j, \hat{\mu}), \tag{5.33}$$

for $j = 1, \ldots, k$.

(b) If W is constructed as

$$\begin{aligned} W_1 &= \mathcal{K}(\varsigma_\theta, \hat{\mu})^{-\mathsf{H}}\mathcal{C}(\varsigma_\theta, \hat{\mu})^{\mathsf{H}}, \\ W_i &= \mathcal{K}(\varsigma_{\theta-i+1}, \hat{\mu})^{-\mathsf{H}}\overline{\boldsymbol{\mathcal{N}}(\varsigma_{\theta-i+1}, \hat{\mu})^{(2)}}(I_m \otimes W_{i-1}), \qquad 2 \le i \le \theta \\ \operatorname{span}(W) &\supseteq \operatorname{span}\left(\begin{bmatrix} W_1 & \ldots & W_\theta \end{bmatrix}\right), \end{aligned}$$

where $\boldsymbol{\mathcal{N}}^{(2)}$ is the 2-mode matricization of the tensor defined by $\boldsymbol{\mathcal{N}}^{(1)} = \mathcal{N}$, then the following interpolation conditions hold true:

$$\mathcal{G}_{\mathrm{B},i}(\varsigma_{\theta-i+1}, \ldots, \varsigma_\theta, \hat{\mu}) = \widehat{\mathcal{G}}_{\mathrm{B},i}(\varsigma_{\theta-i+1}, \ldots, \varsigma_\theta, \hat{\mu}), \tag{5.34}$$

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

(c) Let V be constructed as in Part (a) and W as in Part (b). Then, in addition to (5.33) and (5.34), the interpolation conditions

$$\begin{aligned} &\mathcal{G}_{\mathrm{B},q+\eta}(\sigma_1, \ldots, \sigma_q, \varsigma_{\theta-\eta+1}, \ldots, \varsigma_\theta, \hat{\mu}) \\ &= \widehat{\mathcal{G}}_{\mathrm{B},q+\eta}(\sigma_1, \ldots, \sigma_q, \varsigma_{\theta-\eta+1}, \ldots, \varsigma_\theta, \hat{\mu}) \end{aligned} \tag{5.35}$$

hold, for $1 \le q \le k$ and $1 \le \eta \le \theta$. ◊

Proof. Given the fixed parameter point $\hat{\mu} \in \mathbb{M}$, the matrix functions $\mathcal{C}(s, \hat{\mu})$, $\mathcal{K}(s, \hat{\mu})$, $\mathcal{N}(s, \hat{\mu})$ and $\mathcal{B}(s, \hat{\mu})$ can be viewed as realization of a non-parametric bilinear system. Then, the interpolation conditions (5.33)–(5.35) can be considered as subsystem interpolation of a non-parametric bilinear system as these conditions do not involve any variation/sensitivity with respect to μ. Therefore, the subspace conditions in Theorem 5.9, for interpolating a non-parametric structured bilinear system, apply here as well, which are precisely the subspace conditions listed in Parts (a)–(c). □

In Theorem 5.13, only function values are matched, i.e., the zeroth-order derivative. The following theorem extends these results to matching higher-order derivatives in the frequency arguments, i.e., to enforce Hermite interpolation conditions.

Theorem 5.14 (Parametric bilinear Hermite matrix interpolation):
Let $\mathcal{G}_{\mathrm{B}}$ be a parametric bilinear system, with its structured subsystem transfer functions $\mathcal{G}_{\mathrm{B},k}$ in (5.27), and $\widehat{\mathcal{G}}_{\mathrm{B}}$ be the reduced-order parametric bilinear system, constructed by (5.28) with its subsystem transfer functions $\widehat{\mathcal{G}}_{\mathrm{B},k}$ in (5.29). Given sets of frequency interpolation points $\sigma_1, \ldots, \sigma_k \in \mathbb{C}$ and $\varsigma_1, \ldots, \varsigma_\theta \in \mathbb{C}$, and the parameter interpolation point $\hat{\mu} \in \mathbb{M}$ for which the matrix functions $\mathcal{C}$, $\mathcal{K}^{-1}$, $\mathcal{N}$, $\mathcal{B}$ and $\widehat{\mathcal{K}}^{-1}$ are complex differentiable, and given the orders of partial derivatives $\ell_1, \ldots, \ell_k \in \mathbb{N}_0$ and $\nu_1, \ldots, \nu_\theta \in \mathbb{N}_0$, the following statements hold:

(a) If V is constructed as

$$\begin{aligned} V_{1,j_1} &= \partial_{s^{j_1}}(\mathcal{K}^{-1}\mathcal{B})(\sigma_1, \hat{\mu}), \\ V_{q,j_q} &= \partial_{s^{j_q}}\mathcal{K}^{-1}(\sigma_q, \hat{\mu}) \\ &\quad \times \left(\prod_{j=1}^{q-2} \partial_{s^{\ell_{q-j}}}\Big((I_{m^{j-1}} \otimes \mathcal{N})(I_{m^j} \otimes \mathcal{K})\Big)(\sigma_{q-j}, \hat{\mu})\right) \\ &\quad \times \partial_{s^{\ell_1}}\Big((I_{m^{q-2}} \otimes \mathcal{N})(I_{m^{q-1}} \otimes \mathcal{K})(I_{m^{q-1}} \otimes \mathcal{B})\Big)(\sigma_1, \hat{\mu}), \\ \operatorname{span}(V) &\supseteq \operatorname{span}\left(\begin{bmatrix} V_{1,0} & \ldots & V_{k,\ell_k} \end{bmatrix}\right), \end{aligned}$$

for $0 \leq j_1 \leq \ell_1$ and $2 \leq q \leq k$; $0 \leq j_q \leq \ell_q$, then the following interpolation conditions hold true:

$$\partial_{s_1^{\ell_1} \ldots s_{q-1}^{\ell_{q-1}} s_q^{j_q}} \mathcal{G}_{\mathrm{B},q}(\sigma_1, \ldots, \sigma_q, \hat{\mu}) = \partial_{s_1^{\ell_1} \ldots s_{q-1}^{\ell_{q-1}} s_q^{j_q}} \widehat{\mathcal{G}}_{\mathrm{B},q}(\sigma_1, \ldots, \sigma_q, \hat{\mu}), \tag{5.36}$$

for $q = 1, \ldots, k$ and $j_q = 0, \ldots, \ell_q$.

(b) If W is constructed as

$$\begin{aligned} W_{1,i_\theta} &= \partial_{s^{i_\theta}}(\mathcal{K}^{-\mathsf{H}}\mathcal{C}^{\mathsf{H}})(\varsigma_\theta, \hat{\mu}), \\ W_{\eta,i_{\theta-\eta+1}} &= \partial_{s^{i_{\theta-\eta+1}}}(\mathcal{K}^{-\mathsf{H}}\overline{\boldsymbol{\mathcal{N}}^{(2)}})(\varsigma_{\theta-\eta+1}, \hat{\mu})(I_m \otimes W_{\eta-1,\nu_{\theta-\eta+2}}), \\ \operatorname{span}(W) &\supseteq \operatorname{span}\left(\begin{bmatrix} W_{1,0} & \ldots & W_{\theta,\nu_\theta} \end{bmatrix}\right), \end{aligned}$$

for $2 \leq \eta \leq \theta$ and $0 \leq i_\theta \leq \nu_\theta$; $0 \leq i_{\theta-\eta+1} \leq \nu_{\theta-\eta+1}$, and where $\boldsymbol{\mathcal{N}}^{(2)}$ is the 2-mode matricization of the tensor defined by $\boldsymbol{\mathcal{N}}^{(1)} = \mathcal{N}$, then the following interpolation conditions hold true:

$$\begin{aligned} &\partial_{s_1^{i_{\theta-\eta+1}} s_2^{\nu_{\theta-\eta+2}} \cdots s_\theta^{\nu_\theta}} \mathcal{G}_{\mathrm{B},\eta}(\varsigma_{\theta-\eta+1}, \ldots, \varsigma_\theta, \hat{\mu}) \\ &= \partial_{s_1^{i_{\theta-\eta+1}} s_2^{\nu_{\theta-\eta+2}} \cdots s_\theta^{\nu_\theta}} \widehat{\mathcal{G}}_{\mathrm{B},\eta}(\varsigma_{\theta-\eta+1}, \ldots, \varsigma_\theta, \hat{\mu}), \end{aligned} \tag{5.37}$$

for $\eta = 1, \ldots, \theta$ and $i_{\theta-\eta+1} = 0, \ldots, \nu_{\theta-\eta+1}$.

(c) Let V be constructed as in Part (a) and W as in Part (b). Then, in addition to (5.36) and (5.37), the interpolation conditions

$$\begin{aligned} &\partial_{s_1^{\ell_1} \cdots s_{q-1}^{\ell_{q-1}} s_q^{j_q} s_{q+1}^{i_{\theta-\eta+1}} s_{q+2}^{\nu_{\theta-\eta+2}} \cdots s_{q+\eta}^{\nu_\theta}} \mathcal{G}_{\mathrm{B},q+\eta}(\sigma_1, \ldots, \sigma_q, \varsigma_{\theta-\eta+1}, \ldots, \varsigma_\theta, \hat{\mu}) \\ &= \partial_{s_1^{\ell_1} \cdots s_{q-1}^{\ell_{q-1}} s_q^{j_q} s_{q+1}^{i_{\theta-\eta+1}} s_{q+2}^{\nu_{\theta-\eta+2}} \cdots s_{q+\eta}^{\nu_\theta}} \widehat{\mathcal{G}}_{\mathrm{B},q+\eta}(\sigma_1, \ldots, \sigma_q, \varsigma_{\theta-\eta+1}, \ldots, \varsigma_\theta, \hat{\mu}) \end{aligned} \tag{5.38}$$

hold, for $j_q = 0, \ldots, \ell_q$; $i_{\theta-\eta+1} = 0, \ldots, \nu_{\theta-\eta+1}$; $1 \leq q \leq k$ and $1 \leq \eta \leq \theta$. ◊

Proof. As in Theorem 5.13, all the interpolation conditions are for a fixed parameter $\hat{\mu} \in \mathbb{M}$. Therefore, the subspace conditions from Theorem 5.10 can be applied here, which are precisely the subspace conditions listed in Theorem 5.14. □

5.5.3 Matching parameter sensitivities

So far, the interpolation conditions enforced did not show variability with respect to the parameter μ. Even in the Hermite conditions in Theorem 5.14, the matched derivatives (sensitivities) are only with respect to the frequency points. This enabled to directly employ the conditions and analysis from Section 5.4. However, for parametric systems it is important to match the sensitivity with respect to the parameter variation as well. This is what will be established in the next result, extending similar results from linear dynamics [21] and unstructured bilinear dynamics [160] to the new parametric structured framework. An important conclusion is that the parameter sensitivity is matched implicitly, i.e., without any explicit computation of it. This is achieved via the two-sided projection approach using the same set of frequency interpolation points (and orders of partial derivatives) for V and W.

Theorem 5.15 (Implicit parametric bilinear matrix interpolation):
Let $\mathcal{G}_\mathrm{B}$ be a parametric bilinear system, with its structured subsystem transfer functions $\mathcal{G}_{\mathrm{B},k}$ in (5.27), and $\widehat{\mathcal{G}}_\mathrm{B}$ be the reduced-order parametric bilinear system, constructed by (5.28) with its subsystem transfer functions $\widehat{\mathcal{G}}_{\mathrm{B},k}$ in (5.29). Given a set of frequency interpolation points $\sigma_1, \ldots, \sigma_k \in \mathbb{C}$ and the parameter interpolation point $\hat{\mu} \in \mathbb{M}$ for which the matrix functions $\mathcal{C}$, $\mathcal{K}^{-1}$, $\mathcal{N}$, $\mathcal{B}$ and $\widehat{\mathcal{K}}^{-1}$ are complex differentiable, the following statements hold:

(a) Let V be constructed as in Theorem 5.13 Part (a) and W as in Theorem 5.13 Part (b) with $\varsigma_i = \sigma_i$ for $i = 1, 2, \ldots, k$. Then, in addition to (5.33)–(5.35), it holds

$$\nabla \mathcal{G}_{\mathrm{B},k}(\sigma_1, \ldots, \sigma_k, \hat{\mu}) = \nabla \widehat{\mathcal{G}}_{\mathrm{B},k}(\sigma_1, \ldots, \sigma_k, \hat{\mu}). \tag{5.39}$$

(b) Let V be constructed as in Theorem 5.14 Part (a) and W as in Theorem 5.14 Part (b) with $\varsigma_i = \sigma_i$ and $\ell_i = \nu_i$ for $i = 1, 2, \ldots, k$. Then, in addition to (5.36)–(5.38), it holds

$$\nabla\left(\partial_{s_1^{\ell_1} \cdots s_k^{\ell_k}} \mathcal{G}_{\mathrm{B},k}\right)(\sigma_1, \ldots, \sigma_k, \hat{\mu}) = \nabla\left(\partial_{s_1^{\ell_1} \cdots s_k^{\ell_k}} \widehat{\mathcal{G}}_{\mathrm{B},k}\right)(\sigma_1, \ldots, \sigma_k, \hat{\mu}). \tag{5.40}$$

◊

Proof. For brevity, only (5.39) will be proven. The proof of (5.40) follows analogously using the correct subspaces and projectors. As in the proof of, e.g., Theorem 5.9, appropriate projectors onto the projection spaces $\operatorname{span}(V)$ and $\operatorname{span}(W)$ need to be constructed. In contrast to Theorem 5.14, now also the partial derivatives with respect to the parameters are interpolated. Using the product rule, the partial derivative of $\widehat{\mathcal{G}}_{\mathrm{B},k}$ with respect to a single parameter entry μ_i, for $1 \leq i \leq d$, is given by

$$\begin{aligned}
&\partial_{\mu_i} \widehat{\mathcal{G}}_{\mathrm{B},k}(\sigma_1, \ldots, \sigma_k, \hat{\mu}) \\
&= \sum_{\alpha \in \mathbb{A}} \left(\partial_{\mu_i^{\alpha_1}} \widehat{\mathcal{C}}(\sigma_k, \hat{\mu})\right) \left(\partial_{\mu_i^{\alpha_2}} \widehat{\mathcal{K}}^{-1}(\sigma_k, \hat{\mu})\right) \\
&\quad \times \left(\prod_{j=1}^{k-1} (I_{m^{j-1}} \otimes \partial_{\mu_i^{\alpha_{2j+1}}} \widehat{\mathcal{N}}(\sigma_{k-j}, \hat{\mu}))(I_{m^j} \otimes \partial_{\mu_i^{\alpha_{2j+2}}} \widehat{\mathcal{K}}^{-1}(\sigma_{k-j}, \hat{\mu}))\right) \\
&\quad \times (I_{m^{k-1}} \otimes \partial_{\mu_i^{\alpha_{2k+1}}} \widehat{\mathcal{B}}(\sigma_1, \hat{\mu})),
\end{aligned} \tag{5.41}$$

where $\mathbb{A}$ denotes the set of all columns of I_{2k+1}, the identity matrix of size $2k+1$. In other words, the right-hand side of (5.41) is a sum of $2k+1$ terms, where each term corresponds to the vector α taking a value from this set of columns. Therefore, in each term only a single matrix function is differentiated. It will be shown that every single term in the sum (5.41) matches the same term in the full-order model, thus, summed together, proving the desired interpolation property (5.39). Consider, e.g., the second term in (5.41), i.e., the term in which α is the second column of the identity matrix:

$$\alpha = \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \ldots & \alpha_{2k+1} \end{bmatrix}^{\mathsf{T}} = \begin{bmatrix} 0 & 1 & 0 & \ldots & 0 \end{bmatrix}^{\mathsf{T}}$$

Denote the corresponding term in the sum (5.41) by $\widehat{\mathcal{A}}_2$ such that

$$\widehat{\mathcal{A}}_2 := \widehat{\mathcal{C}}(\sigma_k, \hat{\mu})\left(\partial_{\mu_i} \widehat{\mathcal{K}}^{-1}(\sigma_k, \hat{\mu})\right) \left(\prod_{j=1}^{k-1} (I_{m^{j-1}} \otimes \widehat{\mathcal{N}}(\sigma_{k-j}, \hat{\mu}))(I_{m^j} \otimes \widehat{\mathcal{K}}(\sigma_{k-j}, \hat{\mu})^{-1})\right)$$

$$\times (I_{m^{k-1}} \otimes \widehat{\mathcal{B}}(\sigma_1, \hat{\mu})).$$

The derivative of the inverse appearing in $\widehat{\mathcal{A}}_2$ is given by

$$\partial_{\mu_i} \widehat{\mathcal{K}}^{-1}(\sigma_k, \hat{\mu}) = -\widehat{\mathcal{K}}(\sigma_k, \hat{\mu})^{-1} \Big(\partial_{\mu_i} \widehat{\mathcal{K}}(\sigma_k, \hat{\mu}) \Big) \widehat{\mathcal{K}}(\sigma_k, \hat{\mu})^{-1}.$$

Therefore, $\widehat{\mathcal{A}}_2$ can be rewritten as

$$\begin{aligned}
\widehat{\mathcal{A}}_2 &= -\widehat{\mathcal{C}}(\sigma_k, \hat{\mu}) \widehat{\mathcal{K}}(\sigma_k, \hat{\mu})^{-1} \Big(\partial_{\mu_i} \widehat{\mathcal{K}}(\sigma_k, \hat{\mu}) \Big) \widehat{\mathcal{K}}(\sigma_k, \hat{\mu})^{-1} \\
&\quad \times \left(\prod_{j=1}^{k-1} \Big(I_{m^{j-1}} \otimes \widehat{\mathcal{N}}(\sigma_{k-j}, \hat{\mu}) \Big) \Big(I_{m^j} \otimes \widehat{\mathcal{K}}(\sigma_{k-j}, \hat{\mu})^{-1} \Big) \right) \\
&\quad \times (I_{m^{k-1}} \otimes \widehat{\mathcal{B}}(\sigma_1, \hat{\mu})) \\
&=: -\widehat{W}_1^{\mathsf{H}} \Big(\partial_{\mu_i} \widehat{\mathcal{K}}(\sigma_k, \hat{\mu}) \Big) \widehat{V}_k.
\end{aligned}$$

Noting that the projection space $\operatorname{span}(V)$ was constructed as in Theorem 5.13, it holds

$$\begin{aligned}
V\widehat{V}_k &= V \widehat{\mathcal{K}}(\sigma_k, \hat{\mu})^{-1} \left(\prod_{j=1}^{k-1} (I_{m^{j-1}} \otimes \widehat{\mathcal{N}}(\sigma_{k-j}, \hat{\mu}))(I_{m^j} \otimes \widehat{\mathcal{K}}(\sigma_{k-j}, \hat{\mu})^{-1}) \right) \\
&\quad \times (I_{m^{k-1}} \otimes \widehat{\mathcal{B}}(\sigma_1, \hat{\mu})) \\
&= \underbrace{V \widehat{\mathcal{K}}(\sigma_k, \hat{\mu})^{-1} W^{\mathsf{H}} \mathcal{K}(\sigma_k, \hat{\mu})}_{= P_{\mathrm{V}}(\sigma_k)} \mathcal{K}(\sigma_k, \hat{\mu})^{-1} \left(\prod_{j=1}^{k-1} (I_{m^{j-1}} \otimes \mathcal{N}(\sigma_{k-j}, \hat{\mu})) \right. \\
&\quad \left. \times (I_{m^j} \otimes \mathcal{K}(\sigma_{k-j}, \hat{\mu})^{-1}) \right) (I_{m^{k-1}} \otimes \mathcal{B}(\sigma_1, \hat{\mu})) \\
&= P_{\mathrm{V}}(\sigma_k) V_k \\
&= V_k,
\end{aligned}$$

where $P_{\mathrm{V}}(\sigma_k)$ is the projector onto $\operatorname{span}(V)$ from (3.24). The other necessary projectors $P_{\mathrm{V}}(\sigma_1), \ldots, P_{\mathrm{V}}(\sigma_{k-1})$ were directly applied in the second step without further mentioning. Similarly, it holds

$$\begin{aligned}
W\widehat{W}_1 &= W \widehat{\mathcal{K}}(\sigma_k, \hat{\mu})^{-\mathsf{H}} \widehat{\mathcal{C}}(\sigma_k, \hat{\mu})^{\mathsf{H}} \\
&= \underbrace{W \widehat{\mathcal{K}}(\sigma_k, \hat{\mu})^{-\mathsf{H}} V \mathcal{K}(\sigma_k, \hat{\mu})^{\mathsf{H}}}_{= P_{\mathrm{W}}(\sigma_k)} \underbrace{\mathcal{K}(\sigma_k, \hat{\mu})^{-\mathsf{H}} \mathcal{C}(\sigma_k, \hat{\mu})^{\mathsf{H}}}_{= W_1} \\
&= W_1,
\end{aligned}$$

with $P_{\mathrm{W}}(\sigma_k)$ the projector onto span(W) from (3.25). Using these two identities, one obtains

$$\begin{aligned}
\widehat{\mathcal{A}}_2 &= -\widehat{W}_1^{\mathsf{H}}\Big(\partial_{\mu_i}\widehat{\mathcal{K}}(\sigma_k,\hat{\mu})\Big)\widehat{V}_k \\
&= -\widehat{W}_1^{\mathsf{H}}W^{\mathsf{H}}\Big(\partial_{\mu_i}\mathcal{K}(\sigma_k,\hat{\mu})\Big)V\widehat{V}_k \\
&= -W_1^{\mathsf{H}}\Big(\partial_{\mu_i}\mathcal{K}(\sigma_k,\hat{\mu})\Big)V_k \\
&= \mathcal{C}(\sigma_k,\hat{\mu})\left(\partial_{\mu_i}\mathcal{K}^{-1}(\sigma_k,\hat{\mu})\right)\left(\prod_{j=1}^{k-1}(I_{m^{j-1}}\otimes\mathcal{N}(\sigma_{k-j},\hat{\mu}))(I_{m^j}\otimes\mathcal{K}(\sigma_{k-j},\hat{\mu})^{-1})\right) \\
&\quad\times(I_{m^{k-1}}\otimes\mathcal{B}(\sigma_1,\hat{\mu})),
\end{aligned}$$

i.e., $\widehat{\mathcal{A}}_2$ is identical to the same term using the original matrix functions. Since the same technique can be used for all other possible α vectors, it holds

$$\partial_{\mu_i}\widehat{\mathcal{G}}_{\mathrm{B},k}(\sigma_1,\ldots,\sigma_k,\hat{\mu}) = \partial_{\mu_i}\mathcal{G}_{\mathrm{B},k}(\sigma_1,\ldots,\sigma_k,\hat{\mu}), \tag{5.42}$$

for all $1 \leq i \leq d$. Interpolation of the partial derivatives with respect to the frequency parameters follows by using Corollary 5.11 with the fixed parameter $\hat{\mu}$. Together with (5.42), this proves (5.39). □

5.5.4 Numerical experiments

The results for structured interpolation of parametric bilinear systems have a strong similarity to the non-parametric results in Sections 5.3 and 5.4. Therefore and for brevity, only two short proof-of-concept experiments are performed for parametric versions of the models in Section 5.3.3. Only the equidistant interpolation point selection and the averaged subspace approach with a light oversampling in frequency and parameter arguments will be compared. Instead of full comparisons via MORscores, a fixed order is directly assumed for the model reduction and the results are then compared in parametric extensions of the pointwise relative errors (4.19), (4.20), and (5.17), namely

$$\epsilon_{\mathrm{rel}}(\omega_1,\mu) = \frac{\|G_{\mathrm{L}}(\omega\mathbf{i},\mu)-\widehat{G}_{\mathrm{L}}(\omega\mathbf{i},\mu)\|_2}{\|G_{\mathrm{L}}(\omega\mathbf{i},\mu)\|_2} \qquad \text{and}$$

$$\epsilon_{\mathrm{rel}}(\omega_1,\omega_2,\mu) = \frac{\|\mathcal{G}_{\mathrm{B},2}(\omega_1\mathbf{i},\omega_2\mathbf{i},\mu)-\widehat{\mathcal{G}}_{\mathrm{B},2}(\omega_1\mathbf{i},\omega_2\mathbf{i},\mu)\|_2}{\|\mathcal{G}_{\mathrm{B},2}(\omega_1\mathbf{i},\omega_2\mathbf{i},\mu)\|_2}$$

in frequency domain, and

$$\epsilon_{\mathrm{rel}}(t,\mu) = \frac{\|y(t;\mu)-\hat{y}(t;\mu)\|_2}{\|y(t;\mu)\|_2},$$

for the time simulation error.

Table 5.3: Maximum pointwise relative errors for the parametric bilinear mass-spring-damper example and reduced models of order $r_2 = 40$.

	StrInt(equi.)	StrInt(avg.)
$\max_{\omega_1} \epsilon_{\text{rel}}(\omega_1)$	3.897330e-04	1.472924e-07
$\max_{\omega_1,\omega_2,\mu} \epsilon_{\text{rel}}(\omega_1,\omega_2,\mu)$	2.928265e-03	1.076899e-05
$\max_{t,\mu} \epsilon_{\text{rel}}(t,\mu)$	1.120365e-03	7.867531e-10

5.5.4.1 Parametric bilinear mass-spring-damper system

The first example is an extension of the bilinear mass-spring-damper system from Section 5.3.3.1. The input vector is split into two according to its range of affection and a second bilinear term is introduced acting into the opposite direction such that

$$B_{\text{u}} = \begin{bmatrix} \mathbb{1}_5 & 0 \\ 0 & \vdots \\ \vdots & 0 \\ 0 & \mathbb{1}_5 \end{bmatrix}, \quad N_{\text{p},1} = -S_1 K S_1, \quad N_{\text{p},2} = S_2 K S_2,$$

with S_1 a diagonal matrix with entries `linspace(0.5,0,`n_2`)` and S_2 a diagonal matrix with entries `linspace(0,0.5,`n_2`)`. Also, the output matrix is split into two rows by taking the observations of the first half of the chain into the first row and the rest into the second row. The number of masses $n_2 = 10\,000$ stays unchanged and two parameters $(\mu_1, \mu_2) = \mu \in \mathbb{M} = [0,1] \times [0,1]$ are used to control the strength of the bilinearities in the system. The resulting parametric mechanical bilinear MIMO system has the form

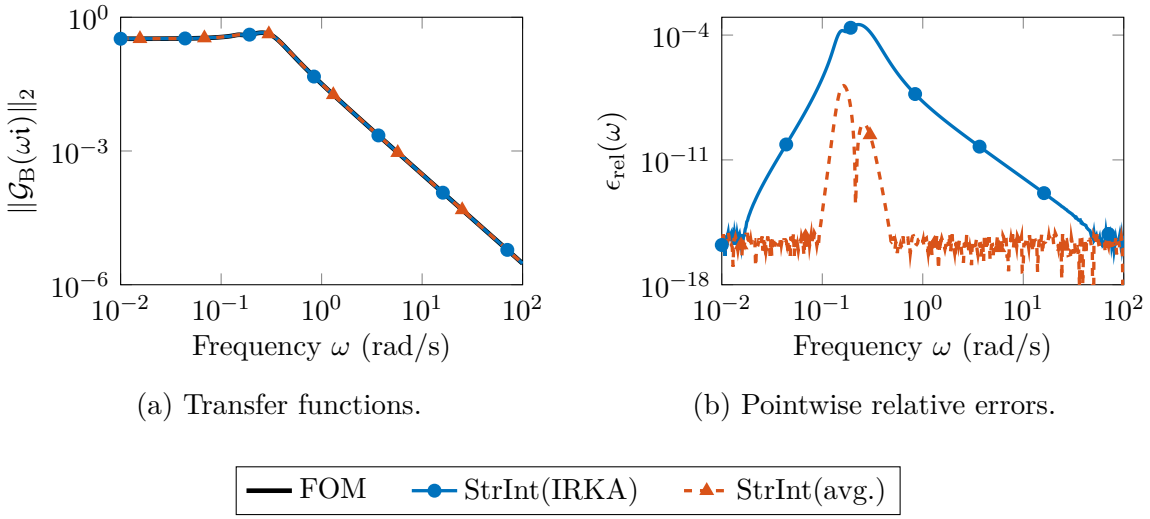
$$\begin{aligned} M\ddot{x}(t) + E\dot{x}(t) + Kx(t) &= \mu_1 N_{\text{p},1} x(t) u_1(t) + \mu_2 N_{\text{p},2} x(t) u_2(t) + B_{\text{u}} u(t), \\ y(t) &= C_{\text{p}} x(t). \end{aligned} \tag{5.43}$$

Note that for $\mu_1 = \mu_2 = 0$, the system (5.43) is linear. In the setting of structured subsystem transfer functions, the matrix-valued functions in (5.27) are realized by

$$\begin{aligned} \mathcal{C}(s,\mu) &= C_{\text{p}} + sC_{\text{v}}, & \mathcal{K}(s,\mu) &= s^2 M + sE + K, \\ \mathcal{B}(s,\mu) &= B_{\text{u}}, & \mathcal{N}(s,\mu) &= \begin{bmatrix} \mu_1 N_{\text{p},1} & \mu_2 N_{\text{p},2} \end{bmatrix}, \end{aligned}$$

such that only the matrix function representing the bilinear terms depends on the parameters.

To preserve definiteness of the system matrices, only a one-sided projection, $V = W$, is employed in the model reduction process. Theorem 5.13 Part (a) was thereby used



(a) Transfer functions.

(b) Pointwise relative errors.

Figure 5.7: First subsystem transfer functions and approximation errors for the parametric bilinear mass-spring-damper example.

in StrInt(equi.) to interpolate the first and second subsystem transfer functions in the frequency interpolation points $\pm\{10^{-2}, 10^2\}\mathrm{i}$ and in the parameter interpolation points $\{(0,1),(1,0)\}$. Observing that the first subsystem transfer function is independent of the parameters, the resulting reduced-order model is of order $r_2 = 40$. For StrInt(avg.), Theorem 5.15 Part (a) was used to construct interpolation bases for ten logarithmically equidistant points in frequency and four linearly equidistant points in both parameters for the first and second subsystem transfer functions. The resulting matrices were then concatenated and truncated by the pivoted QR decomposition into a single orthogonal basis to the desired reduced order of $r_2 = 40$.

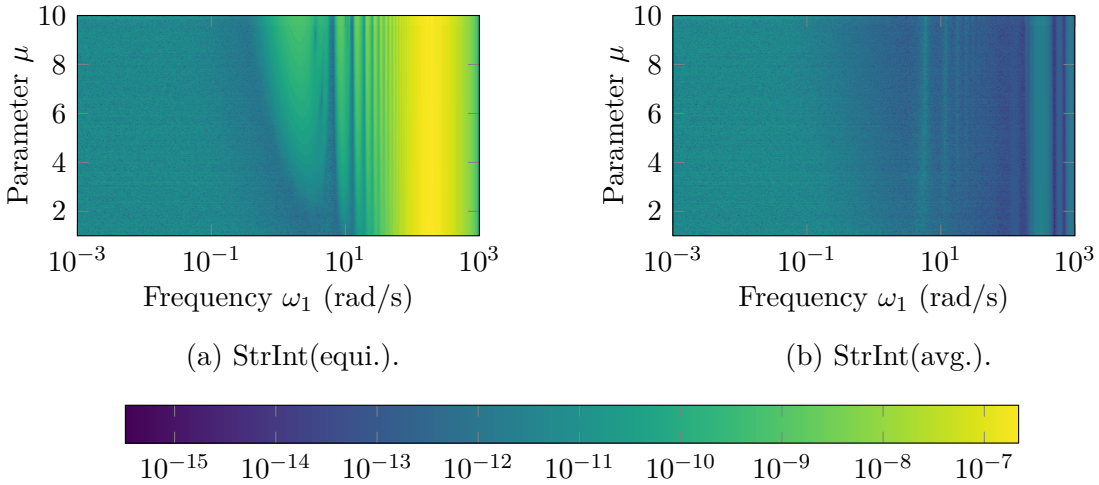
The pointwise relative errors were computed for both approximations for the first and second subsystem transfer functions in the frequency range $\omega_1, \omega_2 \in [10^{-2}, 10^2]$ rad/s, as well as for the time simulations in the interval $[0, 100]$ s. The following input signal was used in the simulations

$$u(t) = 10 \cdot \begin{bmatrix} \eta_1(t_j) \\ \eta_2(t_j) \end{bmatrix}, \quad \text{for } t_j \leq t < t_{j+1}, \tag{5.44}$$

with $j = 0, \ldots, 99$, equidistant time steps $t_j = j \cdot \frac{100}{99}$ and presampled Gaussian white noise $\eta_1(t), \eta_2(t)$. The maximum attained pointwise relative erros are shown in Table 5.3. Both methods perform very well, where the reduced-order model of choice would be StrInt(avg.). The parameter-independent first subsystem transfer functions are shown in Figure 5.7. One can see that both approaches deliver accurate reduced-order models, where the worst case deviations happen to be in the middle of the frequency range, which is far away from the chosen interpolation points in StrInt(equi.). This indicates that

Table 5.4: Maximum pointwise relative errors for the parametric time-delay example and reduced models of order $r_1 = 24$.

	StrInt(equi.)	StrInt(avg.)
$\max_{\omega_1} \epsilon_{\text{rel}}(\omega_1, \mu)$	2.183130e-07	4.370906e-11
$\max_{\omega_1,\omega_2,\mu} \epsilon_{\text{rel}}(\omega_1, \omega_2, \mu)$	9.491342e-07	7.536612e-11
$\max_{t,\mu} \epsilon_{\text{rel}}(t, \mu)$	2.370008e-05	7.469150e-06



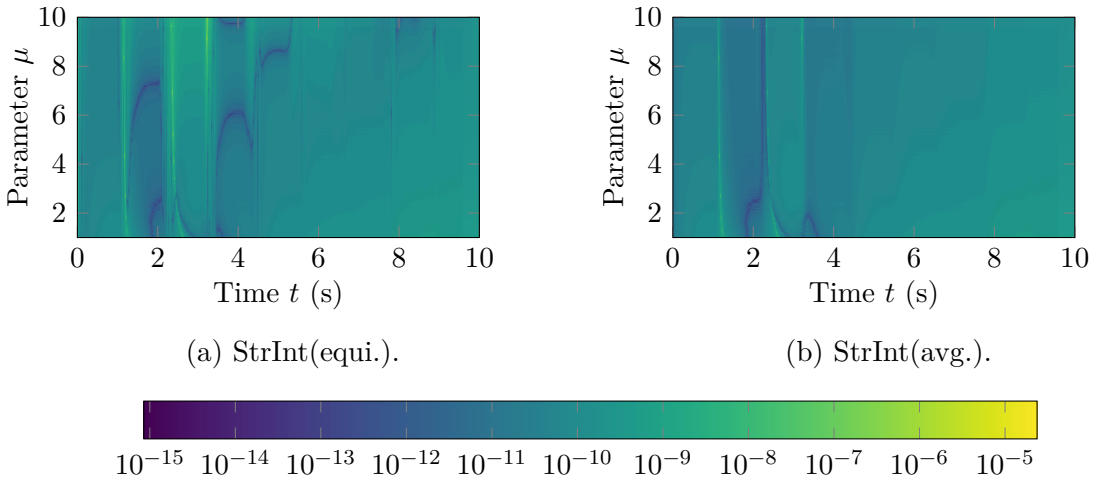
(a) StrInt(equi.). (b) StrInt(avg.).

Figure 5.8: Relative approximation errors $\epsilon_{\text{rel}}(\omega_1, \mu)$ of the first subsystem transfer functions for the parametric time-delay example.

another interpolation point at the maximum attained error might be very beneficial for StrInt(equi.), or alternatively, that a more sophisticated selection of interpolation points using, for example, the IRKA and $\mathcal{H}_\infty$-greedy methods, will provide better reduced-order models. StrInt(avg.) also shows an interesting error behavior in Figure 5.7. The visible sink in the pointwise relative error in the middle of the frequency range leads to the assumption that StrInt(avg.) is still interpolating at this point, i.e., that in the compression of the projection space bases this specific information was dominant enough to be preserved.

5.5.4.2 Parametric time-delayed heated rod

As second example, a parametric version of the time-delayed heated rod from Section 5.3.3.2 is used. Therefore, the diffusivity coefficients in (5.20) are parametrized with



(a) StrInt(equi.). (b) StrInt(avg.).

Figure 5.9: Relative approximation errors $\epsilon_{\text{rel}}(t, \mu)$ of the time simulations for the parametric time-delay example.

$\mu \in [1, 10]$. This results in a parametric bilinear time-delay SISO system of the form

$$\begin{aligned} \mathsf{E}\dot{\mathsf{x}}(t) &= (\mathsf{A}_0 - \mu\mathsf{A}_\mathsf{d})\mathsf{x}(t) + \mu\mathsf{A}_\mathsf{d}\mathsf{x}(t-\tau) + \mathsf{N}\mathsf{x}(t)u(t) + \mathsf{B}u(t), \\ y(t) &= \mathsf{C}\mathsf{x}(t), \end{aligned} \tag{5.45}$$

with the time delay $\tau = 1$ and $n_1 = 5\,000$ differential equations. Using the framework of parametric structured subsystem transfer functions (5.30), the frequency domain representation of (5.45) is given by the matrix-valued functions

$$\mathcal{C}(s, \mu) = \mathsf{C}, \quad \mathcal{K}(s, \mu) = s\mathsf{E} - (\mathsf{A}_0 - \mu\mathsf{A}_\mathsf{d}) - \mu e^{-s\tau}\mathsf{A}_\mathsf{d}, \quad \mathcal{B}(s, \mu) = \mathsf{B}, \quad \mathcal{N}(s, \mu) = \mathsf{N},$$

where only the center term representing the linear dynamics depends on the parameter.

For model order reduction, a two-sided projection is used, where both approaches StrInt(equi.) and StrInt(avg.) are based on Theorem 5.15 for the first and second subsystem transfer functions. For the equidistant interpolation point selection, in frequency domain the points $\pm\{10^{-3}, 10^{3}\}\mathrm{i}$ and in parameter domain $\{1, 5.5, 10\}$ are used. This results in the reduced order $r_1 = 24$. For the averaged subspace approach, both left and right interpolation spaces are set up for 40 logarithmically equidistant points in the frequency range $[10^{-3}, 10^{3}]$ rad/s and 10 linearly equidistant points in $[1, 10]$. Afterwards, the resulting bases are orthogonalized and truncated via the pivoted QR decomposition to order $r_1 = 24$.

The maxima of the pointwise relative errors are shown in Table 5.4, were in freqency domain the range $\omega_1, \omega_2 \in [10^{-3}, 10^{3}]$ rad/s was used, and in time domain the systems were simulated in the interval $[0, 10]$ s with the same input signal as in the non-parametric example (5.21). Both reduced-order models yield a suitable approximation quality. In the

frequency domain comparison, the averaged subspaces perform clearly better. The reason can be seen in Figure 5.8 as StrInt(equi.) has a stronger increase in the error behavior for larger frequencies. Another frequency interpolation point in this region might fix this issue. In time domain, the approximation errors of StrInt(equi.) and StrInt(avg.) are very close to each other. The relative time domain errors are shown in Figure 5.9. Some sinks and peaks of the approximation errors are visible for both methods but otherwise the errors are very uniform over time and parameter.

5.6 Tangential interpolation framework for structured bilinear systems

A general problem in matrix interpolation for MIMO systems is the fast growth of the underlying projection spaces and, consequently, of the resulting reduced-order models. This comes from matching interpolation conditions in each entry of the matrix-valued transfer functions. A remedy used in case of linear systems is the tangential interpolation approach (see Section 3.3.2) allowing for a fine control of the projection space dimensions. In case of unstructured bilinear systems (2.27), a first attempt of generalizing tangential interpolation to subsystem transfer functions was done in [31, 160]. This approach is referred to as *blockwise tangential interpolation* as it is based on considering the single block-matrix entries of the subsystem transfer functions (2.33) separately. For example, the right blockwise tangential interpolation problem for the k-th subsystem transfer function with interpolation points $\sigma_1, \ldots, \sigma_k \in \mathbb{C}$ and right tangential direction $b \in \mathbb{C}^m$ aims for the construction of a reduced-order model that interpolates

$$\begin{aligned} \mathsf{G}_{\mathrm{B},k}(\sigma_1, \ldots, \sigma_k)(I_m \otimes b) = \Big[& \mathsf{C}(\sigma_k \mathsf{E} - \mathsf{A})^{-1} \mathsf{N}_1 \cdots \mathsf{N}_1 (\sigma_1 \mathsf{E} - \mathsf{A})^{-1} \mathsf{B} b, \\ & \mathsf{C}(\sigma_k \mathsf{E} - \mathsf{A})^{-1} \mathsf{N}_1 \cdots \mathsf{N}_2 (\sigma_1 \mathsf{E} - \mathsf{A})^{-1} \mathsf{B} b, \\ & \ldots, \\ & \mathsf{C}(\sigma_k \mathsf{E} - \mathsf{A})^{-1} \mathsf{N}_m \cdots \mathsf{N}_m (\sigma_1 \mathsf{E} - \mathsf{A})^{-1} \mathsf{B} b \Big]. \end{aligned} \tag{5.46}$$

In this section, the idea of tangential interpolation (3.16) for model order reduction is extended to structured bilinear systems in a much broader sense than in [31, 160]. Therefore, re-interpretations of tangential interpolation in frequency and time domain are done in Sections 5.6.1 and 5.6.2, followed by a general framework for tangential interpolation of structured bilinear systems in Section 5.6.3. The special case of blockwise tangential interpolation from the literature is considered separately in Section 5.6.4 as a special instance of the new framework. The tangential approaches are then tested in different numerical examples.

5.6.1 Frequency domain interpretation of tangential interpolation

Looking back to the origins of tangential interpolation (3.15) and the multivariate transfer functions (5.3), a first idea is to consider an appropriately sized vector $\tilde{b} \in \mathbb{C}^{m^k}$ as right tangential direction, which results by multiplication with the subsystem transfer functions (5.3) in

$$\begin{aligned}\mathcal{G}_{\mathrm{B},k}(s_1,\ldots,s_k)\tilde{b} = \sum_{j_1=1}^{m} \ldots \sum_{j_{k-1}=1}^{m} & \mathcal{C}(s_k)\mathcal{K}(s_k)^{-1}\mathcal{N}_{j_{k-1}}(s_{k-1})\mathcal{K}(s_{k-1})^{-1} \\ & \times \ldots \times \mathcal{N}_{j_1}(s_1)\mathcal{K}(s_1)^{-1}\mathcal{B}(s_1)\tilde{b}^{(\alpha)},\end{aligned} \tag{5.47}$$

where α is an appropriately changing index according to the $k-1$ sums and the partition of the full direction vector

$$\tilde{b} = \begin{bmatrix} \left(\tilde{b}^{(1)}\right)^{\mathsf{H}} & \ldots & \left(\tilde{b}^{(m^{k-1})}\right)^{\mathsf{H}} \end{bmatrix}^{\mathsf{H}},$$

where $\tilde{b}^{(\alpha)} \in \mathbb{C}^m$ for $\alpha = 1,\ldots,m^{k-1}$. This general approach leads to a problem concerning the recursive structure of the transfer functions and the corresponding construction of the projection spaces. For every new level of the subsystem transfer functions, a different part of $\tilde{b}$ is multiplied with the input function $\mathcal{B}(s)$ in each term of the sum (5.47). Therefore, the corresponding projection bases for the tangential interpolation would grow vastly according to the number of different block entries in $\tilde{b}$, which is not suited to produce small reduced-order models. A solution to this problem is the restriction of the full direction vector to the repetition of a smaller one $b \in \mathbb{C}^m$ such that

$$\tilde{b} = \mathbb{1}_{m^{k-1}} \otimes b = \begin{bmatrix} b \\ \vdots \\ b \end{bmatrix}, \tag{5.48}$$

with $\mathbb{1}_{m^{k-1}}$ the vector of length m^{k-1} containing only ones. With the particular choice of (5.48), the right tangential interpolation problem can be written as

$$\mathcal{G}_{\mathrm{B},k}(\sigma_1,\ldots,\sigma_k)(\mathbb{1}_{m^{k-1}} \otimes b) = \widehat{\mathcal{G}}_{\mathrm{B},k}(\sigma_1,\ldots,\sigma_k)(\mathbb{1}_{m^{k-1}} \otimes b), \tag{5.49}$$

for a given set of interpolation points $\sigma_1,\ldots,\sigma_k \in \mathbb{C}$. In (5.49), the interpolation problem is restricted to vectors of constant length p, independent of the input dimension m. Therefore, it allows for an efficient construction of projection bases.

The left tangential interpolation problem in the classical approach (3.15) would lead to the same idea as in the blockwise tangential interpolation, since the output dimension p of the transfer function is constant over all subsystem levels. To consider a dual formulation of (5.47) and, corresponding to that, a projection basis that does not increase its dimension exponentially with the transfer function level, the natural choice is

$$c^{\mathsf{H}}\mathcal{G}_{\mathrm{B},k}(\sigma_1,\ldots,\sigma_k)(\mathbb{1}_{m^{k-1}} \otimes I_m) = c^{\mathsf{H}}\widehat{\mathcal{G}}_{\mathrm{B},k}(\sigma_1,\ldots,\sigma_k)(\mathbb{1}_{m^{k-1}} \otimes I_m), \tag{5.50}$$

for a given direction $c \in \mathbb{C}^p$ and interpolation points $\sigma_1, \ldots, \sigma_k \in \mathbb{C}$, as the left tangential interpolation problem and, consequently,

$$c^{\mathsf{H}}\mathcal{G}_{\mathrm{B},k}(\sigma_1, \ldots, \sigma_k)(\mathbb{1}_{m^{k-1}} \otimes b) = c^{\mathsf{H}}\widehat{\mathcal{G}}_{\mathrm{B},k}(\sigma_1, \ldots, \sigma_k)(\mathbb{1}_{m^{k-1}} \otimes b) \tag{5.51}$$

for two-sided tangential interpolation.

5.6.2 Time domain interpretation of tangential interpolation

A different way to look at tangential interpolation of transfer functions with underlying dynamical systems is (re-)interpretation in the time domain. Consider the tangential interpolation problem for linear dynamical systems (3.16). For simplicity, the following discussion is restricted to the simplified case of linear unstructured first-order systems (2.8) with transfer function (2.14). But note that the upcoming ideas work analogously in the general structured case [24]. The multiplication with tangential directions in the frequency domain can be considered independent of the chosen interpolation points. This yields new systems in the frequency domain described by transfer functions that are restricted in one or both dimensions:

$$\widetilde{\mathsf{G}}_{\mathrm{L,b}}(s) = \mathsf{G}_{\mathrm{L}}(s)b, \quad \widetilde{\mathsf{G}}_{\mathrm{L,c}}(s) = c^{\mathsf{H}}\mathsf{G}_{\mathrm{L}}(s) \quad \text{and} \quad \widetilde{\mathsf{G}}_{\mathrm{L,cb}}(s) = c^{\mathsf{H}}\mathsf{G}_{\mathrm{L}}(s)b, \tag{5.52}$$

with the tangential directions $b \in \mathbb{C}^m$ and $c \in \mathbb{C}^p$. These new restricted systems (5.52) can now be transformed back into time domain. The resulting tangential systems can be seen as embedding the original linear system G_{L} into single-input and/or single-output systems. Let the outer inputs and outputs be set to be $u(t) = b\tilde{u}(t)$ and $\tilde{y}(t) = c^{\mathsf{H}}y(t)$, respectively, the three embedded (restricted) systems are given by

$$\widetilde{\mathsf{G}}_{\mathrm{L,b}} : \begin{cases} \mathsf{E}\dot{\mathsf{x}}(t) = \mathsf{A}\mathsf{x}(t) + \mathsf{B}b\tilde{u}(t), \\ y(t) = \mathsf{C}\mathsf{x}(t), \end{cases} \tag{5.53}$$

for the right tangential interpolation problem,

$$\widetilde{\mathsf{G}}_{\mathrm{L,c}} : \begin{cases} \mathsf{E}\dot{\mathsf{x}}(t) = \mathsf{A}\mathsf{x}(t) + \mathsf{B}u(t), \\ \tilde{y}(t) = c^{\mathsf{H}}\mathsf{C}\mathsf{x}(t), \end{cases} \tag{5.54}$$

for the left tangential interpolation problem, and

$$\widetilde{\mathsf{G}}_{\mathrm{L,cb}} : \begin{cases} \mathsf{E}\dot{\mathsf{x}}(t) = \mathsf{A}\mathsf{x}(t) + \mathsf{B}b\tilde{u}(t), \\ \tilde{y}(t) = c^{\mathsf{H}}\mathsf{C}\mathsf{x}(t), \end{cases} \tag{5.55}$$

for the two-sided tangential interpolation. The systems (5.53)–(5.55) correspond to the three identically denoted transfer functions in (5.52). With (5.53)–(5.55) one can

interpret tangential interpolation as the restriction of the system inputs to a single input signal that is spread along a given direction b to be fed into the original system (2.8) and/or the restriction of the output to a linear combination of the observations of the original system (2.8) using the direction c.

Now, consider the case of bilinear unstructured systems (2.27). The time domain interpretation of tangential interpolation of the linear case (5.53)–(5.55) can be directly transferred to bilinear systems. Using the same tangential directions b and c as before, and the embedding strategy for the bilinear system (2.27), one gets

$$\widetilde{\mathsf{G}}_{\mathrm{B,b}} : \begin{cases} \mathsf{E}\dot{\mathsf{x}}(t) = \mathsf{A}\mathsf{x}(t) + \sum_{j=1}^{m} \mathsf{N}_j \mathsf{x}(t) b_j \tilde{u}(t) + \mathsf{B} b \tilde{u}(t), \\ y(t) = \mathsf{C}\mathsf{x}(t), \end{cases} \tag{5.56}$$

for the inputs,

$$\widetilde{\mathsf{G}}_{\mathrm{B,c}} : \begin{cases} \mathsf{E}\dot{\mathsf{x}}(t) = \mathsf{A}\mathsf{x}(t) + \sum_{j=1}^{m} \mathsf{N}_j \mathsf{x}(t) u_j(t) + \mathsf{B} u(t), \\ \tilde{y}(t) = c^{\mathsf{H}} \mathsf{C}\mathsf{x}(t), \end{cases} \tag{5.57}$$

for the outputs, and

$$\widetilde{\mathsf{G}}_{\mathrm{B,cb}} : \begin{cases} \mathsf{E}\dot{\mathsf{x}}(t) = \mathsf{A}\mathsf{x}(t) + \sum_{j=1}^{m} \mathsf{N}_j \mathsf{x}(t) b_j \tilde{u}(t) + \mathsf{B} b \tilde{u}(t), \\ \tilde{y}(t) = c^{\mathsf{H}} \mathsf{C}\mathsf{x}(t), \end{cases} \tag{5.58}$$

for the fully embedded system. These restricted bilinear systems (5.56)–(5.58) can be transformed into their frequency domain representations to get back to the tangential interpolation problem for subsystem transfer functions. The corresponding regular subsystem transfer functions for the restricted systems look like follows:

$$\widetilde{\mathsf{G}}_{\mathrm{B,b},k}(s_1, \ldots, s_k) = \mathsf{C}(s_k\mathsf{E} - \mathsf{A})^{-1} \left(\prod_{j=1}^{k-1} \left(\sum_{i=1}^{m} b_i \mathsf{N}_i \right) (s_{k-j}\mathsf{E} - \mathsf{A})^{-1} \right) \mathsf{B} b, \tag{5.59}$$

$$\begin{aligned} \widetilde{\mathsf{G}}_{\mathrm{B,c},k}(s_1, \ldots, s_k) = c^{\mathsf{H}} \mathsf{C}(s_k\mathsf{E} - \mathsf{A})^{-1} \left(\prod_{j=1}^{k-1} (I_{m^{j-1}} \otimes \mathsf{N})(I_{m^j} \otimes (s_{k-j}\mathsf{E} - \mathsf{A})^{-1}) \right) \\ \times (I_{m^{k-1}} \otimes \mathsf{B}), \end{aligned} \tag{5.60}$$

$$\widetilde{\mathsf{G}}_{\mathrm{B,cb},k}(s_1, \ldots, s_k) = c^{\mathsf{H}} \mathsf{C}(s_k\mathsf{E} - \mathsf{A})^{-1} \left(\prod_{j=1}^{k-1} \left(\sum_{i=1}^{m} b_i \mathsf{N}_i \right) (s_{k-j}\mathsf{E} - \mathsf{A})^{-1} \right) \mathsf{B} b, \tag{5.61}$$

for $k \geq 1$. These new transfer functions (5.59)–(5.61) can now be combined with the setting of structured subsystem transfer function (5.3). Denote the scaled summation of

the bilinear terms in the structured multivariate transfer functions by

$$\widetilde{\mathcal{G}}_{\mathrm{B},k}(s_1,\ldots,s_k) = \mathcal{C}(s_k)\mathcal{K}(s_k)^{-1}\left(\sum_{j=1}^{m} b_j\mathcal{N}_j(s_{k-1})\right)\mathcal{K}(s_{k-1})^{-1}\cdots \times \left(\sum_{j=1}^{m} b_j\mathcal{N}_j(s_1)\right)\mathcal{K}(s_1)^{-1}\mathcal{B}(s_1),$$

with a given direction $b \in \mathbb{C}^m$, and let the scaled and summed transfer function of the reduced-order model be denoted by $\widehat{\widetilde{\mathcal{G}}}_{\mathrm{B},k}(s_1,\ldots,s_k)$. The new right tangential interpolation problem can then be written as

$$\widetilde{\mathcal{G}}_{\mathrm{B},k}(\sigma_1,\ldots,\sigma_k)b = \widehat{\widetilde{\mathcal{G}}}_{\mathrm{B},k}(\sigma_1,\ldots,\sigma_k)b, \tag{5.62}$$

for a given set of interpolation points $\sigma_1,\ldots,\sigma_k \in \mathbb{C}$. Again motivated by duality, the left and two-sided tangential interpolation problems are chosen to be

$$c^{\mathsf{H}}\widetilde{\mathcal{G}}_{\mathrm{B},k}(\sigma_1,\ldots,\sigma_k) = c^{\mathsf{H}}\widehat{\widetilde{\mathcal{G}}}_{\mathrm{B},k}(\sigma_1,\ldots,\sigma_k), \tag{5.63}$$

$$c^{\mathsf{H}}\widetilde{\mathcal{G}}_{\mathrm{B},k}(\sigma_1,\ldots,\sigma_k)b = c^{\mathsf{H}}\widehat{\widetilde{\mathcal{G}}}_{\mathrm{B},k}(\sigma_1,\ldots,\sigma_k)b, \tag{5.64}$$

respectively.

Remark 5.16 (Relation to other control systems):
The idea of the time domain interpretation of tangential interpolation offers a wide range of applications. It can easily be transferred for other types of control systems, e.g., systems with polynomial nonlinearities, such that it may be used to develop new and efficient tangential interpolation approaches for nonlinear MIMO systems. ◊

5.6.3 Structured tangential interpolation framework

With the two re-interpretations of tangential interpolation from Sections 5.6.1 and 5.6.2 in mind, in the following, a unifying framework is developed, which allows for an interpolation theory covering the problems from Sections 5.6.1 and 5.6.2, as well as the blockwise tangential approach from [31, 160], for structured bilinear control systems. Therefore, define the *modified subsystem transfer functions* to be:

$$\mathtt{G}_{\mathrm{B},k}(s_1,\ldots,s_k \,|\, d^{(1)},\ldots,d^{(k-1)}) := \mathcal{C}(s_k)\mathcal{K}(s_k)^{-1}\left(\prod_{j=1}^{k-1}\mathtt{N}(s_{k-j} \,|\, d^{(k-j)}) \times \mathcal{K}(s_{k-j})^{-1}\right)\mathcal{B}(s_1), \tag{5.65}$$

for $k \geq 1$, with the frequency variables $s_1, \ldots, s_k \in \mathbb{C}$, scaling vectors $d^{(1)}, \ldots, d^{(k-1)} \in \mathbb{C}^m$, and where

$$\mathtt{N}(s_j \,|\, d^{(j)}) := \mathcal{N}(s)(d^{(j)} \otimes I_n) = \sum_{i=1}^{m} d_i^{(j)} \mathcal{N}_i(s_j) \tag{5.66}$$

denotes the linear combination of the single matrix functions representing the bilinear terms. Note that the first modified subsystem transfer function does not depend on scaling vectors and corresponds again to the linear case since

$$\mathtt{G}_{\mathrm{B},1}(s_1) = \mathcal{G}_{\mathrm{B},1}(s_1) = \mathcal{G}_{\mathrm{L}}(s_1).$$

In this setting, the modified transfer functions of reduced-order models will be denoted by $\widehat{\mathtt{G}}_{\mathrm{B},k}(s_1, \ldots, s_k \,|\, d^{(1)}, \ldots, d^{(k-1)})$. The resulting tangential interpolation problem for modified transfer functions reads as follows: For a given set of interpolation points $\sigma_1, \ldots, \sigma_k \in \mathbb{C}$, scaling vectors $d^{(1)}, \ldots, d^{(k-1)} \in \mathbb{C}^m$, and right and left tangential directions $b \in C^m$ and $c \in \mathbb{C}^p$, find a reduced-order model such that

$$\begin{aligned}
\mathtt{G}_{\mathrm{B},k}(\sigma_1, \ldots, \sigma_k \,|\, d^{(1)}, \ldots, d^{(k-1)})b &= \widehat{\mathtt{G}}_{\mathrm{B},k}(\sigma_1, \ldots, \sigma_k \,|\, d^{(1)}, \ldots, d^{(k-1)})b, \\
c^{\mathsf{H}}\mathtt{G}_{\mathrm{B},k}(\sigma_1, \ldots, \sigma_k \,|\, d^{(1)}, \ldots, d^{(k-1)}) &= c^{\mathsf{H}}\widehat{\mathtt{G}}_{\mathrm{B},k}(\sigma_1, \ldots, \sigma_k \,|\, d^{(1)}, \ldots, d^{(k-1)}), \quad \text{or} \\
c^{\mathsf{H}}\mathtt{G}_{\mathrm{B},k}(\sigma_1, \ldots, \sigma_k \,|\, d^{(1)}, \ldots, d^{(k-1)})b &= c^{\mathsf{H}}\widehat{\mathtt{G}}_{\mathrm{B},k}(\sigma_1, \ldots, \sigma_k \,|\, d^{(1)}, \ldots, d^{(k-1)})b
\end{aligned} \tag{5.67}$$

hold. The following corollary summarizes some motivated choices of scaling vectors.

Corollary 5.17 (Motivated choices of the scaling vectors):
With an appropriate choice of scaling vectors $d^{(j)}$ in (5.65), different tangential interpolation problems can be recovered from (5.67):

(a) Choosing $d^{(1)} = \ldots = d^{(k-1)} = \mathbb{1}_m$ yields the extension of classical tangential interpolation to the subsystem transfer functions of bilinear systems (5.49)–(5.51) from Section 5.6.1.

(b) Choosing $d^{(1)} = \ldots = d^{(k-1)} = b$, with $b \in \mathbb{C}^m$ the right tangential direction, yields the tangential interpolation problems (5.62)–(5.64) resulting from the time domain interpretation in Section 5.6.2. ◊

The following theorem solves the new tangential interpolation problems (5.67) via conditions for the underlying projection spaces in the projection framework (5.4).

Theorem 5.18 (Bilinear tangential interpolation):
Let $\mathcal{G}_{\mathrm{B}}$ be a bilinear system, with its modified transfer functions $\mathtt{G}_{\mathrm{B},k}$ in (5.65), and $\widehat{\mathcal{G}}_{\mathrm{B}}$ the reduced-order bilinear system constructed by (5.4), with its modified transfer functions $\widehat{\mathtt{G}}_{\mathrm{B},k}$. Given sets of interpolation points $\sigma_1, \ldots, \sigma_k \in \mathbb{C}$ and $\varsigma_1, \ldots, \varsigma_\theta \in \mathbb{C}$, for which the matrix functions $\mathcal{C}$, $\mathcal{K}^{-1}$, $\mathcal{N}$, $\mathcal{B}$ and $\widehat{\mathcal{K}}^{-1}$ are defined, two tangential directions $b \in \mathbb{C}^m$ and $c \in \mathbb{C}^p$, and two sets of scaling vectors $d^{(1)}, \ldots, d^{(k-1)} \in \mathbb{C}^m$ and $\delta^{(1)}, \ldots, \delta^{(\theta-1)} \in \mathbb{C}^m$, the following statements hold:

(a) If V is constructed as

$$\begin{aligned} v_1 &= \mathcal{K}(\sigma_1)^{-1}\mathcal{B}(\sigma_1)b, \\ v_j &= \mathcal{K}(\sigma_j)^{-1}\mathtt{N}(\sigma_{j-1} \,|\, d^{(j-1)})v_{j-1}, \qquad 2 \le j \le k, \\ \operatorname{span}(V) &\supseteq \operatorname{span}\left(\begin{bmatrix} v_1 & \dots & v_k \end{bmatrix}\right), \end{aligned}$$

then the following interpolation conditions hold true:

$$\begin{aligned} \mathtt{G}_{\mathrm{B},1}(\sigma_1)b &= \widehat{\mathtt{G}}_{\mathrm{B},1}(\sigma_1)b, \\ \mathtt{G}_{\mathrm{B},2}(\sigma_1,\sigma_2 \,|\, d^{(1)})b &= \widehat{\mathtt{G}}_{\mathrm{B},2}(\sigma_1,\sigma_2 \,|\, d^{(1)})b, \\ &\vdots \\ \mathtt{G}_{\mathrm{B},k}(\sigma_1,\dots,\sigma_k \,|\, d^{(1)},\dots,d^{(k-1)})b &= \widehat{\mathtt{G}}_{\mathrm{B},k}(\sigma_1,\dots,\sigma_k \,|\, d^{(1)},\dots,d^{(k-1)})b. \end{aligned}$$

(b) If W is constructed as

$$\begin{aligned} w_1 &= \mathcal{K}(\varsigma_\theta)^{-\mathsf{H}}\mathcal{C}(\varsigma_\theta)^{\mathsf{H}}c, \\ w_i &= \mathcal{K}(\varsigma_{\theta-i+1})^{-\mathsf{H}}\mathtt{N}(\varsigma_{\theta-i+1} \,|\, \delta^{(\theta-i+1)})^{\mathsf{H}}w_{i-1}, \qquad 2 \le i \le \theta, \\ \operatorname{span}(W) &\supseteq \operatorname{span}\left(\begin{bmatrix} w_1 & \dots & w_\theta \end{bmatrix}\right), \end{aligned}$$

then the following interpolation conditions hold true:

$$\begin{aligned} c^{\mathsf{H}}\mathtt{G}_{\mathrm{B},1}(\varsigma_\theta) &= c^{\mathsf{H}}\widehat{\mathtt{G}}_{\mathrm{B},1}(\varsigma_\theta), \\ c^{\mathsf{H}}\mathtt{G}_{\mathrm{B},2}(\varsigma_{\theta-1},\varsigma_\theta \,|\, \delta^{(\theta-1)}) &= c^{\mathsf{H}}\widehat{\mathtt{G}}_{\mathrm{B},2}(\varsigma_{\theta-1},\varsigma_\theta \,|\, \delta^{(\theta-1)}), \\ &\vdots \\ c^{\mathsf{H}}\mathtt{G}_{\mathrm{B},\theta}(\varsigma_1,\dots,\varsigma_\theta \,|\, \delta^{(1)},\dots,\delta^{(\theta-1)}) &= c^{\mathsf{H}}\widehat{\mathtt{G}}_{\mathrm{B},\theta}(\varsigma_1,\dots,\varsigma_\theta \,|\, \delta^{(1)},\dots,\delta^{(\theta-1)}). \end{aligned}$$

(c) Let V be constructed as in Part (a) and W as in Part (b), then, additionally to the results in (a) and (b), the following conditions hold:

$$\begin{aligned} &c^{\mathsf{H}}\mathtt{G}_{\mathrm{B},q+\eta}(\sigma_1,\dots,\sigma_q,\varsigma_{\theta-\eta+1},\dots,\varsigma_\theta \,|\, d^{(1)},\dots,d^{(q-1)},z,\delta^{(\theta-\eta+1)},\dots,\delta^{(\theta-1)})b \\ &= c^{\mathsf{H}}\widehat{\mathtt{G}}_{\mathrm{B},q+\eta}(\sigma_1,\dots,\sigma_q,\varsigma_{\theta-\eta+1},\dots,\varsigma_\theta \,|\, d^{(1)},\dots,d^{(q-1)},z,\delta^{(\theta-\eta+1)},\dots,\delta^{(\theta-1)})b, \end{aligned}$$

for $1 \le q \le k$, $1 \le \eta \le \theta$, and an additional arbitrary scaling vector $z \in \mathbb{C}^m$. ◊

Proof. Parts (a) and (b) follow directly from Theorem 5.9 by using the vector-valued inputs $\mathcal{B}(s)b$ and outputs $c^{\mathsf{H}}\mathcal{C}(s)$. One can observe that for given fixed scaling vectors $d^{(1)},\dots,d^{(k-1)}$ and $\delta^{(1)},\dots,\delta^{(\theta-1)}$, the modified bilinear terms (5.66) are functions depending on a single frequency variable. Therefore, a single block entry of a full MIMO subsystem transfer function is resembled.

To prove Part (c), the modified transfer functions of the reduced-order model are given by

$$\begin{aligned}
&c^{\mathsf{H}}\widehat{\mathbb{G}}_{\mathrm{B},q+\eta}(\sigma_1,\ldots,\sigma_q,\varsigma_{\theta-\eta+1},\ldots,\varsigma_\theta \,|\, d^{(1)},\ldots,d^{(q-1)},z,\delta^{(\theta-\eta+1)},\ldots,\delta^{(\theta-1)})b \\
&= c^{\mathsf{H}}\widehat{\mathcal{C}}(\varsigma_\theta)\widehat{\mathcal{K}}(\varsigma_\theta)^{-1}\left(\prod_{i=1}^{\eta-1}\widehat{\mathbb{N}}(\varsigma_{\theta-i}\,|\,\delta^{(\theta-i)})\widehat{\mathcal{K}}(\varsigma_{\theta-i})^{-1}\right)\widehat{\mathbb{N}}(\sigma_q\,|\,z) \\
&\quad\times\left(\prod_{j=0}^{q-2}\widehat{\mathcal{K}}(\sigma_{q-j})^{-1}\widehat{\mathbb{N}}(\sigma_{q-j-1}\,|\,d^{(q-j-1)})\right)\widehat{\mathcal{K}}(\sigma_1)^{-1}\widehat{\mathcal{B}}(\sigma_1)b \\
&=: \hat{w}_\eta^{\mathsf{H}}\widehat{\mathbb{N}}(\sigma_q\,|\,z)\hat{v}_q \\
&= \hat{w}_\eta^{\mathsf{H}}W^{\mathsf{H}}\mathbb{N}(\sigma_q\,|\,z)V\hat{v}_q,
\end{aligned}$$

for $1 \leq q \leq k$; $1 \leq \eta \leq \theta$ and an arbitrary scaling vector $z \in \mathbb{C}^m$. The right factor can then be rewritten using the construction of V such that

$$\begin{aligned}
V\hat{v}_q &= V\left(\prod_{j=0}^{q-3}\widehat{\mathcal{K}}(\sigma_{q-j})^{-1}\widehat{\mathbb{N}}(\sigma_{q-j-1}\,|\,d^{(q-j-1)})\right)\widehat{\mathcal{K}}(\sigma_2)^{-1}\widehat{\mathbb{N}}(\sigma_1\,|\,d^{(1)})\widehat{\mathcal{K}}(\sigma_1)^{-1}\widehat{\mathcal{B}}(\sigma_1)b \\
&= V\left(\prod_{j=0}^{q-3}\widehat{\mathcal{K}}(\sigma_{q-j})^{-1}\widehat{\mathbb{N}}(\sigma_{q-j-1}\,|\,d^{(q-j-1)})\right)\widehat{\mathcal{K}}(\sigma_2)^{-1}W^{\mathsf{H}}\mathbb{N}(\sigma_1\,|\,d^{(1)}) \\
&\quad\times \underbrace{V\widehat{\mathcal{K}}(\sigma_1)^{-1}W^{\mathsf{H}}\mathcal{K}(\sigma_1)}_{=P_{\mathrm{V}}(\sigma_1)}\underbrace{\mathcal{K}(\sigma_1)^{-1}\mathcal{B}(\sigma_1)b}_{=v_1} \\
&= V\left(\prod_{j=0}^{q-3}\widehat{\mathcal{K}}(\sigma_{q-j})^{-1}\widehat{\mathbb{N}}(\sigma_{q-j-1}\,|\,d^{(q-j-1)})\right)\widehat{\mathcal{K}}(\sigma_2)^{-1}W^{\mathsf{H}}\mathbb{N}(\sigma_1\,|\,d^{(1)})v_1 \\
&= \ldots \\
&= V\widehat{\mathcal{K}}(\sigma_q)^{-1}W^{\mathsf{H}}\mathbb{N}(\sigma_{q-1}\,|\,d^{(q-1)})v_{q-1} \\
&= \underbrace{V\widehat{\mathcal{K}}(\sigma_q)^{-1}W^{\mathsf{H}}\mathcal{K}(\sigma_q)}_{=:P_{\mathrm{V}}(\sigma_q)}\underbrace{\mathcal{K}(\sigma_q)^{-1}\mathbb{N}(\sigma_{q-1}\,|\,d^{(q-1)})v_{q-1}}_{=v_q} \\
&= v_q,
\end{aligned}$$

where $P_{\mathrm{V}}(\sigma_1),\ldots,P_{\mathrm{V}}(\sigma_q)$ are the projectors onto $\operatorname{span}(V)$ from (3.24). Analogously, one can show the identity

$$W\hat{w}_\eta = w_\eta,$$

by constructing the projectors (3.25) onto $\operatorname{span}(W)$ and using $w_1,\ldots,w_\eta \in \operatorname{span}(W)$.

Combining the identities yields

$$\begin{aligned}
&c^{\mathsf{H}}\widehat{\mathtt{G}}_{\mathrm{B},q+\eta}(\sigma_1,\ldots,\sigma_q,\varsigma_{\theta-\eta+1},\ldots,\varsigma_\theta \,|\, d^{(1)},\ldots,d^{(q-1)},z,\delta^{(\theta-\eta+1)},\ldots,\delta^{(\theta-1)})b \\
&= \hat{w}_\eta^{\mathsf{H}} W^{\mathsf{H}} \mathtt{N}(\sigma_q \,|\, z) V \hat{v}_q \\
&= w_\eta^{\mathsf{H}} \mathtt{N}(\sigma_q \,|\, z) v_q \\
&= c^{\mathsf{H}}\mathtt{G}_{\mathrm{B},q+\eta}(\sigma_1,\ldots,\sigma_q,\varsigma_{\theta-\eta+1},\ldots,\varsigma_\theta \,|\, d^{(1)},\ldots,d^{(q-1)},z,\delta^{(\theta-\eta+1)},\ldots,\delta^{(\theta-1)})b,
\end{aligned}$$

for $1 \leq q \leq k$; $1 \leq \eta \leq \theta$ and an arbitrary scaling vector $z \in \mathbb{C}^m$. □

Theorem 5.18 Part (c) is an interesting result, as the modified bilinear term in the middle between the interpolation by left and right projection allows for a completely arbitrary scaling vector. Especially, by using certain realizations of z, blockwise interpolation conditions hold true corresponding to the centering bilinear term, as the following example demonstrates: With Theorem 5.18, construct $\operatorname{span}(V)$ and $\operatorname{span}(W)$ such that $\mathtt{G}_{\mathrm{B},1}(\sigma)b$ and $c^{\mathsf{H}}\mathtt{G}_{\mathrm{B},1}(\varsigma)$ are interpolated for chosen interpolation points $\sigma, \varsigma \in \mathbb{C}$, and tangential directions $b \in \mathbb{C}^m$ and $c \in \mathbb{C}^p$. Then, by two-sided projection it holds additionally

$$c^{\mathsf{H}}\mathtt{G}_{\mathrm{B},2}(\sigma,\varsigma \,|\, z)b = c^{\mathsf{H}}\widehat{\mathtt{G}}_{\mathrm{B},2}(\sigma,\varsigma \,|\, z)b,$$

for all $z \in \mathbb{C}^m$. Especially, choosing $z = \begin{bmatrix} 1 & 0 \end{bmatrix}^{\mathsf{T}}$ and $z = \begin{bmatrix} 0 & 1 \end{bmatrix}^{\mathsf{T}}$ yields the blockwise two-sided tangential interpolation condition

$$c^{\mathsf{H}}\mathcal{G}_{\mathrm{B},2}(\sigma,\varsigma)(I_m \otimes b) = c^{\mathsf{H}}\widehat{\mathcal{G}}_{\mathrm{B},2}(\sigma,\varsigma)(I_m \otimes b);$$

cf. Section 5.6.4.

Besides matching transfer function values, in practice, the interpolation of partial derivatives with respect to the frequency points is important as it can improve the approximation quality of the computed reduced-order model around the chosen interpolation points significantly. The following theorem states conditions on the projection spaces to satisfy tangential Hermite interpolation conditions.

Theorem 5.19 (Bilinear tangential Hermite interpolation):
Let $\mathcal{G}_{\mathrm{B}}$ be a bilinear system, with its modified transfer functions $\mathtt{G}_{\mathrm{B},k}$ in (5.65), and $\widehat{\mathcal{G}}_{\mathrm{B}}$ the reduced-order bilinear system constructed by (5.4), with its modified transfer functions $\widehat{\mathtt{G}}_{\mathrm{B},k}$. Given sets of interpolation points $\sigma_1,\ldots,\sigma_k \in \mathbb{C}$ and $\varsigma_1,\ldots,\varsigma_\theta \in \mathbb{C}$, for which the matrix functions $\mathcal{C}$, $\mathcal{K}^{-1}$, $\mathcal{N}$, $\mathcal{B}$ and $\widehat{\mathcal{K}}^{-1}$ are complex differentiable, orders of partial derivatives $\ell_1,\ldots,\ell_k \in \mathbb{N}_0$ and $\nu_1,\ldots,\nu_\theta \in \mathbb{N}_0$, two tangential directions $b \in \mathbb{C}^m$ and $c \in \mathbb{C}^p$, and two sets of scaling vectors $d^{(1)},\ldots,d^{(k-1)} \in \mathbb{C}^m$ and $\delta^{(1)},\ldots,\delta^{(\theta-1)} \in \mathbb{C}^m$, the following statements hold:

(a) If V is constructed as

$$\begin{aligned}
v_{1,j_1} &= \partial_{s^{j_1}}\Big(\mathcal{K}^{-1}\mathcal{B}\Big)(\sigma_1)b, && j_1 = 0,\ldots,\ell_1,\\
v_{2,j_2} &= \partial_{s^{j_2}}\mathcal{K}^{-1}(\sigma_2)\partial_{s^{\ell_1}}\Big(\mathtt{N}(.\,|\,d^{(1)})\mathcal{K}^{-1}\mathcal{B}\Big)(\sigma_1)b, && j_2 = 0,\ldots,\ell_2,\\
&\vdots\\
v_{k,j_k} &= \partial_{s^{j_k}}\mathcal{K}^{-1}(\sigma_k)\left(\prod_{j=1}^{k-2}\partial_{s^{\ell_{k-j}}}\Big(\mathtt{N}(.\,|\,d^{(k-j)})\mathcal{K}^{-1}\Big)(\sigma_{k-j})\right)\\
&\quad\times\partial_{s^{\ell_1}}\Big(\mathtt{N}(.\,|\,d^{(1)})\mathcal{K}^{-1}\mathcal{B}\Big)(\sigma_1)b, && j_k = 0,\ldots,\ell_k,\\
\operatorname{span}(V) &\supseteq \operatorname{span}\left(\begin{bmatrix} v_{1,0} & \ldots & v_{k,\ell_k}\end{bmatrix}\right),
\end{aligned}$$

then the following interpolation conditions hold true:

$$\begin{aligned}
\partial_{s_1^{j_1}}\mathtt{G}_{\mathrm{B},1}(\sigma_1)b &= \partial_{s_1^{j_1}}\widehat{\mathtt{G}}_{\mathrm{B},1}(\sigma_1)b, && j_1 = 0,\ldots,\ell_1,\\
\partial_{s_1^{\ell_1}s_2^{j_2}}\mathtt{G}_{\mathrm{B},2}(\sigma_1,\sigma_2\,|\,d^{(1)})b &= \partial_{s_1^{\ell_1}s_2^{j_2}}\widehat{\mathtt{G}}_{\mathrm{B},2}(\sigma_1,\sigma_2\,|\,d^{(1)})b, && j_2 = 0,\ldots,\ell_2,\\
&\vdots\\
&\partial_{s_1^{\ell_1}\ldots s_{k-1}^{\ell_{k-1}}s_k^{j_k}}\mathtt{G}_{\mathrm{B},k}(\sigma_1,\ldots,\sigma_k\,|\,d^{(1)},\ldots,d^{(k-1)})b\\
&= \partial_{s_1^{\ell_1}\ldots s_{k-1}^{\ell_{k-1}}s_k^{j_k}}\widehat{\mathtt{G}}_{\mathrm{B},k}(\sigma_1,\ldots,\sigma_k\,|\,d^{(1)},\ldots,d^{(k-1)})b, && j_k = 0,\ldots,\ell_k.
\end{aligned}$$

(b) If W is constructed as

$$\begin{aligned}
w_{1,i_\theta} &= \partial_{s^{i_\theta}}\Big(\mathcal{K}^{-\mathsf{H}}\mathcal{C}^{\mathsf{H}}\Big)(\varsigma_\theta)c, && i_\theta = 0,\ldots,\nu_\theta,\\
w_{2,i_{\theta-1}} &= \partial_{s^{i_{\theta-1}}}\Big(\mathcal{K}^{-\mathsf{H}}\mathtt{N}(.\,|\,\delta^{(\theta-1)})^{\mathsf{H}}\Big)(\varsigma_{\theta-1})w_{1,\nu_\theta}, && i_{\theta-1} = 0,\ldots,\nu_{\theta-1},\\
&\vdots\\
w_{\theta,i_1} &= \partial_{s^{i_1}}\Big(\mathcal{K}^{-\mathsf{H}}\mathtt{N}(.\,|\,\delta^{(1)})^{\mathsf{H}}\Big)(\varsigma_1)w_{\theta-1,\nu_2}, && i_1 = 0,\ldots,\nu_1,\\
\operatorname{span}(W) &\supseteq \operatorname{span}\left(\begin{bmatrix} w_{1,0} & \ldots & w_{\theta,\nu_\theta}\end{bmatrix}\right),
\end{aligned}$$

then the following interpolation conditions hold true:

$$
\begin{aligned}
&c^{\mathsf{H}}\partial_{s_1^{i_\theta}}\mathtt{G}_{\mathrm{B},1}(\varsigma_\theta) = c^{\mathsf{H}}\partial_{s_1^{i_\theta}}\widehat{\mathtt{G}}_{\mathrm{B},1}(\varsigma_\theta), && i_\theta = 0,\dots,\nu_\theta,\\
&c^{\mathsf{H}}\partial_{s_1^{i_{\theta-1}}s_2^{\nu_\theta}}\mathtt{G}_{\mathrm{B},2}(\varsigma_{\theta-1},\varsigma_\theta \mid \delta^{(\theta-1)})\\
&= c^{\mathsf{H}}\partial_{s_1^{i_{\theta-1}}s_2^{\nu_\theta}}\widehat{\mathtt{G}}_{\mathrm{B},2}(\varsigma_{\theta-1},\varsigma_\theta \mid \delta^{(\theta-1)}), && i_{\theta-1} = 0,\dots,\nu_{\theta-1},\\
&\qquad\qquad\vdots\\
&c^{\mathsf{H}}\partial_{s_1^{i_1}s_2^{\nu_2}\cdots s_\theta^{\nu_\theta}}\mathtt{G}_{\mathrm{B},\theta}(\varsigma_1,\dots,\varsigma_\theta \mid \delta^{(1)},\dots,\delta^{(\theta-1)})\\
&= c^{\mathsf{H}}\partial_{s_1^{i_1}s_2^{\nu_2}\cdots s_\theta^{\nu_\theta}}\widehat{\mathtt{G}}_{\mathrm{B},\theta}(\varsigma_1,\dots,\varsigma_\theta \mid \delta^{(1)},\dots,\delta^{(\theta-1)}), && i_1 = 0,\dots,\nu_1.
\end{aligned}
$$

(c) Let V be constructed as in Part (a) and W as in Part (b), then, additionally to the results in (a) and (b), the following conditions hold:

$$
\begin{aligned}
&c^{\mathsf{H}}\partial_{s_1^{\ell_1}\cdots s_{q-1}^{\ell_{q-1}}s_q^{j_q}s_{q+1}^{i_{\theta-\eta+1}}s_{q+2}^{\nu_{\theta-\eta+2}}s_{q+\eta}^{\nu_\theta}}\mathtt{G}_{\mathrm{B},q+\eta}(\sigma_1,\dots,\sigma_q,\varsigma_{\theta-\eta+1},\dots,\varsigma_\theta \mid\\
&\qquad d^{(1)},\dots,d^{(q-1)},z,\delta^{(\theta-\eta+1)},\dots,\delta^{(\theta-1)})b\\
&= c^{\mathsf{H}}\partial_{s_1^{\ell_1}\cdots s_{q-1}^{\ell_{q-1}}s_q^{j_q}s_{q+1}^{i_{\theta-\eta+1}}s_{q+2}^{\nu_{\theta-\eta+2}}s_{q+\eta}^{\nu_\theta}}\widehat{\mathtt{G}}_{\mathrm{B},q+\eta}(\sigma_1,\dots,\sigma_q,\varsigma_{\theta-\eta+1},\dots,\varsigma_\theta \mid\\
&\qquad d^{(1)},\dots,d^{(q-1)},z,\delta^{(\theta-\eta+1)},\dots,\delta^{(\theta-1)})b,
\end{aligned}
$$

for $j_q = 0,\dots,\ell_q$; $i_{\theta-\eta+1} = 0,\dots,\nu_{\theta-\eta+1}$; $1 \le q \le k$; $1 \le \eta \le \theta$ and an additional arbitrary scaling vector $z \in \mathbb{C}^m$. ◊

Proof. The proof follows the ideas of the proofs of Theorems 5.4, 5.5, 5.7 and 5.18 using the projectors (3.24) and (3.25) onto either $\operatorname{span}(V)$ or $\operatorname{span}(W)$. □

To complete the theory for the new tangential interpolation framework, the special cases of Theorems 5.18 and 5.19 by using identical sets of interpolation points and scaling vectors in the two-sided tangential interpolation case is left. As in Proposition 3.2, Theorem 5.6 and Corollaries 5.8 and 5.11, this allows to implicitly interpolate partial derivatives. Due to the dependency of the modified transfer functions on the scaling vectors, also partial derivatives with respect to the scaling vectors will be considered for interpolation, very similar to the results in the parametric system case (Theorem 5.15). For the following theorem, the full Jacobi matrix (2.6) for the modified transfer functions is given by

$$
\nabla\mathtt{G}_{\mathrm{B},k} = \left[\partial_{s_1}\mathtt{G}_{\mathrm{B},k},\dots,\partial_{s_k}\mathtt{G}_{\mathrm{B},k},\partial_{d_1^{(1)}}\mathtt{G}_{\mathrm{B},k},\dots,\partial_{d_m^{(1)}}\mathtt{G}_{\mathrm{B},k},\dots,\partial_{d_1^{(k-1)}}\mathtt{G}_{\mathrm{B},k},\dots,\partial_{d_m^{(k-1)}}\mathtt{G}_{\mathrm{B},k}\right].
$$

Theorem 5.20 (Implicit bilinear tangential interpolation):
Let $\mathcal{G}_{\mathrm{B}}$ be a bilinear system, with its modified transfer functions $\mathtt{G}_{\mathrm{B},k}$ in (5.65), and $\widehat{\mathcal{G}}_{\mathrm{B}}$ the reduced-order bilinear system constructed by (5.4), with its modified transfer functions $\widehat{\mathtt{G}}_{\mathrm{B},k}$. Given a set of interpolation points $\sigma_1, \ldots, \sigma_k \in \mathbb{C}$, for which the matrix functions $\mathcal{C}$, $\mathcal{K}^{-1}$, $\mathcal{N}$, $\mathcal{B}$ and $\widehat{\mathcal{K}}^{-1}$ are complex differentiable, two tangential directions $b \in \mathbb{C}^m$ and $c \in \mathbb{C}^p$, and scaling vectors $d^{(1)}, \ldots, d^{(k-1)} \in \mathbb{C}^m$, the following statements hold:

(a) Let V and W be constructed as in Theorem 5.18 Parts (a) and (b) for matching interpolation points $\sigma_1 = \varsigma_1$, …, $\sigma_k = \varsigma_k$ and the scaling vectors $d^{(1)} = \delta^{(1)}$, …, $d^{(k-1)} = \delta^{(k-1)}$, then additionally it holds

$$\begin{aligned}&\nabla \left(c^{\mathsf{H}} \mathtt{G}_{\mathrm{B},k} b\right)(\sigma_1, \ldots, \sigma_k \,|\, d^{(1)}, \ldots, d^{(k-1)})\\ &= \nabla \left(c^{\mathsf{H}} \widehat{\mathtt{G}}_{\mathrm{B},k} b\right)(\sigma_1, \ldots, \sigma_k \,|\, d^{(1)}, \ldots, d^{(k-1)}).\end{aligned}$$

(b) Let V and W be constructed as in Theorem 5.19 Parts (a) and (b) for matching interpolation points $\sigma_1 = \varsigma_1$, …, $\sigma_k = \varsigma_k$, scaling vectors $d^{(1)} = \delta^{(1)}$, …, $d^{(k-1)} = \delta^{(k-1)}$ and orders of partial derivatives $\ell_1 = \nu_1$, …, $\ell_k = \nu_k$, then additionally it holds

$$\begin{aligned}&\nabla \left(c^{\mathsf{H}} \partial_{s_1^{\ell_1} \cdots s_k^{\ell_k}} \mathtt{G}_{\mathrm{B},k} b\right)(\sigma_1, \ldots, \sigma_k \,|\, d^{(1)}, \ldots, d^{(k-1)})\\ &= \nabla \left(c^{\mathsf{H}} \partial_{s_1^{\ell_1} \cdots s_k^{\ell_k}} \widehat{\mathtt{G}}_{\mathrm{B},k} b\right)(\sigma_1, \ldots, \sigma_k \,|\, d^{(1)}, \ldots, d^{(k-1)}).\end{aligned}$$ ◊

Proof. The proof of Part (b) is analogous to Part (a) by replacing the simple interpolation by the Hermite conditions from Theorem 5.19. Therefore, it is enough to prove Part (a). First, consider the partial derivatives with respect to the scaling vectors. For arbitrary $1 \leq j \leq k-1$ and $1 \leq i \leq m$, it holds

$$\begin{aligned}&\partial_{d_i^{(j)}} \left(c^{\mathsf{H}} \widehat{\mathtt{G}}_{\mathrm{B},k} b\right)(\sigma_1, \ldots, \sigma_k \,|\, d^{(1)}, \ldots, d^{(k-1)})\\ &= c^{\mathsf{H}} \widehat{\mathcal{C}}(\sigma_k) \widehat{\mathcal{K}}(\sigma_k)^{-1} \left(\prod_{\ell=1}^{k-j-1} \widehat{\mathtt{N}}(\sigma_{k-\ell} \,|\, d^{(k-\ell)}) \widehat{\mathcal{K}}(\sigma_{k-\ell})^{-1}\right) \left(\partial_{d_i^{(j)}} \widehat{\mathtt{N}}(\sigma_j \,|\, d^{(j)})\right)\\ &\quad \times \left(\prod_{\ell=j+1}^{k-1} \widehat{\mathtt{N}}(\sigma_{k-\ell} \,|\, d^{(k-\ell)}) \widehat{\mathcal{K}}(\sigma_{k-\ell})^{-1}\right) \widehat{\mathcal{B}}(s_1) b\\ &=: \hat{w}_{k-j-1}^{\mathsf{H}} \left(\partial_{d_i^{(j)}} \widehat{\mathtt{N}}(\sigma_j \,|\, d^{(j)})\right) \hat{v}_{k-j-1}\\ &= \hat{w}_{k-j-1}^{\mathsf{H}} W^{\mathsf{H}} \left(\partial_{d_i^{(j)}} \mathtt{N}(\sigma_j \,|\, d^{(j)})\right) V \hat{v}_{k-j-1}\end{aligned}$$

such that only the modified bilinear term corresponding to the scaling vector $d^{(j)}$ needs to be differentiated. Using exactly the approach from the proof of Theorem 5.18 and the construction of $\operatorname{span}(V)$ and $\operatorname{span}(W)$ yields the two identities

$$V\hat{v}_{k-j-1} = v_{k-j-1} \quad \text{and} \quad W\hat{w}_{k-j-1} = w_{k-j-1},$$

which give the Hermite interpolation condition

$$\begin{aligned}
&\partial_{d_i^{(j)}} \left(c^{\mathsf{H}} \widehat{\mathbb{G}}_{\mathrm{B},k} b\right)(\sigma_1, \dots, \sigma_k \,|\, d^{(1)}, \dots, d^{(k-1)}) \\
&= \hat{w}_{k-j-1}^{\mathsf{H}} W^{\mathsf{H}} \left(\partial_{d_i^{(j)}} \mathbb{N}(\sigma_j \,|\, d^{(j)})\right) V \hat{v}_{k-j-1} \\
&= w_{k-j-1}^{\mathsf{H}} \left(\partial_{d_i^{(j)}} \mathbb{N}(\sigma_j \,|\, d^{(j)})\right) v_{k-j-1} \\
&= \partial_{d_i^{(j)}} \left(c^{\mathsf{H}} \mathbb{G}_{\mathrm{B},k} b\right)(\sigma_1, \dots, \sigma_k \,|\, d^{(1)}, \dots, d^{(k-1)}),
\end{aligned}$$

for all $1 \leq j \leq k-1$ and $1 \leq i \leq m$. Therefore, the interpolation of all partial derivatives with respect to the scaling vectors holds. The interpolation of the partial derivatives with respect to the frequency arguments can be proven in the same fashion but, in principle, follows directly from Corollary 5.11. □

5.6.4 Special case: Structured blockwise tangential interpolation

As mentioned before, the new tangential interpolation framework for structured bilinear systems from Section 5.6.3 also covers the case of blockwise tangential interpolation. Due to its relevance in the literature [31, 160], the blockwise tangential interpolation results are summarized here for the structured bilinear system case.

As first step, the blockwise tangential interpolation problem as in (5.46) needs to be generalized to the structured system case. Therefore, take a look at the structured subsystem transfer functions in the MIMO system case (5.3). Multiplying out the Kronecker products in (5.3) yields the column concatenation of products of the dynamics and the bilinear terms

$$\begin{aligned}
&\mathcal{G}_{\mathrm{B},k}(s_1, \dots, s_k) \\
&= \Big[\mathcal{C}(s_k)\mathcal{K}(s_k)^{-1}\mathcal{N}_1(s_{k-1})\mathcal{K}(s_{k-1})^{-1} \cdots \mathcal{N}_1(s_1)\mathcal{K}(s_1)^{-1}\mathcal{B}(s_1), \\
&\quad \mathcal{C}(s_k)\mathcal{K}(s_k)^{-1}\mathcal{N}_1(s_{k-1})\mathcal{K}(s_{k-1})^{-1} \cdots \mathcal{N}_2(s_1)\mathcal{K}(s_1)^{-1}\mathcal{B}(s_1), \\
&\quad \dots \\
&\quad \mathcal{C}(s_k)\mathcal{K}(s_k)^{-1}\mathcal{N}_m(s_{k-1})\mathcal{K}(s_{k-1})^{-1} \cdots \mathcal{N}_m(s_1)\mathcal{K}(s_1)^{-1}\mathcal{B}(s_1)\Big].
\end{aligned} \tag{5.68}$$

Now, each block entry of (5.68) is considered as a separate transfer function such that tangential interpolation can be used for each of them with the same tangential directions.

For example, given the right tangential direction $b \in \mathbb{C}^m$,

$$\begin{aligned}
&\mathcal{G}_{\mathrm{B},k}(s_1,\ldots,s_k)(I_m \otimes b) \\
&= \Big[\mathcal{C}(s_k)\mathcal{K}(s_k)^{-1}\mathcal{N}_1(s_{k-1})\mathcal{K}(s_{k-1})^{-1}\cdots\mathcal{N}_1(s_1)\mathcal{K}(s_1)^{-1}\mathcal{B}(s_1)b, \\
&\quad \mathcal{C}(s_k)\mathcal{K}(s_k)^{-1}\mathcal{N}_1(s_{k-1})\mathcal{K}(s_{k-1})^{-1}\cdots\mathcal{N}_2(s_1)\mathcal{K}(s_1)^{-1}\mathcal{B}(s_1)b, \\
&\quad \ldots \\
&\quad \mathcal{C}(s_k)\mathcal{K}(s_k)^{-1}\mathcal{N}_m(s_{k-1})\mathcal{K}(s_{k-1})^{-1}\cdots\mathcal{N}_m(s_1)\mathcal{K}(s_1)^{-1}\mathcal{B}(s_1)b\Big]
\end{aligned}$$

is the blockwise evaluation of the transfer function in the direction b.

The general tangential interpolation framework from Section 5.6.3 can now be used to construct results for blockwise tangential interpolation in the following way: Choose the scaling vectors $d^{(j)}$ in (5.65) to be columns of the m-dimensional identity matrix I_m. Then, the single block entries of (5.68) are equivalently given by the modified transfer functions (5.65) using different combinations of the possible scaling vectors. For example, choosing $d^{(1)} = \ldots = d^{(k-1)} = e_1$ as the first column of I_m yields

$$\mathtt{G}_{\mathrm{B},k}(s_1,\ldots,s_k \,|\, e_1,\ldots,e_1) = \mathcal{C}(s_k)\mathcal{K}(s_k)^{-1}\mathcal{N}_1(s_{k-1})\mathcal{K}(s_{k-1})^{-1}\cdots\mathcal{N}_1(s_1)\mathcal{K}(s_1)^{-1}\mathcal{B}(s_1),$$

which is the first block entry in (5.68). Concatenation of these modified transfer functions results in the recovery of (5.68) by

$$\begin{aligned}
\mathcal{G}_{\mathrm{B},k}(s_1,\ldots,s_1) = \Big[&\mathtt{G}_{\mathrm{B},k}(s_1,\ldots,s_k \,|\, e_1,\ldots,e_1), \\
&\mathtt{G}_{\mathrm{B},k}(s_1,\ldots,s_k \,|\, e_1,\ldots,e_2), \\
&\ldots, \\
&\mathtt{G}_{\mathrm{B},k}(s_1,\ldots,s_k \,|\, e_m,\ldots,e_m)\Big].
\end{aligned}$$

Consequently, the blockwise tangential interpolation results are given by the concatenation of the projection space bases in Theorems 5.18 to 5.20 for all possible combinations of the columns of I_m as scaling vectors.

For practical usage of the blockwise tangential interpolation for structured bilinear systems, the theoretical results are stated below in three corollaries following the same structure as the tangential interpolation in Section 5.6.3. Due to the argumentation above, all proofs of the blockwise tangential interpolation results are omitted.

Corollary 5.21 (Bilinear blockwise tangential interpolation):
Let $\mathcal{G}_{\mathrm{B}}$ be a bilinear system, described by its subsystem transfer functions in (5.3), and $\widehat{\mathcal{G}}_{\mathrm{B}}$ the reduced-order bilinear system constructed by (5.4), with the corresponding subsystem transfer functions $\widehat{\mathcal{G}}_{\mathrm{B},k}$. Given sets of interpolation points $\sigma_1,\ldots,\sigma_k \in \mathbb{C}$ and $\varsigma_1,\ldots,\varsigma_\theta \in \mathbb{C}$, for which the matrix functions $\mathcal{C}$, $\mathcal{K}^{-1}$, $\mathcal{N}$, $\mathcal{B}$ and $\widehat{\mathcal{K}}^{-1}$ are defined, and two tangential directions $b \in \mathbb{C}^m$ and $c \in \mathbb{C}^p$, the following statements hold:

(a) If V is constructed as

$$\begin{aligned} V_1 &= \mathcal{K}(\sigma_1)^{-1}\mathcal{B}(\sigma_1)b, \\ V_j &= \mathcal{K}(\sigma_j)^{-1}\mathcal{N}(\sigma_{j-1})(I_m \otimes V_{j-1}), \qquad 2 \le j \le k, \\ \operatorname{span}(V) &\supseteq \operatorname{span}\left(\begin{bmatrix} V_1 & \dots & V_k \end{bmatrix}\right), \end{aligned}$$

then the following interpolation conditions hold true:

$$\begin{aligned} \mathcal{G}_{\mathrm{B},1}(\sigma_1)b &= \widehat{\mathcal{G}}_{\mathrm{B},1}(\sigma_1)b, \\ \mathcal{G}_{\mathrm{B},2}(\sigma_1,\sigma_2)(I_m \otimes b) &= \widehat{\mathcal{G}}_{\mathrm{B},2}(\sigma_1,\sigma_2)(I_m \otimes b), \\ &\vdots \\ \mathcal{G}_{\mathrm{B},k}(\sigma_1,\dots,\sigma_k)(I_{m^{k-1}} \otimes b) &= \widehat{\mathcal{G}}_{\mathrm{B},k}(\sigma_1,\dots,\sigma_k)(I_{m^{k-1}} \otimes b). \end{aligned}$$

(b) If W is constructed as

$$\begin{aligned} W_1 &= \mathcal{K}(\varsigma_\theta)^{-\mathsf{H}}\mathcal{C}(\varsigma_\theta)^{\mathsf{H}}c, \\ W_i &= \mathcal{K}(\varsigma_{\theta-i+1})^{-\mathsf{H}}\overline{\boldsymbol{\mathcal{N}}^{(2)}(\varsigma_{k-i+1})}(I_m \otimes W_{i-1}), \qquad 2 \le i \le \theta, \\ \operatorname{span}(W) &\supseteq \operatorname{span}\left(\begin{bmatrix} W_1 & \dots & W_\theta \end{bmatrix}\right), \end{aligned}$$

where $\boldsymbol{\mathcal{N}}^{(2)}$ is the 2-mode matricization of the tensor defined by $\boldsymbol{\mathcal{N}}^{(1)} = \mathcal{N}$, then the following interpolation conditions hold true:

$$\begin{aligned} c^{\mathsf{H}}\mathcal{G}_{\mathrm{B},1}(\varsigma_\theta) &= c^{\mathsf{H}}\widehat{\mathcal{G}}_{\mathrm{B},1}(\varsigma_\theta), \\ c^{\mathsf{H}}\mathcal{G}_{\mathrm{B},2}(\varsigma_{\theta-1},\varsigma_\theta) &= c^{\mathsf{H}}\widehat{\mathcal{G}}_{\mathrm{B},2}(\varsigma_{\theta-1},\varsigma_\theta), \\ &\vdots \\ c^{\mathsf{H}}\mathcal{G}_{\mathrm{B},\theta}(\varsigma_1,\dots,\varsigma_\theta) &= c^{\mathsf{H}}\widehat{\mathcal{G}}_{\mathrm{B},\theta}(\varsigma_1,\dots,\varsigma_\theta). \end{aligned}$$

(c) Let V be constructed as in Part (a) and W as in Part (b), then, additionally to the results in (a) and (b), the following conditions hold:

$$\begin{aligned} &c^{\mathsf{H}}\mathcal{G}_{\mathrm{B},q+\eta}(\sigma_1,\dots,\sigma_q,\varsigma_{\theta-\eta+1},\dots,\varsigma_\theta)(I_{m^{q+\eta-1}} \otimes b) \\ &= c^{\mathsf{H}}\widehat{\mathcal{G}}_{\mathrm{B},q+\eta}(\sigma_1,\dots,\sigma_q,\varsigma_{\theta-\eta+1},\dots,\varsigma_\theta)(I_{m^{q+\eta-1}} \otimes b), \end{aligned}$$

for $1 \le q \le k$ and $1 \le \eta \le \theta$. ◊

Corollary 5.22 (Bilinear blockwise tangential Hermite interpolation):
Let $\mathcal{G}_{\mathrm{B}}$ be a bilinear system, described by its subsystem transfer functions in (5.3), and $\widehat{\mathcal{G}}_{\mathrm{B}}$ the reduced-order bilinear system constructed by (5.4), with the corresponding subsystem transfer functions $\widehat{\mathcal{G}}_{\mathrm{B},k}$. Given sets of interpolation points $\sigma_1, \ldots, \sigma_k \in \mathbb{C}$ and $\varsigma_1, \ldots, \varsigma_\theta \in \mathbb{C}$, for which the matrix functions $\mathcal{C}$, $\mathcal{K}^{-1}$, $\mathcal{N}$, $\mathcal{B}$ and $\widehat{\mathcal{K}}^{-1}$ are complex differentiable, orders of partial derivatives $\ell_1, \ldots, \ell_k \in \mathbb{N}_0$ and $\nu_1, \ldots, \nu_\theta \in \mathbb{N}_0$, and two tangential directions $b \in \mathbb{C}^m$ and $c \in \mathbb{C}^p$, the following statements hold:

(a) If V is constructed as

$$
\begin{aligned}
V_{1,j_1} &= \partial_{s^{j_1}} \left(\mathcal{K}^{-1}\mathcal{B}\right)(\sigma_1)b, && j_1 = 0, \ldots, \ell_1,\\
V_{2,j_2} &= \partial_{s^{j_2}} \mathcal{K}^{-1}(\sigma_2)\partial_{s^{\ell_1}} \left(\mathcal{N}(I_m \otimes \mathcal{K}^{-1}\mathcal{B})\right)(\sigma_1)(I_m \otimes b), && j_2 = 0, \ldots, \ell_2,\\
&\vdots\\
V_{k,j_k} &= \partial_{s^{j_k}} \mathcal{K}^{-1}(\sigma_k) \left(\prod_{j=1}^{k-2} \partial_{s^{\ell_{k-j}}} \left((I_{m^{j-1}} \otimes \mathcal{N})(I_{m^j} \otimes \mathcal{K})\right)(\sigma_{k-j})\right)\\
&\quad \times \partial_{s^{\ell_1}} \left((I_{m^{k-2}} \otimes \mathcal{N})(I_{m^{k-1}} \otimes \mathcal{K}\mathcal{B})\right)(\sigma_1)(I_{m^{k-1}} \otimes b), && j_k = 0, \ldots, \ell_k,\\
\operatorname{span}(V) &\supseteq \operatorname{span}\left(\begin{bmatrix} V_{1,0} & \ldots & V_{k,\ell_k} \end{bmatrix}\right),
\end{aligned}
$$

then the following interpolation conditions hold true:

$$
\begin{aligned}
\partial_{s_1^{j_1}} \mathcal{G}_{\mathrm{B},1}(\sigma_1)b &= \partial_{s_1^{j_1}} \widehat{\mathcal{G}}_{\mathrm{B},1}(\sigma_1)b, && j_1 = 0, \ldots, \ell_1,\\
&\vdots\\
&\partial_{s_1^{\ell_1} \cdots s_{k-1}^{\ell_{k-1}} s_k^{j_k}} \mathcal{G}_{\mathrm{B},k}(\sigma_1, \ldots, \sigma_k)(I_{m^{k-1}} \otimes b)\\
&= \partial_{s_1^{\ell_1} \cdots s_{k-1}^{\ell_{k-1}} s_k^{j_k}} \widehat{\mathcal{G}}_{\mathrm{B},k}(\sigma_1, \ldots, \sigma_k)(I_{m^{k-1}} \otimes b), && j_k = 0, \ldots, \ell_k.
\end{aligned}
$$

(b) If W is constructed as

$$
\begin{aligned}
W_{1,i_\theta} &= \partial_{s^{i_\theta}} \left(\mathcal{K}^{-\mathsf{H}}\mathcal{C}^{\mathsf{H}}\right)(\varsigma_\theta)c, && i_\theta = 0, \ldots, \nu_\theta,\\
W_{2,i_{\theta-1}} &= \partial_{s^{i_{\theta-1}}} \left(\mathcal{K}^{-\mathsf{H}}\overline{\boldsymbol{\mathcal{N}}^{(2)}}\right)(\varsigma_{\theta-1})(I_m \otimes W_{1,\nu_\theta}), && i_{\theta-1} = 0, \ldots, \nu_{\theta-1},\\
&\vdots\\
W_{\theta,i_1} &= \partial_{s^{i_1}} \left(\mathcal{K}^{-\mathsf{H}}\overline{\boldsymbol{\mathcal{N}}^{(2)}}\right)(\varsigma_1)(I_m \otimes W_{\theta-1,\nu_2}), && i_1 = 0, \ldots, \nu_1,\\
\operatorname{span}(W) &\supseteq \operatorname{span}\left(\begin{bmatrix} W_{1,0} & \ldots & W_{\theta,\nu_\theta} \end{bmatrix}\right),
\end{aligned}
$$

where $\boldsymbol{\mathcal{N}}^{(2)}$ is the 2-mode matricization of the tensor defined by $\boldsymbol{\mathcal{N}}^{(1)} = \mathcal{N}$, then the following interpolation conditions hold true:

$$\begin{aligned} c^{\mathsf{H}} \partial_{s_1^{i_\theta}} \mathcal{G}_{\mathrm{B},1}(\varsigma_\theta) &= c^{\mathsf{H}} \partial_{s_1^{i_\theta}} \widehat{\mathcal{G}}_{\mathrm{B},1}(\varsigma_\theta), && i_\theta = 0, \ldots, \nu_\theta, \\ &\vdots \\ c^{\mathsf{H}} \partial_{s_1^{i_1} s_2^{\nu_2} \cdots s_\theta^{\nu_\theta}} \mathcal{G}_{\mathrm{B},\theta}(\varsigma_1, \ldots, \varsigma_\theta) &= c^{\mathsf{H}} \partial_{s_1^{i_1} s_2^{\nu_2} \cdots s_\theta^{\nu_\theta}} \widehat{\mathcal{G}}_{\mathrm{B},\theta}(\varsigma_1, \ldots, \varsigma_\theta), && i_1 = 0, \ldots, \nu_1. \end{aligned}$$

(c) Let V be constructed as in Part (a) and W as in Part (b), then, additionally to the interpolation conditions in (a) and (b), the following conditions hold:

$$\begin{aligned} &c^{\mathsf{H}} \partial_{s_1^{\ell_1} \cdots s_{q-1}^{\ell_{q-1}} s_q^{j_q} s_{q+1}^{i_{\theta-\eta+1}} s_{q+2}^{\nu_{\theta-\eta+2}} \cdots s_{q+\eta}^{\nu_\theta}} \mathcal{G}_{\mathrm{B},q+\eta}(\sigma_1, \ldots, \sigma_q, \varsigma_{\theta-\eta+1}, \ldots, \varsigma_\theta)(I_{m^{q+\eta-1}} \otimes b) \\ &= c^{\mathsf{H}} \partial_{s_1^{\ell_1} \cdots s_{q-1}^{\ell_{q-1}} s_q^{j_q} s_{q+1}^{i_{\theta-\eta+1}} s_{q+2}^{\nu_{\theta-\eta+2}} \cdots s_{q+\eta}^{\nu_\theta}} \widehat{\mathcal{G}}_{\mathrm{B},q+\eta}(\sigma_1, \ldots, \sigma_q, \varsigma_{\theta-\eta+1}, \ldots, \varsigma_\theta)(I_{m^{q+\eta-1}} \otimes b), \end{aligned}$$

for $j_q = 0, \ldots, \ell_q$; $i_{\theta-\eta+1} = 0, \ldots, \nu_{\theta-\eta+1}$; $1 \leq q \leq k$ and $1 \leq \eta \leq \theta$. ◊

Corollary 5.23 (Implicit bilinear blockwise tangential interpolation):
Let $\mathcal{G}_{\mathrm{B}}$ be a bilinear system, described by its subsystem transfer functions in (5.3), and $\widehat{\mathcal{G}}_{\mathrm{B}}$ the reduced-order bilinear system constructed by (5.4), with the corresponding subsystem transfer functions $\widehat{\mathcal{G}}_{\mathrm{B},k}$. Given a set of interpolation points $\sigma_1, \ldots, \sigma_k \in \mathbb{C}$, for which the matrix functions $\mathcal{C}$, $\mathcal{K}^{-1}$, $\mathcal{N}$, $\mathcal{B}$ and $\widehat{\mathcal{K}}^{-1}$ are complex differentiable, and two tangential directions $b \in \mathbb{C}^m$ and $c \in \mathbb{C}^p$, the following statements hold:

(a) Let V and W be constructed as in Corollary 5.21 Parts (a) and (b) for a matching sequence of interpolation points $\sigma_1 = \varsigma_1, \ldots, \sigma_k = \varsigma_k$, then additionally it holds

$$\nabla\Big(c^{\mathsf{H}} \mathcal{G}_{\mathrm{B},k}(I_{m^{k-1}} \otimes b)\Big)(\sigma_1, \ldots, \sigma_k) = \nabla\Big(c^{\mathsf{H}} \widehat{\mathcal{G}}_{\mathrm{B},k}(I_{m^{k-1}} \otimes b)\Big)(\sigma_1, \ldots, \sigma_k).$$

(b) Let V and W be constructed as in Corollary 5.22 Parts (a) and (b) for a matching sequence of interpolation points $\sigma_1 = \varsigma_1, \ldots, \sigma_k = \varsigma_k$ and matching orders of partial derivatives $\ell_1 = \nu_1, \ldots, \ell_k = \nu_k$, then additionally it holds

$$\begin{aligned} &\nabla\left(c^{\mathsf{H}} \partial_{s_1^{\ell_1} \cdots s_k^{\ell_k}} \mathcal{G}_{\mathrm{B},k}(I_{m^{k-1}} \otimes b)\right)(\sigma_1, \ldots, \sigma_k) \\ &= \nabla\left(c^{\mathsf{H}} \partial_{s_1^{\ell_1} \cdots s_k^{\ell_k}} \widehat{\mathcal{G}}_{\mathrm{B},k}(I_{m^{k-1}} \otimes b)\right)(\sigma_1, \ldots, \sigma_k). \end{aligned}$$ ◊

An important point in the motivation of tangential interpolation was the vast growth of the projection space dimensions in case of matrix interpolation. Now with the different ideas of tangential interpolation for bilinear systems, the question of the resulting projection space dimensions, and consequently the sizes of the constructed reduced-order models, arises again. The following remark gives an overview about the dimensions arising in the different presented approaches for structured interpolation of the k-th subsystem transfer function.

Remark 5.24 (Dimensions of projection spaces):
In practice, the subsystem transfer functions are not only interpolated in a single set of interpolation points, but in multiple sets. Therefore, let $n_\mathrm{s} \in \mathbb{N}$ be the number of sets of interpolation points and tangential directions to be used for interpolation. For the matrix interpolation approach from Theorem 5.9, the dimensions are given by

$$\dim\Big(\operatorname{span}(V_\mathrm{mtx})\Big) \leq n_\mathrm{s}\left(\sum_{j=1}^{k} m^k\right) \quad \text{and} \quad \dim\Big(\operatorname{span}(W_\mathrm{mtx})\Big) \leq n_\mathrm{s}\left(\sum_{j=1}^{k} pm^{k-1}\right), \tag{5.69}$$

for the right and left projection spaces, respectively. The blockwise tangential approach from Corollary 5.21 reduces these dimensions to

$$\dim\Big(\operatorname{span}(V_\mathrm{bwt})\Big) = \dim\Big(\operatorname{span}(W_\mathrm{bwt})\Big) \leq n_\mathrm{s}\left(\sum_{j=1}^{k} m^{k-1}\right). \tag{5.70}$$

Comparing (5.69) and (5.70) reveals the difference in the resulting dimensions using matrix or blockwise tangential interpolation to be the reduction by the factor m (or p in case of the left projection space). Still, the blockwise tangential interpolation leads to an exponential growth of the projection space dimension with respect to the subsystem transfer function level. In contrast, the new generalized tangential interpolation approach as in Theorem 5.18 reduces the dimensions significantly to

$$\dim\Big(\operatorname{span}(V_\mathrm{st})\Big) = \dim\Big(\operatorname{span}(W_\mathrm{st})\Big) \leq n_\mathrm{s} k.$$

These dimensions only grow linearly with the transfer function level k. Therefore, the tangential interpolation from Section 5.6.3 gives the most freedom in terms of choosing the reduced order as well as interpolation points and tangential directions. Also note that the new approach has an additional tuning opportunity with the scaling vectors, which enables the recovery of blockwise tangential interpolation conditions or even matrix interpolation if required. ◊

5.6.5 Numerical experiments

To test the obtained theoretical results in computations, the different structured tangential interpolation approaches are compared to the matrix interpolation from Section 5.4 in two numerical examples. The following interpolation methods will be compared:

MtxInt is the matrix interpolation approach from Section 5.4,

BwtInt denotes blockwise tangential interpolation from Section 5.6.4,

SftInt is tangential interpolation with the generalized framework from Section 5.6.3, choosing the scaling vectors as in the frequency domain motivation to be the all ones vector (Corollary 5.17 Part (a)),

SttInt is tangential interpolation with the generalized framework from Section 5.6.3, choosing the scaling vectors as in the time domain motivation to be identical to the right tangential directions (Corollary 5.17 Part (b)).

The same interpolation point selections (equi./$\mathcal{H}_\infty$/IRKA) and the averaged subspace approach (avg.) are used as in Section 5.3.3, where the tangential directions are either vectors with uniformly randomly generated entries from $[0, 1]$ or the $\mathcal{H}_2$-optimal tangential directions computed by the TF-IRKA approach in case of (IRKA). As in previous numerical experiments, the MORscore will be used for a general comparison. This is followed by a more detailed comparison of some selected methods for a chosen reduced order via pointwise relative errors. The same error measures as in Section 5.3.3 are used here. The different interpolation methods have also different restrictions concerning the computable reduced orders as mentioned in Remark 5.24, i.e., if a reduced-order model cannot be computed for a particular order the next smaller one is used instead. Note that the following experiments are intended to compare the general interpolation approaches rather than the selection strategies for interpolation points and tangential directions. Also, only the structured interpolation methods are used in the comparison due to the general lack of structure-preserving model reduction methods for bilinear systems and the results from Section 5.3.3 for unstructured alternatives.

5.6.5.1 MIMO bilineaer mass-spring-damper system

As first example, the bilinear mass-spring-damper system from previous experiments is reconsidered. In fact, the MIMO system from Section 5.5.4.1 is chosen with both parameters fixed to 1. As a reminder, this bilinear mechanical system takes the form

$$\begin{aligned} M\ddot{x}(t) + E\dot{x}(t) + Kx(t) &= N_{\mathrm{p},1}x(t)u_1(t) + N_{\mathrm{p},2}x(t)u_2(t) + B_{\mathrm{u}}u(t), \\ y(t) &= C_{\mathrm{p}}x(t), \end{aligned}$$

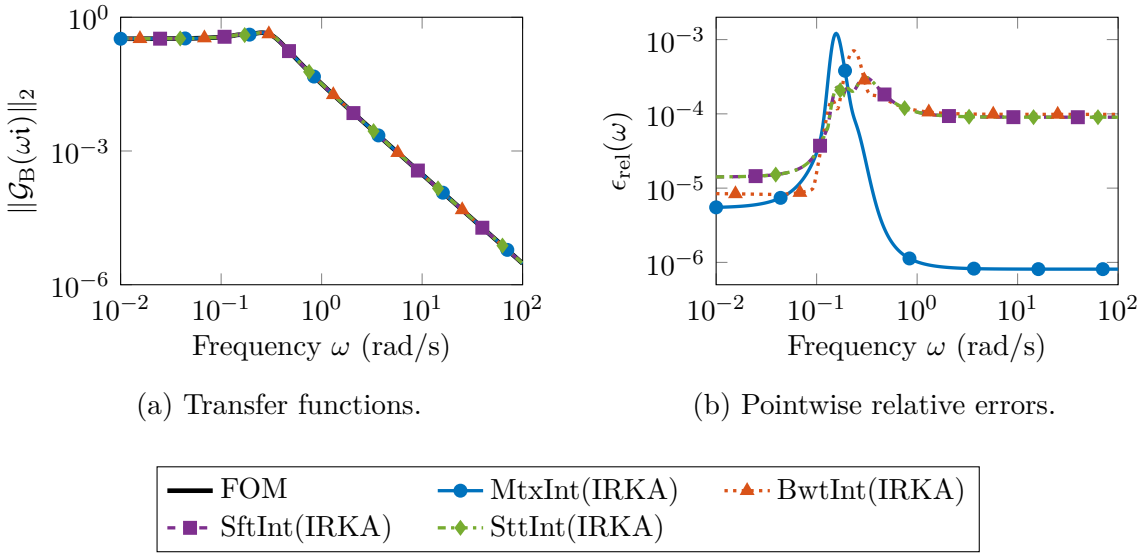
with $n_2 = 10\,000$ states, $m = 2$ inputs and $p = 2$ outputs. For model order reduction, only a one-sided projection is used with $W = V$ to preserve beside the internal system structure also symmetry and definiteness of the system matrices. Therefore, in the interpolation methods with (equi./$\mathcal{H}_\infty$/IRKA) points, only the right projection space $\operatorname{span}(V)$ is constructed. For the averaged subspaces, an oversampling via the different interpolation approaches is used to compute the left and right interpolatory projection spaces. An one-sided projection of appropriate size is constructed by bases concatenation and truncation using the pivoted QR decomposition.

The resulting MORscores for reduced orders from 1 to 48 are shown in Table 5.5. For the time domain simulations in the interval $[0, 100]\,\mathrm{s}$, the same Gaussian white

Table 5.5: MORscores for the MIMO bilinear mass-spring-damper example with reduction orders from 1 to 48.

Method	$\mathcal{H}_\infty^{(1)}$	$\mathcal{H}_\infty^{(2)}$	L_2	L_∞
MtxInt(equi.)	0.2025	0.1549	0.2380	0.2478
MtxInt($\mathcal{H}_\infty$)	0.1752	0.1389	0.1770	0.1917
MtxInt(IRKA)	0.2034	0.1566	0.2137	0.2256
MtxInt(avg.)	0.3139	0.2634	0.2972	0.3113
BwtInt(equi.)	0.2113	0.1632	0.2585	0.2681
BwtInt($\mathcal{H}_\infty$)	0.2015	0.1654	0.1839	0.1929
BwtInt(IRKA)	0.2320	0.1841	0.2218	0.2334
BwtInt(avg.)	0.2845	0.2467	0.2585	0.2718
SftInt(equi.)	0.2169	0.1659	0.2696	0.2767
SftInt($\mathcal{H}_\infty$)	0.2111	0.1708	0.1814	0.1938
SftInt(IRKA)	0.2423	0.1924	0.2376	0.2486
SftInt(avg.)	0.2975	0.2616	0.2600	0.2723
SttInt(equi.)	0.2188	0.1675	0.2751	0.2828
SttInt($\mathcal{H}_\infty$)	0.2111	0.1709	0.1848	0.1973
SttInt(IRKA)	0.2401	0.1903	0.2365	0.2467
SttInt(avg.)	0.3010	0.2583	0.2573	0.2693

noise-based input signal as in (5.44) is used. The general tendency of the MORscores revelas the tangential interpolation methods in the generalized framework, SftInt and SttInt, to work best for the different choices of interpolation points. Therefore, the new tangential framework seems to be a suitable alternative to matrix interpolation for the purpose of model reduction, which allows for a more accurate choice of the reduced order (Remark 5.24). However, this observation does not hold for the averaged subspaces, where MtxInt(avg.) performs best followed by SttInt(avg.) and SftInt(avg.). This behavior can be explained by the used oversampling procedure in which the matrix interpolation provided more useful information for the model reduction process than the tangential interpolations, which only consider the transfer function evaluations in certain directions. Still, the MORscores are more than close enough to each other to suggest the tangential interpolation-based averaging as suitable model reduction method, especially in cases with more inputs and outputs when the matrix interpolation easily leads to uncomputable large intermediate matrices and projection spaces. The performance of the blockwise tangential interpolation lies more or less in between the matrix and the new tangential interpolation methods.



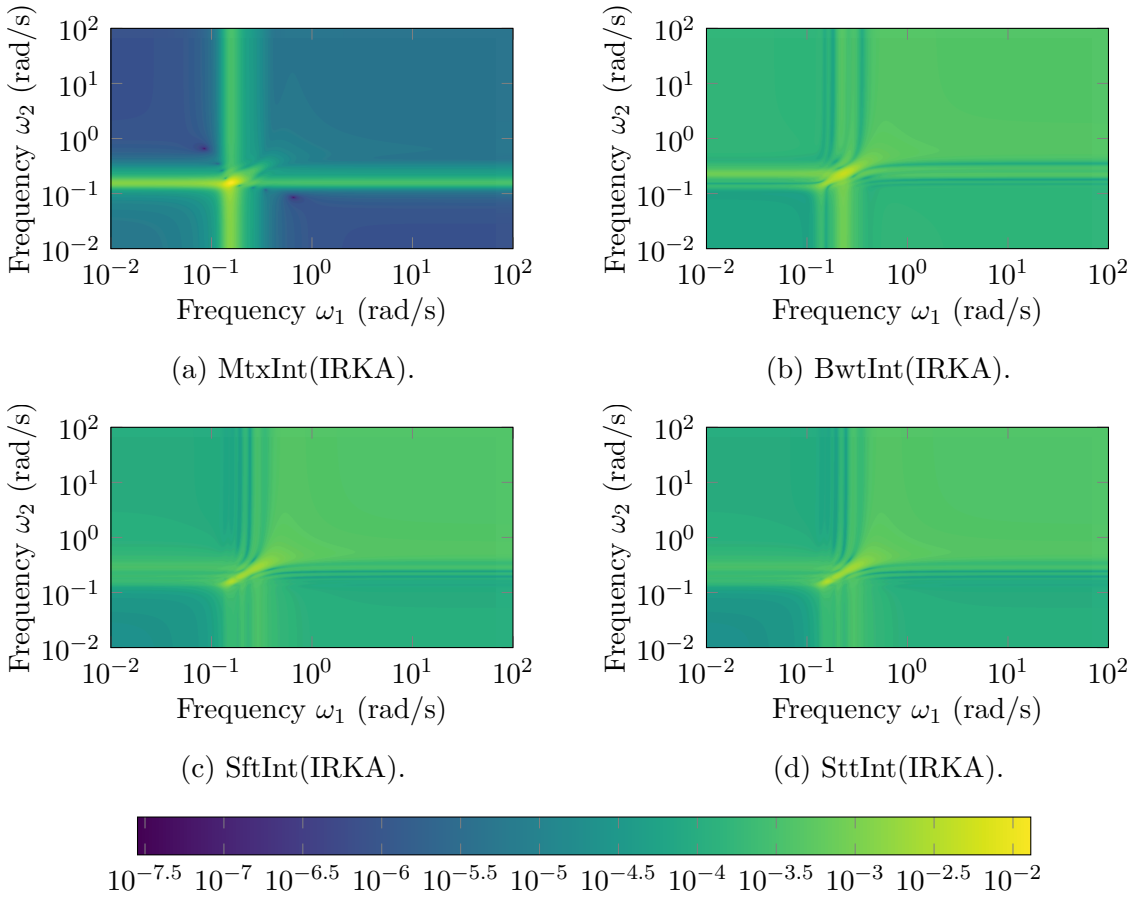
(a) Transfer functions.

(b) Pointwise relative errors.

Figure 5.10: First subsystem transfer functions and approximation errors for the MIMO bilinear mass-spring-damper example.

For a more detailed comparison, the reduced order $r_2 = 24$ is chosen such that all approaches can be used to compute a reduced-order model of exactly that order. For clarity, only the methods with IRKA interpolation points are selected in the comparison. The frequency domain results for the linear transfer functions are shown in Figure 5.10. The tangential approaches have a very similar error behavior, with the blockwise tangential interpolation having a slightly larger relative error in the middle of the frequency range. MtxInt starts in a similar order of magnitude with its relative error for low frequencies and overshoots the relative approximation error of the tangential techniques after 10^{-1} rad/s before it converges to the smallest relative error of all approaches for higher frequencies. Similar observations can be done for the second subsystem transfer functions in Figure 5.11. MtxInt shows a comparably small error for low frequencies and a lot smaller error in regions where only one of the transfer function arguments has high frequencies. In comparison, the tangential methods provide mainly larger relative errors with a more uniform error behavior.

Last, the time simulations of the full and reduced-order models are shown in Figure 5.12. The upper plot contains for clarity only the second output signal, while the pointwise relative errors are computed for the complete output vector and shown in the lower plot. MtxInt(IRKA) starts here with a relative error two orders of magnitude better than the tangential approaches, which increases to the same relative error level as the other methods at the end of the time interval. The tangential methods behave again very similar to each other, where SftInt(IRKA) and SttInt(IRKA) look exactly alike.



(a) MtxInt(IRKA). (b) BwtInt(IRKA).

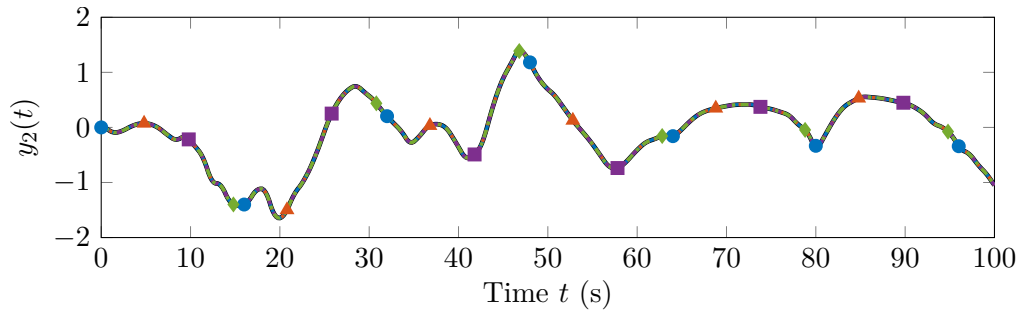
(c) SftInt(IRKA). (d) SttInt(IRKA).

Figure 5.11: Relative approximation errors $\epsilon_{\text{rel}}(\omega_1, \omega_2)$ of the second subsystem transfer functions for the MIMO bilinear mass-spring-damper example.

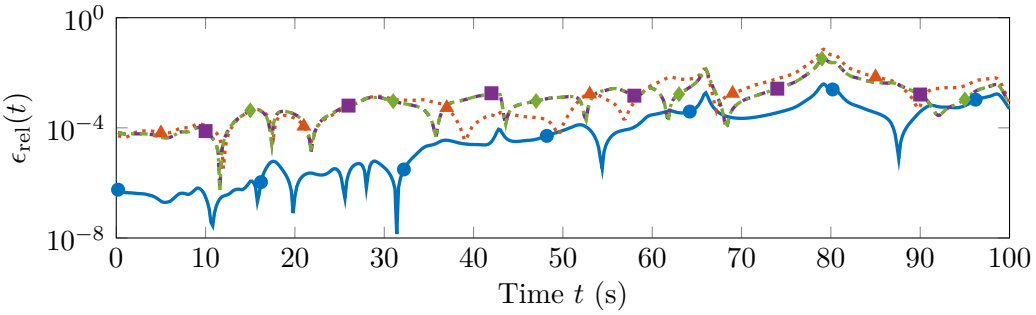
5.6.5.2 MIMO time-delayed heated rod

As second numerical example, the time-delayed heated rod from Section 5.3.3.2 is revisited. The system is extended to the MIMO case by modeling the control signal and measurements to act independently on equally sized sections of the rod. For the experiments, the rod is separated into three sections such that $m = p = 3$ holds. As in Section 5.3.3.2, the number of differential equations is set to be $n_1 = 5\,000$. For the model reduction, two-sided projections are computed for all interpolation approaches and the averaged subspaces.

Reduced-order models have been constructed for orders 1 to 48 and the resulting MORscores are shown in Table 5.6. For the time simulation, the interval $[0, 10]$ s was



(a) Second output entry $y_2(t)$ of the time simulation.



(b) Pointwise relative errors of the complete output.

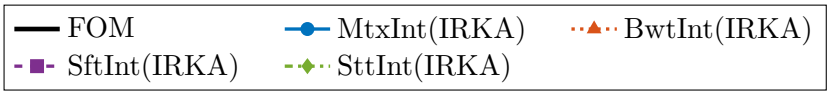


Figure 5.12: Time domain results for the MIMO bilinear mass-spring-damper example.

chosen with the following input signal

$$u(t) = 0.05 \cdot \begin{bmatrix} \eta_1(t_j) \\ \eta_2(t_j) \\ \eta_3(t_j) \end{bmatrix}, \quad \text{for } t_j \leq t < t_{j+1},$$

with $j = 0, \ldots, 9$, equidistant time steps $t_j = j \cdot \frac{10}{9}$, and presampled Gaussian white noise $\eta_1(t), \eta_2(t), \eta_3(t)$. As in the previous example, the MORscores reveal the tangential interpolation methods with (equi./$\mathcal{H}_\infty$/IRKA) to perform vastly better than the matrix interpolation, with a small exception for the averaged subspaces. The larger differences in the MORscores mainly result from the larger number of input and outputs of the system and the corresponding dimensions of the projection spaces, as mentioned in Remark 5.24. In consequence, only a few reduced-order models could be computed for MtxInt and BwtInt. Comparing the tangential interpolation methods, the results

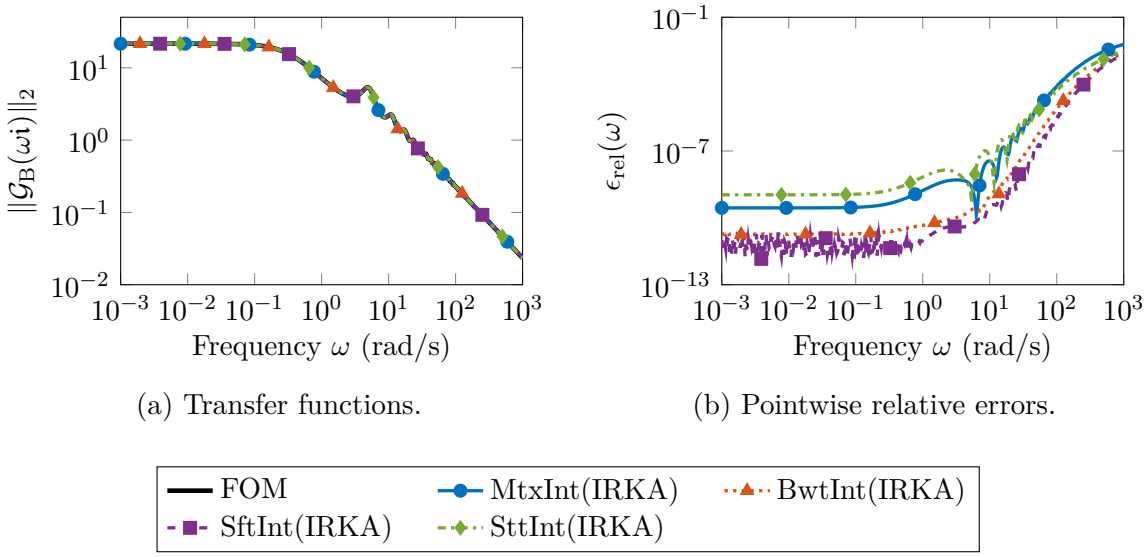
Table 5.6: MORscores for the MIMO time-delay example with reduced orders from 1 to 48.

Method	$\mathcal{H}_\infty^{(1)}$	$\mathcal{H}_\infty^{(2)}$	L_2	L_∞
MtxInt(equi.)	0.2694	0.2753	0.3547	0.3286
MtxInt($\mathcal{H}_\infty$)	0.2728	0.2692	0.3356	0.3154
MtxInt(IRKA)	0.2592	0.2776	0.3908	0.3811
MtxInt(avg.)	0.4044	0.3907	0.4510	0.4415
BwtInt(equi.)	0.2819	0.2871	0.4009	0.3988
BwtInt($\mathcal{H}_\infty$)	0.3260	0.3305	0.3985	0.3867
BwtInt(IRKA)	0.2925	0.2981	0.4070	0.4013
BwtInt(avg.)	0.3540	0.3576	0.4514	0.4417
SftInt(equi.)	0.3012	0.3009	0.4018	0.4042
SftInt($\mathcal{H}_\infty$)	0.3515	0.3395	0.3702	0.3724
SftInt(IRKA)	0.3281	0.3335	0.4274	0.4302
SftInt(avg.)	0.4230	0.3825	0.3849	0.3929
SttInt(equi.)	0.3010	0.3023	0.3982	0.3882
SttInt($\mathcal{H}_\infty$)	0.3395	0.3397	0.3966	0.3829
SttInt(IRKA)	0.3077	0.3114	0.4230	0.4186
SttInt(avg.)	0.3938	0.3650	0.4447	0.4386

are very mixed. For example, the blockwise tangential approach performs worse than the generalized tangential framework in frequency domain but has again a comparable or better MORscore in time domain. Looking at the averaged subspace methods, the tangential interpolation-based approaches are better or comparable to the matrix interpolation results, e.g., SftInt(avg.) is better than MtxInt(avg.) in the $\mathcal{H}_\infty^{(1)}$ measure and BwtInt(avg.) is better than MtxInt(avg.) in the L_2 error.

As reduced order for the detailed comparison, $n_1 = 24$ is chosen such that with all methods a reduced-order model of the appropriate size could be obtained. For the interpolation points, the TF-IRKA selection is used. Figures 5.13 and 5.14 show the results with pointwise relative errors in frequency domain for the first two subsystem transfer functions. Both figures show basically the same error behavior for the methods in frequency domain, with SttInt(IRKA) having the largest relative error for low frequencies followed by MtxInt(IRKA). Also, SftInt(IRKA) and BwtInt(IRKA) perform equally well, with SftInt(IRKA) having an overall slightly smaller error and BwtInt(IRKA) with a smoother error behavior.

The time domain results can be seen in Figure 5.15, with only the second output entry



(a) Transfer functions.

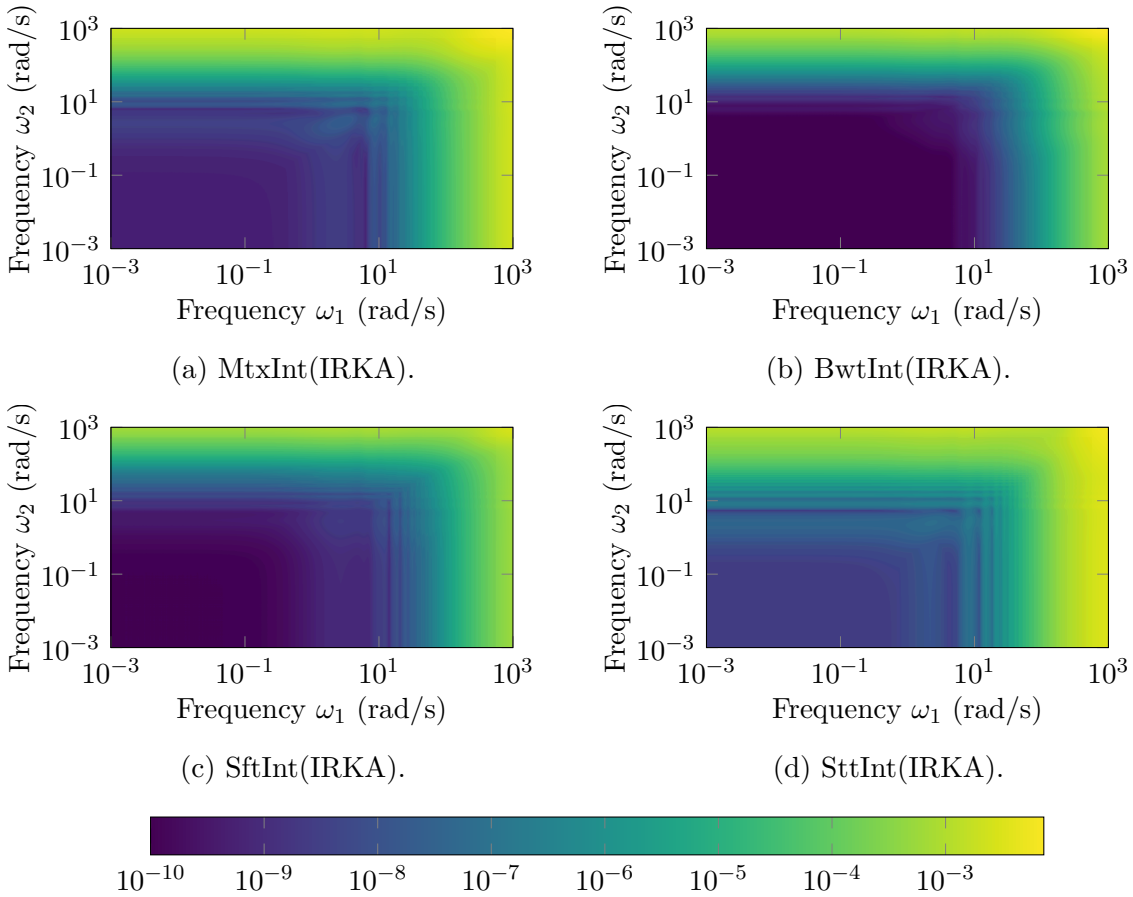
(b) Pointwise relative errors.

Figure 5.13: First subsystem transfer functions and approximation errors for the MIMO time-delay example.

in the upper plot for clarity and the pointwise relative errors of the full output vector in the lower one. Here, BwtInt(IRKA) shows the best apprxoimation behavior of all methods in terms of the error magnitude. Both, SttInt(IRKA) and MtxInt(IRKA) are comparable but with a less accurate approximation close to zero crossings resulting in a spiky intermediate behavior. On the other hand, SftInt(IRKA) provides a very smooth and constant relative error with only small disturbances in the zero crossings.

5.7 Conclusions

This chapter was concerned with the problem of structure-preserving model order reduction for bilinear control systems. Motivated by the structures arising from mechanical and time-delay systems, an extension of the structured transfer functions from [24] to bilinear systems was proposed. This uses matrix-valued functions to allow even more general structures than those used here for motivation and illustration. A new interpolation theory was developed for structured bilinear subsystem transfer functions to construct structure-preserving reduced-order models by projection that satisfy different types of interpolation conditions. A numerical comparison of the new structured interpolation framework with established model reduction methods producing unstructured systems revealed the new approach to be far more effective. While the general question of good or even optimal interpolation point selection is postponed to future work, the chosen selection strategies inspired by the linear system case performed very well. This theory



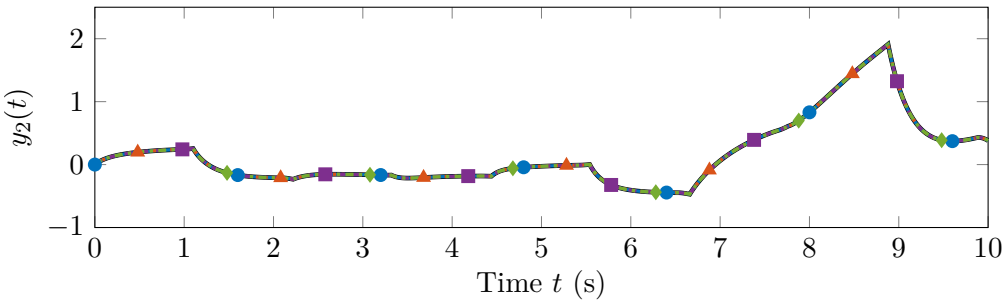
(a) MtxInt(IRKA). (b) BwtInt(IRKA).

(c) SftInt(IRKA). (d) SttInt(IRKA).

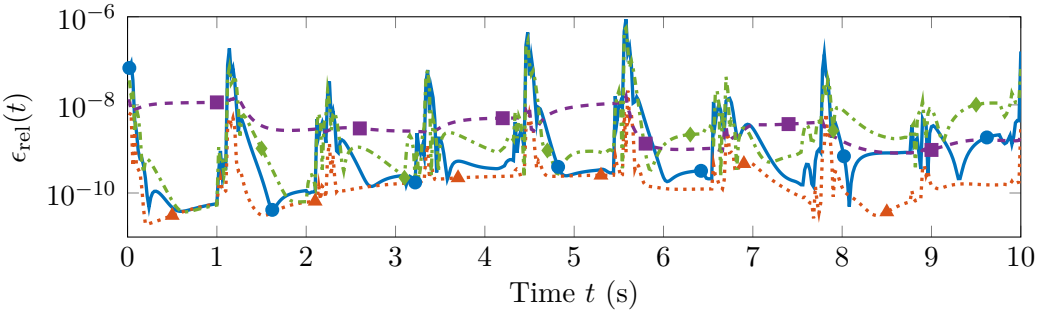
Figure 5.14: Relative approximation errors $\epsilon_{\text{rel}}(\omega_1, \omega_2)$ of the second subsystem transfer functions for the MIMO time-delay example.

got then extended to the case of parametric bilinear systems including results on implicit interpolation of parameter sensitivities. Two quick numerical examples showed the good approximation quality for the structured parametric approach, where further comparisons were omitted due to the results of the previous experiments for SISO systems.

A new unifying framework for tangential interpolation of structured bilinear systems was proposed. The new framework was motivated in time and frequency domains but proven in a far more general setting that gives a lot of freedom for realizations of model reduction methods. It was also used to generate structured results for the blockwise tangential interpolation method, known in the literature for unstructured bilinear systems [31, 160]. All different tangential interpolation approaches were tested and compared in numerical experiments to the structured matrix interpolation method. As result, the new tangential interpolation framework turned out to be an efficient alternative approach especially in the cases when matrix interpolation would result in



(a) Second output entry $y_2(t)$ of the time simulation.



(b) Pointwise relative errors of the complete output.

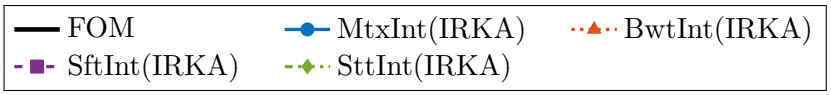


Figure 5.15: Time domain results for the MIMO time-delay example.

very large reduced-order models due to a large number of system inputs and outputs.

CHAPTER 6

STRUCTURED NONLINEAR SYSTEMS

Contents

6.1 Introduction

After studying the cases of mechanical linear and structured bilinear control systems, at last, the general case of structured nonlinear systems is considered. A particular and the most relevant case for this thesis is nonlinear control-affine mechanical systems of the

form

$$\begin{aligned} M\ddot{x}(t) + f(x(t), \dot{x}(t)) &= B_{\mathrm{u}} u(t), \\ y(t) &= C_{\mathrm{p}} x(t) + C_{\mathrm{v}} \dot{x}(t), \end{aligned} \tag{6.1}$$

with the nonlinear description of the state evolution $f\colon \mathbb{R}^{n_2} \times \mathbb{R}^{n_2} \to \mathbb{R}^{n_2}$. The motivational example of the Toda lattice model from Section 1.3.3 belongs to this system class. Model reduction methods for the general nonlinear mechanical system (6.1) or the classical unstructured nonlinear system case often involve time simulations to gain information about the underlying system dynamics. This is done for example in proper orthogonal decomposition (POD); see e.g., [71, 114, 128, 165, 186]; or in the empirical Gramian framework; see, e.g., [110, 112, 131, 133]. A certain problem is the treatment of the nonlinearity f in (6.1) as it acts only on the full system state. This becomes costly when working with a reduced-order approximation of the state $x \approx V\hat{x}$ since in each step, the original state x needs to be recovered to evaluate $f(x, \dot{x}) \approx f(V\hat{x}, V\dot{\hat{x}})$. Therefore, also the nonlinear state evolution needs to be approximated, which leads to the use of hyper-reduction methods to reduce the computational costs of evaluating the nonlinear function f, for example, by the (discrete) empirical interpolation method ((D)EIM) [20, 71, 77]. Overall, the resulting model reduction processes come with several disadvantages such as the dependence on time simulations, involving the choice of input signals, integrators and corresponding parameters, and the additional layer of approximation introduced by the hyper-reduction step, which especially needs a certain influence on the implementation of the nonlinearity.

A different way of handling (6.1) in model order reduction, which avoids simulations and the hyper-reduction, amounts from the reformulation of the general nonlinearities in the system. For smooth enough f, nonlinear systems like (6.1) can be rewritten into quadratic-bilinear systems (Section 2.3.2) to give the nonlinear term an easier manageable structure for the model reduction process. In the sense of mechanical systems, (6.1) would be rewritten into

$$\begin{aligned} 0 = {} & M\ddot{x}(t) + E\dot{x}(t) + Kx(t) \\ & + H_{\mathrm{vv}}\big(\dot{x}(t) \otimes \dot{x}(t)\big) + H_{\mathrm{vp}}\big(\dot{x}(t) \otimes x(t)\big) \\ & + H_{\mathrm{pv}}\big(x(t) \otimes \dot{x}(t)\big) + H_{\mathrm{pp}}\big(x(t) \otimes x(t)\big) \\ & - \sum_{j=1}^{m} N_{\mathrm{p},j} x(t) u(t) - \sum_{j=1}^{m} N_{\mathrm{v},j} \dot{x}(t) u(t) - B_{\mathrm{u}} u(t), \\ y(t) = {} & C_{\mathrm{p}} x(t) + C_{\mathrm{v}} \dot{x}(t), \end{aligned} \tag{6.2}$$

with $M, E, K \in \mathbb{R}^{n_2 \times n_2}$, $B_{\mathrm{u}} \in \mathbb{R}^{n_2 \times m}$ and $C_{\mathrm{p}}, C_{\mathrm{v}} \in \mathbb{R}^{p \times n_2}$, the bilinear terms $N_{\mathrm{p},j}, N_{\mathrm{v},j} \in \mathbb{R}^{n_2 \times n_2}$, for $j = 1, \ldots, m$, and the quadratic terms $H_{\mathrm{vv}}, H_{\mathrm{vp}}, H_{\mathrm{pv}}, H_{\mathrm{pp}} \in \mathbb{R}^{n_2 \times n_2^2}$. The class of (unstructured) first-order quadratic-bilinear systems (2.35) has recently received

a lot of attention in model order reduction as suitable alternative to general nonlinear systems. Methods developed for the reduction of (2.35) are, for example, subsystem transfer function interpolation [1, 3, 4, 13, 30, 101], an IRKA-like method for $\mathcal{H}_2$-(quasi)-optimal reduced-order models [40], balanced truncation [36, 38], or a Loewner approach to generate (2.35) from frequency domain data [92]. However, these methods are restricted to systems of the form (2.35) and do not consider any internal structures arising from physical phenomena. While mechanical quadratic-bilinear systems of the form (6.2) could be rewritten in first-order form (2.35), this would lead to unstructured reduced-order models without physical interpretation. These have already been shown to yield several disadvantages in the bilinear system case; see Section 5.3.3. Also, other structures such as internal time delays could occur, which cannot be rewritten into unstructured first-order form and prevents the use of the intrusive model reduction methods from above. In this case, it would be still possible to generate (2.35) using the Loewner framework [92]. But as demonstrated in the numerical experiments for the bilinear system case, see Section 5.3.3.2, to omit the internal structure easily yields unsatisfactory results.

This chapter is concerned with the problem of structure-preserving model reduction for nonlinear systems via interpolation of structured quadratic-bilinear systems. It begins with the process of rewriting structured nonlinear systems into quadratic-bilinear ones, in Section 6.2, using the example of the Toda lattice model from Section 1.3.3 for illustration. Afterwards, the different concepts of subsystem transfer functions for quadratic-bilinear systems are extended to the structured MIMO system case in Section 6.3. In Section 6.4, interpolation theory for these different structured transfer functions of quadratic-bilinear systems is developed. Finally, the theoretic results are tested in numerical experiments in Section 6.5.

6.2 Quadratic-bilinearization of nonlinear systems

Roughly speaking, every nonlinear system like (6.1) with smooth enough nonlinearities can be re-written into a quadratic-bilinear system like (6.2). This re-modeling procedure is known as *quadratic-bilinearization* [63, 95, 101]. It can be summarized into three basic steps:

1. Introduce appropriate replacement variables for the nonlinear terms.
2. Replace the nonlinear terms in the differential equations by the new variables.
3. Derive differential equations or algebraic constraints to describe the new replacement variables.

Quadratic-bilinearization has the big advantage of transforming difficult nonlinear terms into easier manageable quadratic form allowing the application of model reduction methods and often eases the general use of nonlinear systems, e.g., in time-domain

simulations. On the other hand, it comes with the cost of increasing the number of state variables and differential(-algebraic) equations in the system, as well as dealing with a quadratic tensor as new nonlinearity. The quadratic-bilinearization approach has also been known in the literature for quite some time as McCormick relaxation [141]. Since it is counterintuitive to increase the order of the system in contrast to the process of model order reduction that aims for the reduction of the order, it is not surprising that this approach was only recently re-considered [63, 95, 101]. However, this process can quickly payoff as it allows the application of sophisticated model reduction techniques for the quadratic-bilinear formulations. It should be mentioned that the quadratic-bilinearization of a nonlinear system is neither unique, i.e., there might be different choices of the replacement variables leading to different systems, nor automatable so far.

The following two subsections re-consider the initial example of the nonlinear Toda lattice model from Section 1.3.3 for quadratic-bilinearization. Two versions of the model are constructed, first, with quadratic-bilinear differential-algebraic equations (QBDAEs) and, thereafter, with quadratic-bilinear ordinary differential equations (QBODEs).

6.2.1 Toda lattice model as QBDAE system

The first step of quadratic-bilinearization is the definition of appropriate new variables for the nonlinear terms in (1.4). These are of a repetitive shape involving exponential functions. Therefore, consider the following substitution to linearize the equations in (1.4) by

$$z_j(t) := \begin{cases} e^{k_j(x_j(t)-x_{j+1}(t))} - 1, & \text{if } j < n_2, \\ e^{k_n x_{n_2}(t)} - 1, & \text{if } j = n_2, \end{cases} \tag{6.3}$$

for $j = 1, \ldots, n_2$. This specific choice of substitution is not the first intuitive thing to do since additionally to the nonlinear terms, also the subtraction of 1 is present. The important advantage of using (6.3) becomes clear when thinking about the initial conditions of the system. When inserting the original initial state $x(0) = 0$ into (6.3), one can observe that also $z(0) = 0$ holds, i.e., additional problems arising from non-zero initial conditions in the later model reduction process are avoided. Inserting the substitution (6.3) into (1.4) yields a new system of differential equations of the form

$$\begin{aligned} m_1\ddot{x}_1(t) + \gamma_1\dot{x}_1(t) + z_1(t) &= g_1(t), \\ m_j\ddot{x}_j(t) + \gamma_j\dot{x}_j(t) + z_j(t) - z_{j-1}(t) &= g_j(t), \quad \text{for } 1 < j \leq n_2. \end{aligned} \tag{6.4}$$

The new equations in (6.4) are linear in the state variables. As the last step, n_2 further equations are needed to describe the evolution of the introduced replacement variables z_j. Therefore, the definitions of z_j in (6.3) need to be differentiated with respect to time. This allows rewriting (6.3) in terms of the existing state variables x_j and z_j. The time

derivative of (6.3) yields

$$\dot{z}_j(t) = \begin{cases} k_j\big(\dot{x}_j(t) - \dot{x}_{j+1}(t)\big)\big(z_j(t)+1\big), & \text{if } j < n_2, \\ k_n\dot{x}_{n_2}(t)\big(z_{n_2}(t)+1\big), & \text{if } j = n_2, \end{cases} \tag{6.5}$$

for $j = 1, \ldots, n_2$. Combining (6.4) and (6.5) results in a system of QBDAEs of the form

$$\begin{aligned} \widetilde{M}\ddot{\tilde{x}}(t) + \widetilde{E}\dot{\tilde{x}}(t) + \widetilde{K}\tilde{x}(t) + \widetilde{H}_{\text{pv}}\big(\tilde{x}(t) \otimes \dot{\tilde{x}}(t)\big) &= \tilde{g}(t), \\ y(t) &= \widetilde{C}_{\text{v}}\dot{\tilde{x}}(t), \end{aligned} \tag{6.6}$$

with the concatenated state $\tilde{x}(t)^{\mathsf{T}} = \begin{bmatrix} x(t)^{\mathsf{T}} & z(t)^{\mathsf{T}} \end{bmatrix}$ and initial conditions $\tilde{x}(0) = \dot{\tilde{x}}(0) = 0$. The system matrices of (6.6) are given by

$$\widetilde{M} = \begin{bmatrix} M & 0 \\ 0 & 0 \end{bmatrix}, \quad \widetilde{E} = \begin{bmatrix} E & 0 \\ E_{21} & I_{n_2} \end{bmatrix}, \quad \widetilde{K} = \begin{bmatrix} 0 & K_{12} \\ 0 & 0 \end{bmatrix}, \quad \widetilde{H}_{\text{pv}} = \begin{bmatrix} 0 & 0 \\ 0 & H_{\text{pv}}^{(2,2)} \end{bmatrix}$$
$$\tilde{g}(t) = \begin{bmatrix} g(t) \\ 0 \end{bmatrix}, \quad \widetilde{C}_{\text{v}} = \begin{bmatrix} C_{\text{v}} & 0 \end{bmatrix}.$$

Therein, the single matrix blocks are the original system quantities M, E, g, C_{v} from (1.4) and (1.5), and new matrices resulting from the quadratic-bilinearization, with

$$E_{21} = \begin{bmatrix} -k_1 & k_1 & & \\ & \ddots & \ddots & \\ & & -k_{n-1} & k_{n-1} \\ & & & -k_n \end{bmatrix}, \quad K_{12} = \begin{bmatrix} 1 & & & \\ -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{bmatrix},$$

and the quadratic term such that

$$H_{\text{pv}}^{(2,2)}\left(z(t) \otimes \begin{bmatrix} \dot{x}(t) \\ \dot{z}(t) \end{bmatrix}\right) = \begin{bmatrix} -k_1 z_1(t)\dot{x}_1(t) + k_1 z_1(t)\dot{x}_2(t) \\ \vdots \\ -k_{n_2-1} z_{n_2-1}(t)\dot{x}_{n_2-1}(t) + k_{n_2-1} z_{n_2-1}(t)\dot{x}_{n_2}(t) \\ -k_{n_2} z_{n_2}(t)\dot{x}_{n_2}(t) \end{bmatrix}.$$

6.2.2 Toda lattice model as QBODE system

The QBDAE system (6.6) has its advantages in the reasonably easy block structure of the system matrices and with only a single out of four possible quadratic terms. However, it comes with usual difficulties arising from DAEs that need to be handled, for example, in time simulations. But the process of quadratic-bilinearization can be continued to

transform (6.6) into a system of QBODEs. Therefore, consider the second-order time derivatives of the substitution variables (6.3) such that

$$\ddot{z}_j(t) = \begin{cases} k_j\big(\ddot{x}_j(t) - \ddot{x}_{j+1}(t)\big)\big(z_j(t)+1\big) + k_j\big(\dot{x}_j(t) - \dot{x}_{j+1}(t)\big)\dot{z}_j(t), & \text{if } j < n_2, \\ k_{n_2}\ddot{x}_{n_2}(t)\big(z_{n_2}(t)+1\big) + k_{n_2}\dot{x}_{n_2}(t)\dot{z}_{n_2}(t), & \text{if } j = n_2. \end{cases} \tag{6.7}$$

Rearranging (6.4) in terms of $\ddot{x}_j$ and inserting into (6.7) yields the following QBODE system

$$\begin{aligned} 0 &= \widetilde{M}\ddot{\tilde{x}}(t) + \widetilde{E}\dot{\tilde{x}}(t) + \widetilde{K}\tilde{x}(t) - \tilde{g}(t) - \widetilde{N}_{\mathrm{p}}\big(\tilde{g}(t)\big)\tilde{x}(t) \\ &\quad + \widetilde{H}_{\mathrm{pp}}\big(\tilde{x}(t)\otimes\tilde{x}(t)\big) + \widetilde{H}_{\mathrm{pv}}\big(\tilde{x}(t)\otimes\dot{\tilde{x}}(t)\big) + \widetilde{H}_{\mathrm{vv}}\big(\dot{\tilde{x}}(t)\otimes\dot{\tilde{x}}(t)\big), \\ y(t) &= \widetilde{C}_{\mathrm{v}}\dot{\tilde{x}}(t), \end{aligned} \tag{6.8}$$

with the concatenated state $\tilde{x}(t)^{\mathsf{T}} = \begin{bmatrix} x(t)^{\mathsf{T}} & z(t)^{\mathsf{T}} \end{bmatrix}$ and initial conditions $\tilde{x}(0) = \dot{\tilde{x}}(0) = 0$. The system matrices of (6.8) are given by

$$\widetilde{M} = \begin{bmatrix} M & 0 \\ 0 & M_{22} \end{bmatrix}, \quad \widetilde{E} = \begin{bmatrix} E & 0 \\ E_{21} & 0 \end{bmatrix}, \quad \widetilde{K} = \begin{bmatrix} 0 & K_{12} \\ 0 & K_{22} \end{bmatrix}, \quad \widetilde{C}_{\mathrm{v}} = \begin{bmatrix} C_{\mathrm{v}} & 0 \end{bmatrix},$$
$$\widetilde{H}_{\mathrm{pp}} = \begin{bmatrix} 0 & 0 \\ 0 & H_{\mathrm{pp}}^{(2,2)} \end{bmatrix}, \quad \widetilde{H}_{\mathrm{pv}} = \begin{bmatrix} 0 & 0 \\ 0 & H_{\mathrm{pv}}^{(2,2)} \end{bmatrix}, \quad \widetilde{H}_{\mathrm{vv}} = \begin{bmatrix} 0 & 0 \\ 0 & H_{\mathrm{vv}}^{(2,2)} \end{bmatrix},$$

with the original quantities M, E, C_{v} from (1.4), and additionally

$$M_{22} = \begin{bmatrix} \frac{m_1 m_2}{k_1} & & & \\ & \ddots & & \\ & & \frac{m_{n_2-1}m_{n_2}}{k_{n_2-1}} & \\ & & & \frac{m_{n_2}}{k_{n_2}} \end{bmatrix},$$

$$E_{21} = \begin{bmatrix} m_2\gamma_1 & -m_1\gamma_2 & & \\ & \ddots & \ddots & \\ & & m_{n_2}\gamma_{n_2-1} & -m_{n_2-1}\gamma_{n_2} \\ & & & \gamma_{n_2} \end{bmatrix},$$

$$K_{12} = \begin{bmatrix} 1 & & & \\ -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{bmatrix},$$

$$K_{22} = \begin{bmatrix} m_1+m_2 & -m_1 & & & \\ -m_3 & m_2+m_3 & -m_2 & & \\ & \ddots & \ddots & \ddots & \\ & & -m_{n_2} & m_{n_2-1}+m_{n_2} & -m_{n_2-1} \\ & & & -1 & 1 \end{bmatrix}.$$

The quadratic terms are given by

$$H_{\mathrm{pp}}^{(2,2)}\left(z(t)\otimes\begin{bmatrix}x(t)\\ z(t)\end{bmatrix}\right)=\begin{bmatrix}(m_1+m_2)z_1(t)^2-m_1z_1(t)z_2(t)\\ -m_3z_2(t)z_1(t)+(m_2+m_3)z_2(t)^2-m_2z_2(t)z_3(t)\\ \vdots\\ -m_{n_2}z_{n_2-1}(t)z_{n_2-2}(t)+(m_{n_2-1}+m_{n_2})z_{n_2}(t)^2-m_{n_2-1}z_{n_2-1}(t)z_{n_2}(t)\\ -z_{n_2}(t)z_{n_2-1}(t)+z_{n_2}(t)^2\end{bmatrix},$$

$$H_{\mathrm{pv}}^{(2,2)}\left(z(t)\otimes\begin{bmatrix}\dot{x}(t)\\ \dot{z}(t)\end{bmatrix}\right)=\begin{bmatrix}m_2\gamma_1z_1(t)\dot{x}_1(t)-m_1\gamma_2z_1(t)\dot{x}_2(t)\\ \vdots\\ m_{n_2}\gamma_{n_2-1}z_{n_2-1}(t)\dot{x}_{n_2-1}(t)-m_{n_2-1}\gamma_nz_{n_2-1}(t)\dot{x}_{n_2}(t)\\ \gamma_{n_2}z_{n_2}(t)\dot{x}_{n_2}(t)\end{bmatrix},$$

$$H_{\mathrm{vv}}^{(2,2)}\left(\dot{z}(t)\otimes\begin{bmatrix}\dot{x}(t)\\ \dot{z}(t)\end{bmatrix}\right)=\begin{bmatrix}-m_1m_2\dot{z}_1(t)\dot{x}_1(t)+m_1m_2\dot{z}_1(t)\dot{x}_2(t)\\ \vdots\\ -m_{n_2-1}m_{n_2}\dot{z}_{n_2-1}(t)\dot{x}_{n_2-1}(t)+m_{n_2-1}m_{n_2}\dot{z}_{n_2-1}(t)\dot{x}_{n_2}(t)\\ -m_{n_2}\dot{z}_{n_2}(t)\dot{x}_{n_2}(t)\end{bmatrix}.$$

The two terms left to explain involve the right-hand side $g(t)$ and need adjustments according to the final representation of the external forcing with the input signal $u(t)$. In general, the new right-hand side and the bilinear term are given by

$$\tilde{g}(t)=\begin{bmatrix}g(t)\\ m_2g_1(t)-m_1g_2(t)\\ \vdots\\ m_{n_2}g_{n_2-1}(t)-m_{n_2-1}g_{n_2}(t)\\ g_{n_2}(t)\end{bmatrix},\qquad\text{and}$$

$$\widetilde{N}_{\mathrm{p}}(\tilde{g}(t))\tilde{x}(t)=\begin{bmatrix}0\\ \vdots\\ 0\\ \Big(m_2g_1(t)-m_1g_2(t)\Big)z_1(t)\\ \vdots\\ \Big(m_{n_2}g_{n_2-1}(t)-m_{n_2-1}g_{n_2}(t)\Big)z_{n_2-1}(t)\\ g_{n_2}(t)z_{n_2}\end{bmatrix}.$$

Replacing now the right-hand side by the product of a matrix with an input signal such that $g(t)=B_{\mathrm{u}}u(t)$, the system (6.8) can be formulated in the mechanical quadratic-bilinear form (6.2). Let $b_{i,j}$ be the (i,j)-th entry of the matrix $B_{\mathrm{u}}\in\mathbb{R}^{n_2\times m}$ and let $b_{i,*}$

denote the i-th row of B_u. Then, the new right-hand side can be written as

$$\tilde{g}(t) = \widetilde{B}_\mathrm{u} u(t) = \begin{bmatrix} B_\mathrm{u} \\ m_2 b_{1,*} - m_1 b_{2,*} \\ \vdots \\ m_{n_2} b_{n_2-1,*} - m_{n_2-1} b_{n_2,*} \\ b_{n_2,*} \end{bmatrix} u(t),$$

and the bilinear component becomes $\widetilde{N}_\mathrm{p}\big(\tilde{g}(t)\big)\tilde{x}(t) = \sum_{j=1}^{m} \widetilde{N}_{\mathrm{p},j} \tilde{x}(t) u_j(t)$, with

$$\widetilde{N}_{\mathrm{p},j} = \begin{bmatrix} 0 & & & 0 & \\ & m_2 b_{1,j} - m_1 b_{2,j} & & & \\ 0 & & \ddots & & \\ & & & m_n b_{n_2-1,j} - m_{n_2-1} b_{n_2,j} & \\ & & & & b_{n_2,j} \end{bmatrix},$$

where 0 abbreviates here the zero matrix of dimensions $n_2 \times n_2$.

It can easily be seen that the QBODE system (6.8) is a lot more complex than the QBDAE version (6.6) due to the additional quadratic and bilinear terms involved in the formulation. This cost needs to be paid for removing the differential-algebraic constraints from the QBDAE system. In practice, it is important to outweigh the advantages and disadvantages of the different possible formulations.

6.3 Towards structured quadratic-bilinear systems

As in cases of linear and bilinear systems, for structure-preserving model reduction via interpolation, some concept of structured transfer functions is needed first. In case of quadratic-bilinear systems, different types of transfer functions are used in the literature to represent the systems in the frequency domain; see Section 2.3.2. These types will be extended in this section to the structured setting. Therefore, the initial example of mechanical quadratic-bilinear systems (6.2) will serve as ongoing motivation. Similar to Section 5.2.3, a first-order realization of (6.2) will be used in the known transfer function formulations to develop the structured representations. Using the augmented state vector $\mathsf{x}(t)^\mathsf{T} = \begin{bmatrix} x(t)^\mathsf{T} & \dot{x}(t)^\mathsf{T} \end{bmatrix}$, the second-order system (6.2) can be rewritten in first-order form (2.35) by using the following block matrices

$$\begin{aligned} \mathsf{E} &= \begin{bmatrix} I_{n_2} & 0 \\ 0 & M \end{bmatrix}, \quad \mathsf{A} = \begin{bmatrix} 0 & I_{n_2} \\ -K & -E \end{bmatrix}, \quad \mathsf{B} = \begin{bmatrix} 0 \\ B_\mathrm{u} \end{bmatrix}, \\ \mathsf{C} &= \begin{bmatrix} C_\mathrm{p} & C_\mathrm{v} \end{bmatrix}, \quad \mathsf{N}_j = \begin{bmatrix} 0 & 0 \\ N_{\mathrm{p},j} & N_{\mathrm{v},j} \end{bmatrix}, \end{aligned} \tag{6.9}$$

for $j = 1, \ldots, m$, and the quadratic term

$$\mathsf{H} = -\begin{bmatrix} 0 & 0 & \ldots & 0 & 0 & 0 & 0 & \ldots & 0 & 0 \\ H_{\mathrm{pp},1} & H_{\mathrm{pv},1} & \ldots & H_{\mathrm{pp},n_2} & H_{\mathrm{pv},n_2} & H_{\mathrm{vp},1} & H_{\mathrm{vv},1} & \ldots & H_{\mathrm{vp},n_2} & H_{\mathrm{vv},n_2} \end{bmatrix}. \tag{6.10}$$

For (6.10), the matrices of the quadratic terms in (6.2) are sliced into $n_2 \times n_2$ pieces such that, for example,

$$H_{\mathrm{pp}} = \begin{bmatrix} H_{\mathrm{pp},1} & H_{\mathrm{pp},2} & \ldots & H_{\mathrm{pp},n_2} \end{bmatrix}$$

is used, with $H_{\mathrm{pp},j} \in \mathbb{R}^{n_2 \times n_2}$ for all $j = 1, \ldots, n_2$. The complicated expression in (6.10) mixes the sliced matrices to fit with the Kronecker product of the augmented state vector from the first-order system case. This can be simplified using an appropriate permutation of the quadratic term as well as of the Kronecker product with the states. In fact, one can easily show that

$$\mathsf{H}\big(\mathsf{x}(t) \otimes \mathsf{x}(t)\big) = -\begin{bmatrix} 0 & 0 & 0 & 0 \\ H_{\mathrm{pp}} & H_{\mathrm{pv}} & H_{\mathrm{vp}} & H_{\mathrm{vv}} \end{bmatrix} \begin{bmatrix} x(t) \otimes x(t) \\ x(t) \otimes \dot{x}(t) \\ \dot{x}(t) \otimes x(t) \\ \dot{x}(t) \otimes \dot{x}(t) \end{bmatrix} \tag{6.11}$$

holds. Also, the bilinear terms of the second-order system (6.2) are concatenated concisely such that

$$N_{\mathrm{p}} := \begin{bmatrix} N_{\mathrm{p},1} & \ldots & N_{\mathrm{p},m} \end{bmatrix} \quad \text{and} \quad N_{\mathrm{v}} := \begin{bmatrix} N_{\mathrm{v},1} & \ldots & N_{\mathrm{v},m} \end{bmatrix}.$$

Inserting the matrices from (6.9) and (6.10) into the different transfer functions concepts from Section 2.3.2 allows deriving subsystem transfer function formulations for (6.2). This will motivate a more general formulation of structured transfer functions for quadratic-bilinear systems similar to those in Chapter 5 for structured bilinear systems.

6.3.1 Structured symmetric subsystem transfer functions

For first-order quadratic-bilinear systems, the symmetric transfer functions are given by the formulae (2.38) and (2.39). The special block structure of the system matrices in (6.9) can now be used to develop symmetric transfer functions for second-order

quadratic-bilinear systems (6.2). For the linear case with $k = 1$,

$$\begin{aligned}
\mathsf{S}_{\mathrm{Q,sym},1}(s_1) &= (s_1\mathsf{E} - \mathsf{A})^{-1}\mathsf{B} \\
&= \left(s_1 \begin{bmatrix} I_{n_2} & 0 \\ 0 & M \end{bmatrix} - \begin{bmatrix} 0 & I_{n_2} \\ -K & -E \end{bmatrix}\right)^{-1} \begin{bmatrix} 0 \\ B_{\mathrm{u}} \end{bmatrix} \\
&= \begin{bmatrix} * & (s_1^2M + s_1E + K)^{-1} \\ * & s_1(s_1^2M + s_1E + K)^{-1} \end{bmatrix} \begin{bmatrix} 0 \\ B_{\mathrm{u}} \end{bmatrix} \\
&= \begin{bmatrix} (s_1^2M + s_1E + K)^{-1}B_{\mathrm{u}} \\ s_1(s_1^2M + s_1E + K)^{-1}B_{\mathrm{u}} \end{bmatrix} \\
&=: \begin{bmatrix} S_{\mathrm{Q,sym},1}(s_1) \\ s_1S_{\mathrm{Q,sym},1}(s_1) \end{bmatrix}
\end{aligned}$$

holds, where $*$ denotes entries that are multiplied with zero and, therefore, can be omitted. As consequence, the first symmetric transfer function can be written as

$$\begin{aligned}
G_{\mathrm{Q,sym},1}(s_1) &= (C_{\mathrm{p}} + s_1C_{\mathrm{v}})(s_1^2M + s_1E + K)^{-1}B_{\mathrm{u}} \\
&= (C_{\mathrm{p}} + s_1C_{\mathrm{v}})S_{\mathrm{Q,sym},1}(s_1).
\end{aligned}$$

For the second symmetric subsystem transfer function, the recursion formula (2.39) needs to be used. With (6.11), the application of the quadratic term can be re-written into

$$\begin{aligned}
&\mathsf{H}(\mathsf{S}_{\mathrm{Q,sym},1}(s_1) \otimes \mathsf{S}_{\mathrm{Q,sym},1}(s_2)) \\
&= -\begin{bmatrix} 0 & 0 & 0 & 0 \\ H_{\mathrm{pp}} & H_{\mathrm{pv}} & H_{\mathrm{vp}} & H_{\mathrm{vv}} \end{bmatrix} \left(\begin{bmatrix} S_{\mathrm{Q,sym},1}(s_1) \\ s_1S_{\mathrm{Q,sym},1}(s_1) \end{bmatrix} \otimes \begin{bmatrix} S_{\mathrm{Q,sym},1}(s_2) \\ s_2S_{\mathrm{Q,sym},1}(s_2) \end{bmatrix}\right) \\
&= \begin{bmatrix} 0 \\ -(H_{\mathrm{pp}} + s_2H_{\mathrm{pv}} + s_1H_{\mathrm{vp}} + s_1s_2H_{\mathrm{vv}})\Big(S_{\mathrm{Q,sym},1}(s_1) \otimes S_{\mathrm{Q,sym},1}(s_2)\Big) \end{bmatrix}.
\end{aligned} \tag{6.12}$$

Similarly, the bilinear terms yield

$$\begin{aligned}
&\mathsf{N}\Big(I_m \otimes \big(\mathsf{S}_{\mathrm{Q,sym},1}(s_1) + \mathsf{S}_{\mathrm{Q,sym},1}(s_2)\big)\Big) \\
&= \begin{bmatrix} 0 & 0 \\ N_{\mathrm{p}} & N_{\mathrm{v}} \end{bmatrix} \left(I_m \otimes \left(\begin{bmatrix} S_{\mathrm{Q,sym},1}(s_1) \\ s_1S_{\mathrm{Q,sym},1}(s_1) \end{bmatrix} + \begin{bmatrix} S_{\mathrm{Q,sym},1}(s_2) \\ s_1S_{\mathrm{Q,sym},1}(s_2) \end{bmatrix}\right)\right) \\
&= \begin{bmatrix} 0 \\ (N_{\mathrm{p}} + s_1N_{\mathrm{v}})\Big(I_m \otimes S_{\mathrm{Q,sym},1}(s_1)\Big) \end{bmatrix} + \begin{bmatrix} 0 \\ (N_{\mathrm{p}} + s_2N_{\mathrm{v}})\Big(I_m \otimes S_{\mathrm{Q,sym},1}(s_2)\Big) \end{bmatrix}.
\end{aligned} \tag{6.13}$$

Collecting together linear, bilinear, and quadratic components leads to the second

symmetric subsystem transfer function

$$\begin{aligned}
&G_{\mathrm{Q,sym},2}(s_1,s_2)\\
&= -\frac{1}{2}\Big(C_\mathrm{p}+(s_1+s_2)C_\mathrm{v}\Big)\Big((s_1+s_2)^2M+(s_1+s_2)E+K\Big)^{-1}\\
&\quad\times\Big((H_\mathrm{pp}+s_2H_\mathrm{pv}+s_1H_\mathrm{vp}+s_1s_2H_\mathrm{vv})\Big(S_{\mathrm{Q,sym},1}(s_1)\otimes S_{\mathrm{Q,sym},1}(s_2)\Big)\\
&\quad+(H_\mathrm{pp}+s_1H_\mathrm{pv}+s_2H_\mathrm{vp}+s_1s_2H_\mathrm{vv})\Big(S_{\mathrm{Q,sym},1}(s_2)\otimes S_{\mathrm{Q,sym},1}(s_1)\Big)\\
&\quad-(N_\mathrm{p}+s_1N_\mathrm{v})\Big(I_m\otimes S_{\mathrm{Q,sym},1}(s_1)\Big)-(N_\mathrm{p}+s_2N_\mathrm{v})\Big(I_m\otimes S_{\mathrm{Q,sym},1}(s_2)\Big)\Big)\\
&=:\Big(C_\mathrm{p}+(s_1+s_2)C_\mathrm{v}\Big)S_{\mathrm{Q,sym},2}(s_1,s_2).
\end{aligned}$$

Following the recursion formula in (2.39) for unstructured systems and using the idea of structured matrix-valued functions as in (3.18) and (5.3) lead to the following formulation of *structured symmetric subsystem transfer functions*:

$$\mathcal{G}_{\mathrm{Q,sym},k}(s_1,\dots,s_k)=\mathcal{C}\left(\sum_{j=1}^{k}s_k\right)\mathcal{S}_{\mathrm{Q,sym},k}(s_1,\dots,s_k), \tag{6.14}$$

for $k\geq 1$, with the recursion

$$\begin{aligned}
\mathcal{S}_{\mathrm{Q,sym},1}(s_1)&=\mathcal{K}(s_1)^{-1}\mathcal{B}(s_1),\\
\mathcal{S}_{\mathrm{Q,sym},k}(s_1,\dots,s_k)&=\frac{1}{k!}\left(\mathcal{K}\Big(\sum_{j=1}^{k}s_k\Big)\right)^{-1}\\
&\quad\times\left(\left(\sum_{j=1}^{k-1}\left(\sum_{\substack{1\leq\alpha_1<\ldots<\alpha_j\leq k\\ 1\leq\alpha_{j+1}<\ldots<\alpha_k\leq k\\ \alpha_i\neq\alpha_\ell\text{ for }i\neq\ell}}\mathcal{H}\left(\sum_{i=1}^{j}s_{\alpha_i},\sum_{\ell=j+1}^{k}s_{\alpha_\ell}\right)\right.\right.\right.\\
&\qquad\left.\left.\left.\times\Big(\mathcal{S}_{\mathrm{Q,sym},j}(s_{\alpha_1},\dots,s_{\alpha_j})\otimes\mathcal{S}_{\mathrm{Q,sym},k-j}(s_{\alpha_{j+1}},\dots,s_{\alpha_k})\Big)\right)\right)\right.\\
&\quad\left.+\sum_{1\leq\beta_1<\ldots<\beta_{k-1}\leq k}\mathcal{N}\left(\sum_{j=1}^{k-1}s_{\beta_j}\right)\Big(I_m\otimes\mathcal{S}_{\mathrm{Q,sym},k-1}(s_{\beta_1},\dots,s_{\beta_{k-1}})\Big)\right),
\end{aligned} \tag{6.15}$$

the matrix-valued functions $\mathcal{C}\colon \mathbb{C} \to \mathbb{C}^{p\times n}$, $\mathcal{K}\colon \mathbb{C} \to \mathbb{C}^{n\times n}$, $\mathcal{B}\colon \mathbb{C} \to \mathbb{C}^{n\times m}$, $\mathcal{H}\colon \mathbb{C}\times\mathbb{C} \to \mathbb{C}^{n\times n^2}$, and $\mathcal{N}(s) = \begin{bmatrix}\mathcal{N}_1(s) & \dots & \mathcal{N}_m(s)\end{bmatrix}$, with $\mathcal{N}_j\colon \mathbb{C} \to \mathbb{C}^{n\times n}$ for $j = 1, \dots, m$.

In this new structured symmetric transfer function framework, the classical first-order quadratic-bilinear systems (2.35) are given by setting

$$\mathcal{C}(s) = \mathsf{C}, \quad \mathcal{B}(s) = \mathsf{B}, \quad \mathcal{K}(s) = s\mathsf{E} - \mathsf{A}, \quad \mathcal{H}(s_1, s_2) = \mathsf{H}, \quad \mathcal{N}(s) = \mathsf{N}. \tag{6.16}$$

In the second-order system case (6.2), the symmetric transfer functions can be recovered from the structured formulation using

$$\begin{aligned}
\mathcal{C}(s) &= C_{\mathrm{p}} + sC_{\mathrm{v}},\\
\mathcal{B}(s) &= B_{\mathrm{u}},\\
\mathcal{K}(s) &= s^2M + sE + K,\\
\mathcal{H}(s_1, s_2) &= -(H_{\mathrm{pp}} + s_2H_{\mathrm{pv}} + s_1H_{\mathrm{vp}} + s_1s_2H_{\mathrm{vv}}),\\
\mathcal{N}(s) &= N_{\mathrm{p}} + sN_{\mathrm{v}}.
\end{aligned} \tag{6.17}$$

For illustration of the structured symmetric transfer function formulae (6.14) and (6.15), these are used to write down the third symmetric subsystem transfer function in the SISO system case to be

$$\mathcal{G}_{\mathrm{Q,sym},3}(s_1, s_2, s_3) = \mathcal{C}(s_1 + s_2 + s_3)\mathcal{S}_{\mathrm{Q,sym},3}(s_1, s_2, s_3),$$

with

$$\begin{aligned}
\mathcal{S}_{\mathrm{Q,sym},3}(s_1, s_2, s_3) = \frac{1}{6}&\mathcal{K}(s_1 + s_2 + s_3)^{-1}\\
&\times \Big(\mathcal{H}(s_1, s_2 + s_3)\Big(\mathcal{S}_{\mathrm{Q,sym},1}(s_1) \otimes \mathcal{S}_{\mathrm{Q,sym},2}(s_2, s_3)\Big)\\
&+ \mathcal{H}(s_2, s_1 + s_3)\Big(\mathcal{S}_{\mathrm{Q,sym},1}(s_2) \otimes \mathcal{S}_{\mathrm{Q,sym},2}(s_1, s_3)\Big)\\
&+ \mathcal{H}(s_3, s_1 + s_2)\Big(\mathcal{S}_{\mathrm{Q,sym},1}(s_3) \otimes \mathcal{S}_{\mathrm{Q,sym},2}(s_1, s_2)\Big)\\
&+ \mathcal{H}(s_1 + s_2, s_3)\Big(\mathcal{S}_{\mathrm{Q,sym},2}(s_1, s_2) \otimes \mathcal{S}_{\mathrm{Q,sym},1}(s_3)\Big)\\
&+ \mathcal{H}(s_1 + s_3, s_2)\Big(\mathcal{S}_{\mathrm{Q,sym},2}(s_1, s_3) \otimes \mathcal{S}_{\mathrm{Q,sym},1}(s_2)\Big)\\
&+ \mathcal{H}(s_2 + s_3, s_1)\Big(\mathcal{S}_{\mathrm{Q,sym},2}(s_2, s_3) \otimes \mathcal{S}_{\mathrm{Q,sym},1}(s_1)\Big)\\
&+ \mathcal{N}(s_1 + s_2)\mathcal{S}_{\mathrm{Q,sym},2}(s_1, s_2)\\
&+ \mathcal{N}(s_1 + s_3)\mathcal{S}_{\mathrm{Q,sym},2}(s_1, s_3)\\
&+ \mathcal{N}(s_2 + s_3)\mathcal{S}_{\mathrm{Q,sym},2}(s_2, s_3)\Big).
\end{aligned}$$

6.3.2 Structured regular subsystem transfer functions

Next, regular transfer functions are considered, with (2.40) and (2.41) for unstructured systems. As before, the second-order quadratic-bilinear system (6.2) is used as motivational structure. Since the first regular subsystem transfer function resembles the linear case, from the previous section it is already known that

$$\begin{aligned} G_{\mathrm{Q,reg},1}(s_1) &= (C_\mathrm{p} + s_1 C_\mathrm{v})(s_1^2 M + s_1 E + K)^{-1} B_\mathrm{u} \\ &=: (C_\mathrm{p} + s_1 C_\mathrm{v}) S_{\mathrm{Q,reg},1}(s_1). \end{aligned}$$

Following the same calculations as in (6.12) and (6.13) for the quadratic and bilinear parts as in the previous section with the symmetric transfer functions, the second regular subsystem transfer function of (6.2) is given by

$$\begin{aligned} G_{\mathrm{Q,reg},2}(s_1, s_2) = &-(C_\mathrm{p} + s_2 C_\mathrm{v})(s_2^2 M + s_2 E + K)^{-1} \\ &\times \Big(\big(H_\mathrm{pp} + s_1 H_\mathrm{pv} + (s_2 - s_1) H_\mathrm{vp} + s_1 (s_2 - s_1) H_\mathrm{vv} \big) \\ &\times \big(S_{\mathrm{Q,reg},1}(s_2 - s_1) \otimes S_{\mathrm{Q,reg},1}(s_1) \big) \\ &- (N_\mathrm{p} + s_1 N_\mathrm{v}) \big(I_m \otimes S_{\mathrm{Q,reg},1}(s_1) \big) \Big) \\ =: &\, (C_\mathrm{p} + s_2 C_\mathrm{v}) S_{\mathrm{Q,reg},2}(s_1, s_2). \end{aligned}$$

Following this scheme with the recursion formula (2.41) yields the *structured regular subsystem transfer functions* to be defined by

$$\mathcal{G}_{\mathrm{Q,reg},k}(s_1, \ldots, s_k) = \mathcal{C}(s_k) \mathcal{S}_{\mathrm{Q,reg},k}(s_1, \ldots, s_k), \tag{6.18}$$

for $k \geq 1$, with the recursion

$$\begin{aligned} \mathcal{S}_{\mathrm{Q,reg},1}(s_1) &= \mathcal{K}(s_1)^{-1} \mathcal{B}(s_1), \\ \mathcal{S}_{\mathrm{Q,reg},k}(s_1, \ldots, s_k) &= \mathcal{K}(s_k)^{-1} \Bigg(\sum_{j=1}^{k-1} \mathcal{H}(s_k - s_{k-j}, s_{k-j}) \\ &\quad \times \big(\mathcal{S}_{\mathrm{Q,reg},j}(s_{k-j+1} - s_{k-j}, \ldots, s_k - s_{k-j}) \\ &\quad \otimes \mathcal{S}_{\mathrm{Q,reg},k-j}(s_1, \ldots, s_{k-j}) \big) \\ &\quad + \mathcal{N}(s_{k-1}) \big(I_m \otimes \mathcal{S}_{\mathrm{Q,reg},k-1}(s_1, \ldots, s_{k-1}) \big) \Bigg), \end{aligned} \tag{6.19}$$

and with the matrix-valued functions as before $\mathcal{C}\colon \mathbb{C} \to \mathbb{C}^{p \times n}$, $\mathcal{K}\colon \mathbb{C} \to \mathbb{C}^{n \times n}$, $\mathcal{B}\colon \mathbb{C} \to \mathbb{C}^{n \times m}$, $\mathcal{H}\colon \mathbb{C} \times \mathbb{C} \to \mathbb{C}^{n \times n^2}$, and $\mathcal{N}(s) = \begin{bmatrix} \mathcal{N}_1(s) & \ldots & \mathcal{N}_m(s) \end{bmatrix}$, with $\mathcal{N}_j\colon \mathbb{C} \to \mathbb{C}^{n \times n}$ for $j = 1, \ldots, m$. The regular subsystem transfer functions for first- and second-order

systems can be recovered using the same instances of the matrix-valued functions as in the symmetric case (6.16) and (6.17).

As in the symmetric transfer function case, the third structured regular subsystem transfer function for SISO systems shall serve as additional illustration of (6.18) and (6.19). This transfer function is given by

$$\begin{aligned}\mathcal{G}_{\mathrm{Q,reg},3}(s_1,s_2,s_3) &= \mathcal{C}(s_3)\mathcal{S}_{\mathrm{Q,reg},3}(s_1,s_2,s_3),\\ \mathcal{S}_{\mathrm{Q,reg},3}(s_1,s_2,s_3) &= \mathcal{K}(s_3)^{-1}\Big(\mathcal{H}(s_3-s_2,s_2)\big(\mathcal{S}_{\mathrm{Q,reg},1}(s_3-s_2)\otimes\mathcal{S}_{\mathrm{Q,reg},2}(s_1,s_2)\big)\\ &\quad+\mathcal{H}(s_3-s_1,s_1)\big(\mathcal{S}_{\mathrm{Q,reg},2}(s_2-s_1,s_3-s_1)\otimes\mathcal{S}_{\mathrm{Q,reg},1}(s_1)\big)\\ &\quad+\mathcal{N}(s_2)\mathcal{S}_{\mathrm{Q,reg},2}(s_1,s_2)\Big).\end{aligned}$$

Remark 6.1 (Regular bilinear vs. quadratic-bilinear transfer functions): The structured regular subsystem transfer functions of quadratic-bilinear systems in (6.18) and (6.19) are a direct extension of the structured regular subsystem transfer functions of bilinear systems from (5.3). This can quickly be seen by setting $\mathcal{H}\equiv 0$, since the transfer functions in (6.18) and (6.19) significantly simplify to products of matrix functions for the linear and bilinear terms as all quadratic parts vanish. ◊

6.3.3 Structured generalized transfer functions

Finally, the structured formulation of the generalized transfer functions from (2.42) and (2.43) is considered. Again, the second-order system (6.2) is used to motivate the more general structure. As for the previous transfer function concepts, the first-level transfer function resembles the linear system case such that for (6.2) one gets

$$G^{(\mathrm{B})}_{\mathrm{Q,gen},1}(s_1) = (C_\mathrm{p}+s_1C_\mathrm{v})(s_1^2M+s_1E+K)^{-1}B_\mathrm{u}.$$

The higher-level transfer functions involving only bilinear terms are exactly the regular transfer functions of bilinear systems, for which the structured extension is given by (5.3). Therefore, the third-level transfer function with one quadratic term is considered next. This can be rewritten for (6.2) to be

$$\begin{aligned}G^{(\mathrm{H},(\mathrm{B}),(\mathrm{B}))}_{\mathrm{Q,gen},3}(s_1,s_2,s_3) &= -(C_\mathrm{p}+s_3C_\mathrm{v})(s_3^2M+s_3E+K)^{-1}\\ &\quad\times(H_{\mathrm{pp}}+s_1H_{\mathrm{pv}}+s_2H_{\mathrm{vp}}+s_1s_2H_{\mathrm{vv}})\\ &\quad\times\Big((s_2^2M+s_2E+K)^{-1}B_\mathrm{u}\otimes(s_1^2M+s_1E+K)^{-1}B_\mathrm{u}\Big).\end{aligned}$$

Consequently, the following structured extension of the generalized transfer functions is proposed: Given the function

$$\Gamma(\gamma, s_1, \ldots, s_j) = \begin{cases} \mathcal{K}(s_1)^{-1}\mathcal{B}(s_1), & \text{if } \gamma = (\mathrm{B}) \\ & \text{and } j = 1, \\ \mathcal{K}(s_j)^{-1}\mathcal{N}(s_{j-1})\Big(I_m \otimes \Gamma(\gamma_2, s_1, \ldots, s_{j-1})\Big), & \text{if } \gamma = (\mathrm{N}, \gamma_2) \\ & \text{and } j \geq 2, \\ \mathcal{K}(s_j)^{-1}\mathcal{H}(s_{j-1}, s_{\ell-1})\Big(\Gamma(\gamma_2, s_\ell, \ldots, s_{j-1}) & \text{if } \gamma = (\mathrm{H}, \gamma_2, \gamma_3) \\ \quad \otimes \Gamma(\gamma_3, s_1, \ldots, s_{\ell-1})\Big) & \text{and } j \geq 3, \end{cases} \tag{6.20}$$

that describes the multiplication of the matrix-valued function for the linear dynamics with the input, bilinear and quadratic components, the *structured generalized transfer functions* are defined to be

$$\mathcal{G}^{\gamma}_{\mathrm{Q,gen},k}(s_1, \ldots, s_k) = \mathcal{C}(s_k)\Gamma(\gamma, s_1, \ldots, s_k), \tag{6.21}$$

with the unique ℓ depending on γ_2, γ_3 in the quadratic case, γ, a nested tuple with the possible elements H, N and B, and tuples of those, the matrix-valued functions as before $\mathcal{C}\colon \mathbb{C} \to \mathbb{C}^{p\times n}$, $\mathcal{K}\colon \mathbb{C} \to \mathbb{C}^{n\times n}$, $\mathcal{B}\colon \mathbb{C} \to \mathbb{C}^{n\times m}$, $\mathcal{H}\colon \mathbb{C}\times\mathbb{C} \to \mathbb{C}^{n\times n^2}$, and $\mathcal{N}(s) = \begin{bmatrix}\mathcal{N}_1(s) & \ldots & \mathcal{N}_m(s)\end{bmatrix}$, with $\mathcal{N}_j\colon \mathbb{C} \to \mathbb{C}^{n\times n}$ for $j = 1, \ldots, m$. Like in the regular and symmetric subsystem transfer function cases, the generalized transfer functions for first- and second-order systems can be recovered using (6.16) and (6.17) as particular instances of the matrix-valued functions, respectively.

For illustration of (6.20) and (6.21), the three fourth-level generalized transfer functions in the SISO system case are written out explicitly. These are given by

$$\begin{aligned}\mathcal{G}^{(\mathrm{N},(\mathrm{N},(\mathrm{N},(\mathrm{B}))))}_{\mathrm{Q,gen},4}(s_1, s_2, s_3, s_4) = \mathcal{C}(s_4)\mathcal{K}(s_4)^{-1}\mathcal{N}(s_3)\mathcal{K}(s_3)^{-1}\mathcal{N}(s_2)\mathcal{K}(s_2)^{-1}\mathcal{N}(s_1) \\ \times \mathcal{K}(s_1)^{-1}\mathcal{B}(s_1),\end{aligned}$$

in the purely bilinear case, and

$$\begin{aligned}\mathcal{G}^{(\mathrm{H},(\mathrm{N},(\mathrm{B})),(\mathrm{B}))}_{\mathrm{Q,gen},4}(s_1, s_2, s_3, s_4) &= \mathcal{C}(s_4)\mathcal{K}(s_4)^{-1}\mathcal{H}(s_3, s_1)\Big(\mathcal{K}(s_3)^{-1}\mathcal{N}(s_2)\mathcal{K}(s_2)^{-1}\mathcal{B}(s_2) \\ &\quad \otimes \mathcal{K}(s_1)^{-1}\mathcal{B}(s_1)\Big), \\ \mathcal{G}^{(\mathrm{N},(\mathrm{H},(\mathrm{B}),(\mathrm{B})))}_{\mathrm{Q,gen},4}(s_1, s_2, s_3, s_4) &= \mathcal{C}(s_4)\mathcal{K}(s_4)^{-1}\mathcal{N}(s_3)\mathcal{K}(s_3)^{-1}\mathcal{H}(s_2, s_1)\Big(\mathcal{K}(s_2)^{-1}\mathcal{B}(s_2) \\ &\quad \otimes \mathcal{K}(s_1)^{-1}\mathcal{B}(s_1)\Big),\end{aligned}$$

with a single quadratic term each.

6.4 Structured transfer function interpolation

The aim of this section is the construction of structure-preserving reduced-order quadratic-bilinear systems based on transfer function interpolation: Given interpolation points $\sigma_1, \ldots, \sigma_k \in \mathbb{C}$, the task is to construct a reduced-order system such that

$$\mathcal{G}_{\mathrm{Q},k}(\sigma_1, \ldots, \sigma_k) = \widehat{\mathcal{G}}_{\mathrm{Q},k}(\sigma_1, \ldots, \sigma_k)$$

holds, where $\mathcal{G}_{\mathrm{Q},k}$ and $\widehat{\mathcal{G}}_{\mathrm{Q},k}$ denote the k-th level full and reduced-order structured transfer functions of one of the three types in Section 6.3 and both systems with the same internal structure. Analogously to the results for linear (Section 3.3.4) and bilinear systems (Chapter 5), the solution to this problem will amount to projection-based model order reduction and conditions on the underlying projection spaces.

The following sections will first extend the projection framework to structured quadratic-bilinear systems before the structured interpolation is discussed for each transfer function concept individually.

6.4.1 Structure-preserving model reduction via projection

Let the original full-order system be described in the frequency domain by any of the introduced transfer function concepts with the matrix-valued functions $\mathcal{C}\colon \mathbb{C} \to \mathbb{C}^{p\times n}$, $\mathcal{K}\colon \mathbb{C} \to \mathbb{C}^{n\times n}$, $\mathcal{B}\colon \mathbb{C} \to \mathbb{C}^{n\times m}$, $\mathcal{H}\colon \mathbb{C}\times\mathbb{C} \to \mathbb{C}^{n\times n^2}$, and $\mathcal{N}(s) = \begin{bmatrix}\mathcal{N}_1(s) & \ldots & \mathcal{N}_m(s)\end{bmatrix}$, with $\mathcal{N}_j\colon \mathbb{C} \to \mathbb{C}^{n\times n}$ for $j = 1, \ldots, m$. Given two basis matrices $V, W \in \mathbb{C}^{n\times r}$, the reduced-order model is described by the truncated matrix functions

$$\begin{aligned}
\widehat{\mathcal{C}}(s) &= \mathcal{C}(s)V, & \widehat{\mathcal{B}}(s) &= W^{\mathsf{H}}\mathcal{B}(s),\\
\widehat{\mathcal{K}}(s) &= W^{\mathsf{H}}\mathcal{K}(s)V, & \widehat{\mathcal{N}}(s) &= W^{\mathsf{H}}\mathcal{N}(s)(I_m \otimes V),\\
\widehat{\mathcal{H}}(s_1, s_2) &= W^{\mathsf{H}}\mathcal{H}(s_1, s_2)(V \otimes V).
\end{aligned} \tag{6.22}$$

The structure preservation via projection follows exactly the same idea of the frequency-affine decomposition in (3.21) and (3.22) as for linear and bilinear systems.

As example, the second-order quadratic-bilinear systems (6.2) is considered, where the matrix functions in the frequency domain were determined in Section 6.3 to be

$$\begin{aligned}
\mathcal{C}(s) &= C_{\mathrm{p}} + sC_{\mathrm{v}},\\
\mathcal{B}(s) &= B_{\mathrm{u}},\\
\mathcal{K}(s) &= s^2M + sE + K,\\
\mathcal{N}(s) &= N_{\mathrm{p}} + sN_{\mathrm{v}},\\
\mathcal{H}(s_1, s_2) &= -(H_{\mathrm{pp}} + s_2H_{\mathrm{pv}} + s_1H_{\mathrm{vp}} + s_1s_2H_{\mathrm{vv}}).
\end{aligned} \tag{6.23}$$

Given the model reduction bases $V, W \in \mathbb{C}^{n\times r}$, the reduced-order model is given by

$$\begin{aligned}
\widehat{\mathcal{C}}(s) &= (C_{\mathrm{p}}V) + s(C_{\mathrm{v}}V) = \widehat{C}_{\mathrm{p}} + s\widehat{C}_{\mathrm{v}},\\
\widehat{\mathcal{B}}(s) &= (W^{\mathsf{H}}B_{\mathrm{u}}) = \widehat{B}_{\mathrm{u}},\\
\widehat{\mathcal{K}}(s) &= s^2(W^{\mathsf{H}}MV) + s(W^{\mathsf{H}}EV) + (W^{\mathsf{H}}KV) = s^2\widehat{M} + s\widehat{E} + \widehat{K},\\
\widehat{\mathcal{N}}(s) &= \Big(W^{\mathsf{H}}N_{\mathrm{p}}(I_m \otimes V)\Big) + s\Big(W^{\mathsf{H}}N_{\mathrm{v}}(I_m \otimes V)\Big) = \widehat{N}_{\mathrm{p}} + s\widehat{N}_{\mathrm{v}},\\
\widehat{\mathcal{H}}(s_1, s_2) &= -\Big(\Big(W^{\mathsf{H}}H_{\mathrm{pp}}(V \otimes V)\Big) + s_2\Big(W^{\mathsf{H}}H_{\mathrm{pv}}(V \otimes V)\Big)\\
&\qquad + s_1\Big(W^{\mathsf{H}}H_{\mathrm{vp}}(V \otimes V)\Big) + s_1s_2\Big(W^{\mathsf{H}}H_{\mathrm{vv}})(V \otimes V)\Big)\Big)\\
&= -(\widehat{H}_{\mathrm{pp}} + s_2\widehat{H}_{\mathrm{pv}} + s_1\widehat{H}_{\mathrm{vp}} + s_1s_2\widehat{H}_{\mathrm{vv}}).
\end{aligned}$$

The complete second-order structure is inherited in the reduced-order matrix functions and the truncated matrices give a realization of a reduced-order second-order quadratic-bilinear system.

As noted in [30], the actual computation of the reduced-order model becomes nontrivial in case of quadratic-bilinear systems. The arising problem is the Kronecker product of the right truncation matrix in (6.22), which is used for the reduction of the quadratic term. The resulting matrix $V \otimes V$ will be of dimension $n^2 \times r^2$ and dense, which comes along with a huge demand for memory to save this truncation matrix and even more resources needed to formulate the multiplication with the quadratic term. To avoid the explicit computation of the Kronecker product, one can use the underlying tensor $\boldsymbol{\mathcal{H}}^{(1)} = \mathcal{H}$ as suggested in [30]. In terms of later implementations, it is beneficial to use the ideas from tensor algebra but working directly with parts of the matrix function $\mathcal{H}$ instead of its underlying tensor $\boldsymbol{\mathcal{H}}$. Using the notation from Section 2.1.1, the reduction of the quadratic term can also be written as

$$\begin{aligned}
&W^{\mathsf{H}}\mathcal{H}(s_1, s_2)(V \otimes V)\\
&= W^{\mathsf{H}}\begin{bmatrix}\mathcal{H}_1(s_1, s_2) & \dots & \mathcal{H}_n(s_1, s_2)\end{bmatrix}\begin{bmatrix}v_{11}V & \cdots & v_{1r}V\\ \vdots & \ddots & \vdots\\ v_{n1}V & \cdots & v_{nr}V\end{bmatrix}\\
&= \Big[v_{11}W^{\mathsf{H}}\mathcal{H}_1(s_1, s_2)V + \ldots + v_{n1}W^{\mathsf{H}}\mathcal{H}_n(s_1, s_2)V \quad \ldots\\
&\qquad v_{1r}W^{\mathsf{H}}\mathcal{H}_1(s_1, s_2)V + \ldots + v_{nr}W^{\mathsf{H}}\mathcal{H}_n(s_1, s_2)V\Big]\\
&= \Big[v_{11}\widetilde{\mathcal{H}}_1(s_1, s_2) + \ldots + v_{n1}\widetilde{\mathcal{H}}_n(s_1, s_2) \quad \ldots \quad v_{1r}\widetilde{\mathcal{H}}_1(s_1, s_2) + \ldots + v_{nr}\widetilde{\mathcal{H}}_n(s_1, s_2)\Big]\\
&= \Big[\widehat{\mathcal{H}}_1(s_1, s_2) \quad \ldots \quad \widehat{\mathcal{H}}_n(s_1, s_2)\Big]\\
&= \widehat{\mathcal{H}}(s_1, s_2),
\end{aligned}$$

where v_{ij} is the (i, j)-th element of the matrix V, such that instead of forming $V \otimes V$ explicitly, one can work with the $n \times n$ blocks that form $\mathcal{H}$ and compute in a first step

the $\widetilde{\mathcal{H}}_j$, which are then combined into $\widehat{\mathcal{H}}$. Especially in the large-scale sparse setting, where $\mathcal{H}$ only has a few non-zero elements, a lot of computations and memory usage can be avoided using this idea. An alternative approach, that can be applied even if the matricizations or tensor of $\mathcal{H}$ are not given, considers only the application of $\mathcal{H}$ on vectors and matrices:

$$\begin{aligned} W^{\mathsf{H}}\mathcal{H}(s_1,s_2)(V\otimes V) &= W^{\mathsf{H}}\mathcal{H}(s_1,s_2)\left(\begin{bmatrix} v_1 & \dots & v_r \end{bmatrix}\otimes V\right) \\ &= W^{\mathsf{H}}\begin{bmatrix} \mathcal{H}(s_1,s_2)(v_1\otimes V) & \dots & \mathcal{H}(s_1,s_2)(v_r\otimes V)\end{bmatrix} \\ &= W^{\mathsf{H}}\begin{bmatrix} \widetilde{\mathcal{H}}_1(s_1,s_2) & \dots & \widetilde{\mathcal{H}}_r(s_1,s_2)\end{bmatrix} \\ &= \widehat{\mathcal{H}}(s_1,s_2), \end{aligned}$$

where $v_1,\dots,v_r$ are the columns of V. Products of the form $\widetilde{\mathcal{H}}_j(s_1,s_2) = \mathcal{H}(s_1,s_2)(v_j\otimes V)$ can be evaluated rather cheaply by considering the application of $\mathcal{H}(s_1,s_2)$ as a function acting on a vector and a matrix such that the Kronecker products are never computed explicitly. The same ideas should be applied for the construction of the projection spaces in the upcoming sections to avoid any explicit use of Kronecker products in practical computations. Similarly, the Kronecker product in the truncation of the bilinear terms can be avoided by observing that

$$\begin{aligned} W^{\mathsf{H}}\mathcal{N}(s)(I_m\otimes V) &= \begin{bmatrix} W^{\mathsf{H}}\mathcal{N}_1(s)V & \dots & W^{\mathsf{H}}\mathcal{N}_m(s)V\end{bmatrix} \\ &= \begin{bmatrix} \widehat{\mathcal{N}}_1(s) & \dots & \widehat{\mathcal{N}}_m(s)\end{bmatrix} \\ &= \widehat{\mathcal{N}}(s) \end{aligned}$$

holds, i.e., in practice, the single bilinear terms are independently reduced without using their concatenation.

An important difference to the literature when it comes to two-sided projections is the use of symmetric tensors $\boldsymbol{\mathcal{H}}$ for the quadratic term. In general, as mentioned in Section 2.1.1, if $\boldsymbol{\mathcal{H}}$ is not symmetric from the beginning, the symmetrization process in time domain assumes the multiplication of the quadratic term with the Kronecker product of a vector with itself, $\mathsf{x}(t)\otimes\mathsf{x}(t)$. Otherwise, the symmetrization changes the dynamics of the quadratic-bilinear system. This assumption is, for example, not satisfied for mechanical quadratic-bilinear systems (6.2) since these involve products of the state with its derivative, e.g., $x(t)\otimes\dot{x}(t)$. Consequently, the symmetrization method cannot be used for such systems. However, the case of symmetric tensors might naturally occur or symmetrization might be possible, e.g., in case of first-order systems (2.35), for which further theoretic results are added in the upcoming sections. In general, if the tensor $\boldsymbol{\mathcal{H}}$ is assumed to be symmetric, this is explicitly stated in the assumptions.

6.4.2 Interpolating structured symmetric subsystem transfer functions

According to the historical order in which the transfer function concepts for quadratic-bilinear systems have been developed and are mentioned in Section 6.3, the symmetric transfer functions are considered first. The following theorem gives conditions on the right projection space associated with the truncation matrix V for the interpolation of arbitrary high levels of the symmetric subsystem transfer functions.

Theorem 6.2 (Symmetric transfer function interpolation via V):
Let $\mathcal{G}_{\mathrm{Q}}$ be a quadratic-bilinear system, described by its symmetric subsystem transfer functions $\mathcal{G}_{\mathrm{Q,sym},k}$ in (6.14) and (6.15), and $\widehat{\mathcal{G}}_{\mathrm{Q}}$ the reduced-order quadratic-bilinear system constructed by (6.22), with its reduced-order symmetric subsystem transfer functions $\widehat{\mathcal{G}}_{\mathrm{Q,sym},k}$. Also, let $\sigma_1, \ldots, \sigma_k \in \mathbb{C}$ be a set of interpolation points such that the matrix functions $\mathcal{C}$, $\mathcal{K}^{-1}$, $\mathcal{B}$, $\mathcal{N}$, $\mathcal{H}$ and $\widehat{\mathcal{K}}^{-1}$ are defined in these points and sums of combinations of these points. Construct V using

$$\begin{aligned}
V_1 &= \left[\mathcal{S}_{\mathrm{Q,sym},1}(\sigma_{i_1})\right], && 1 \le i_1 \le k,\\
V_2 &= \left[\mathcal{S}_{\mathrm{Q,sym},2}(\sigma_{i_1}, \sigma_{i_2})\right], && 1 \le i_1 < i_2 \le k\\
&\vdots\\
V_k &= \left[\mathcal{S}_{\mathrm{Q,sym},k}(\sigma_{i_1}, \sigma_{i_2}, \ldots, \sigma_{i_k})\right], && 1 \le i_1 < i_2 < \ldots < i_k \le k,\\
\operatorname{span}(V) &\supseteq \operatorname{span}\left(\begin{bmatrix} V_1 & \ldots & V_k \end{bmatrix}\right),
\end{aligned}$$

with the recursive terms from (6.15), and let W be an arbitrary full-rank truncation matrix of appropriate dimension. Then the symmetric transfer functions of $\widehat{\mathcal{G}}_{\mathrm{Q}}$ interpolate those of $\mathcal{G}_{\mathrm{Q}}$ in the following way:

$$\begin{aligned}
\mathcal{G}_{\mathrm{Q,sym},1}(\sigma_{i_1}) &= \widehat{\mathcal{G}}_{\mathrm{Q,sym},1}(\sigma_{i_1}), && 1 \le i_1 \le k,\\
\mathcal{G}_{\mathrm{Q,sym},2}(\sigma_{i_1}, \sigma_{i_2}) &= \widehat{\mathcal{G}}_{\mathrm{Q,sym},2}(\sigma_{i_1}, \sigma_{i_2}), && 1 \le i_1 < i_2 \le k,\\
&\vdots\\
\mathcal{G}_{\mathrm{Q,sym},k}(\sigma_{i_1}, \sigma_{i_2}, \ldots, \sigma_{i_k}) &= \widehat{\mathcal{G}}_{\mathrm{Q,sym},k}(\sigma_{i_1}, \sigma_{i_2}, \ldots, \sigma_{i_k}), && 1 \le i_1 < i_2 < \ldots < i_k \le k.
\end{aligned}$$

Note that the indices in the last line $1 \le i_1 < i_2 < \ldots < i_k \le k$ are equivalent to $i_1 = 1, i_2 = 2, \ldots, i_k = k$. ◊

Before moving on to the proof of Theorem 6.2, for illustration of the used notation and the results, a closer look at the construction of the projection space in Theorem 6.2 is taken. As example, the aim is to interpolate the fourth symmetric subsystem transfer function in the interpolation points $\sigma_1, \sigma_2, \sigma_3, \sigma_4$. The matrices constructed in Theorem 6.2 are the

concatenation of the different resulting recursive terms using the corresponding indices. The constructed matrices look as follows

$$
\begin{aligned}
V_1 &= \Big[\mathcal{S}_{\mathrm{Q,sym},1}(\sigma_1),\ \mathcal{S}_{\mathrm{Q,sym},1}(\sigma_2),\ \mathcal{S}_{\mathrm{Q,sym},1}(\sigma_3),\ \mathcal{S}_{\mathrm{Q,sym},1}(\sigma_4)\Big],\\
V_2 &= \Big[\mathcal{S}_{\mathrm{Q,sym},2}(\sigma_1,\sigma_2),\ \mathcal{S}_{\mathrm{Q,sym},2}(\sigma_1,\sigma_3),\ \mathcal{S}_{\mathrm{Q,sym},2}(\sigma_1,\sigma_4),\\
&\qquad \mathcal{S}_{\mathrm{Q,sym},2}(\sigma_2,\sigma_3),\ \mathcal{S}_{\mathrm{Q,sym},2}(\sigma_2,\sigma_4),\ \mathcal{S}_{\mathrm{Q,sym},2}(\sigma_3,\sigma_3)\Big],\\
V_3 &= \Big[\mathcal{S}_{\mathrm{Q,sym},3}(\sigma_1,\sigma_2,\sigma_3),\ \mathcal{S}_{\mathrm{Q,sym},3}(\sigma_1,\sigma_2,\sigma_4),\ \mathcal{S}_{\mathrm{Q,sym},3}(\sigma_1,\sigma_3,\sigma_4),\ \mathcal{S}_{\mathrm{Q,sym},3}(\sigma_2,\sigma_3,\sigma_4)\Big],\\
V_4 &= \mathcal{S}_{\mathrm{Q,sym},4}(\sigma_1,\sigma_2,\sigma_3,\sigma_4).
\end{aligned}
$$

By construction of the projection space, the following symmetric transfer function values are interpolated

$$
\begin{aligned}
&\mathcal{G}_{\mathrm{Q,sym},1}(\sigma_1),\ \mathcal{G}_{\mathrm{Q,sym},1}(\sigma_2),\ \mathcal{G}_{\mathrm{Q,sym},1}(\sigma_3),\ \mathcal{G}_{\mathrm{Q,sym},1}(\sigma_4),\\
&\mathcal{G}_{\mathrm{Q,sym},2}(\sigma_1,\sigma_2),\ \mathcal{G}_{\mathrm{Q,sym},2}(\sigma_1,\sigma_3),\ \mathcal{G}_{\mathrm{Q,sym},2}(\sigma_1,\sigma_4),\\
&\mathcal{G}_{\mathrm{Q,sym},2}(\sigma_2,\sigma_3),\ \mathcal{G}_{\mathrm{Q,sym},2}(\sigma_2,\sigma_4),\ \mathcal{G}_{\mathrm{Q,sym},2}(\sigma_3,\sigma_4),\\
&\mathcal{G}_{\mathrm{Q,sym},3}(\sigma_1,\sigma_2,\sigma_3),\ \mathcal{G}_{\mathrm{Q,sym},3}(\sigma_1,\sigma_2,\sigma_4),\ \mathcal{G}_{\mathrm{Q,sym},3}(\sigma_1,\sigma_3,\sigma_4),\ \mathcal{G}_{\mathrm{Q,sym},3}(\sigma_2,\sigma_3,\sigma_4),\\
&\mathcal{G}_{\mathrm{Q,sym},4}(\sigma_1,\sigma_2,\sigma_3,\sigma_4).
\end{aligned}
$$

Next, the proof of this result is given.

Proof of Theorem 6.2. The interpolation of the first transfer function level follows from Proposition 3.2 and by construction of V_1. Thereby, only the interpolation of higher transfer function levels is left to be proven. For simplicity, only the second symmetric subsystem transfer function is considered as the rest follows by induction over the transfer function index k. On the side of the reduced-order model, the interpolating transfer functions are given by

$$
\begin{aligned}
\widehat{\mathcal{G}}_{\mathrm{Q,sym},2}(\sigma_{i_1},\sigma_{i_2}) = \frac{1}{2}\widehat{\mathcal{C}}(\sigma_{i_1}+\sigma_{i_2})\widehat{\mathcal{K}}(\sigma_{i_1}+\sigma_{i_2})^{-1}\\
\times\Big(\widehat{\mathcal{H}}(\sigma_{i_1},\sigma_{i_2})\Big(\widehat{\mathcal{S}}_{\mathrm{Q,sym},1}(\sigma_{i_1})\otimes\widehat{\mathcal{S}}_{\mathrm{Q,sym},1}(\sigma_{i_1})\Big)\\
+\widehat{\mathcal{H}}(\sigma_{i_2},\sigma_{i_1})\Big(\widehat{\mathcal{S}}_{\mathrm{Q,sym},1}(\sigma_{i_2})\otimes\widehat{\mathcal{S}}_{\mathrm{Q,sym},1}(\sigma_{i_1})\Big)\\
+\widehat{\mathcal{N}}(\sigma_{i_1})\Big(I_m\otimes\widehat{\mathcal{S}}_{\mathrm{Q,sym},1}(\sigma_{i_1})\Big)+\widehat{\mathcal{N}}(\sigma_{i_2})\Big(I_m\otimes\widehat{\mathcal{S}}_{\mathrm{Q,sym},1}(\sigma_{i_2})\Big)\Big),
\end{aligned}
$$

for $1\leq i_1<i_2\leq k$, where $\widehat{\mathcal{S}}_{\mathrm{Q,sym},1}$ denotes the symmetric recursive level-1 terms using the reduced-order matrix functions. For the recursive terms, it holds

$$
\begin{aligned}
V\widehat{\mathcal{S}}_{\mathrm{Q,sym},1}(\sigma_i) &= V\widehat{\mathcal{K}}(\sigma_i)^{-1}\widehat{\mathcal{B}}(\sigma_i)\\
&= V\widehat{\mathcal{K}}(\sigma_i)^{-1}W^{\mathsf{H}}\mathcal{K}(\sigma_i)\mathcal{K}(\sigma_i)^{-1}\mathcal{B}(\sigma_i)
\end{aligned}
$$

$$
\begin{aligned}
&= P_{\mathrm{V}}(\sigma_i)\mathcal{S}_{\mathrm{Q,sym,1}}(\sigma_i) \\
&= \mathcal{S}_{\mathrm{Q,sym,1}}(\sigma_i),
\end{aligned}
$$

for all $1 \leq i \leq k$, since $\operatorname{span}\left(\mathcal{S}_{\mathrm{Q,sym,1}}(\sigma_i)\right) \subseteq \operatorname{span}(V)$ by means of V_1 and P_{V} is the projector onto $\operatorname{span}(V)$ from (3.24). It follows that

$$
\begin{aligned}
\widehat{\mathcal{G}}_{\mathrm{Q,sym,2}}(\sigma_{i_1}, \sigma_{i_2}) &= \frac{1}{2}\widehat{\mathcal{C}}(\sigma_{i_1} + \sigma_{i_2})\widehat{\mathcal{K}}(\sigma_{i_1} + \sigma_{i_2})^{-1}W^{\mathsf{H}} \\
&\quad \times \Big(\mathcal{H}(\sigma_{i_1}, \sigma_{i_2})\left(\mathcal{S}_{\mathrm{Q,sym,1}}(\sigma_{i_1}) \otimes \mathcal{S}_{\mathrm{Q,sym,1}}(\sigma_{i_2})\right) \\
&\quad + \mathcal{H}(\sigma_{i_2}, \sigma_{i_1})\left(\mathcal{S}_{\mathrm{Q,sym,1}}(\sigma_{i_2}) \otimes \mathcal{S}_{\mathrm{Q,sym,1}}(\sigma_{i_1})\right) \\
&\quad + \mathcal{N}(\sigma_{i_1})\left(I_m \otimes \mathcal{S}_{\mathrm{Q,sym,1}}(\sigma_{i_1})\right) + \mathcal{N}(\sigma_{i_2})\left(I_m \otimes \mathcal{S}_{\mathrm{Q,sym,1}}(\sigma_{i_2})\right)\Big) \\
&= \frac{1}{2}\mathcal{C}(\sigma_{i_1} + \sigma_{i_2})\underbrace{V\widehat{\mathcal{K}}(\sigma_{i_1} + \sigma_{i_2})^{-1}W^{\mathsf{H}}\mathcal{K}(\sigma_{i_1} + \sigma_{i_2})}_{= P_{\mathrm{V}}(\sigma_{i_1} + \sigma_{i_2})}\mathcal{K}(\sigma_{i_1} + \sigma_{i_2})^{-1} \\
&\quad \times \Big(\mathcal{H}(\sigma_{i_1}, \sigma_{i_2})\left(\mathcal{S}_{\mathrm{Q,sym,1}}(\sigma_{i_1}) \otimes \mathcal{S}_{\mathrm{Q,sym,1}}(\sigma_{i_2})\right) \\
&\quad + \mathcal{H}(\sigma_{i_2}, \sigma_{i_1})\left(\mathcal{S}_{\mathrm{Q,sym,1}}(\sigma_{i_2}) \otimes \mathcal{S}_{\mathrm{Q,sym,1}}(\sigma_{i_1})\right) \\
&\quad + \mathcal{N}(\sigma_{i_1})\left(I_m \otimes \mathcal{S}_{\mathrm{Q,sym,1}}(\sigma_{i_1})\right) + \mathcal{N}(\sigma_{i_2})\left(I_m \otimes \mathcal{S}_{\mathrm{Q,sym,1}}(\sigma_{i_2})\right)\Big) \\
&= \frac{1}{2}\mathcal{C}(\sigma_{i_1} + \sigma_{i_2})P_{\mathrm{V}}(\sigma_{i_1} + \sigma_{i_2})\mathcal{S}_{\mathrm{Q,sym,2}}(\sigma_{i_1}, \sigma_{i_2}) \\
&= \mathcal{G}_{\mathrm{Q,sym,2}}(\sigma_{i_1}, \sigma_{i_2})
\end{aligned}
$$

holds for all $1 \leq i_1 < i_2 \leq k$, using again the projector P_{V} and

$$
\operatorname{span}\left(\mathcal{S}_{\mathrm{Q,sym,2}}(\sigma_{i_1}, \sigma_{i_2})\right) \subseteq \operatorname{span}(V),
$$

due to the construction of V_2. The rest of the proof follows via induction over the transfer function index k. □

The quickly growing number of constructed vectors in Theorem 6.2 triggers the question of the resulting projection space dimensions. The number of computed columns that the matrix V_j, $1 \leq j \leq k$, contributes to the projection space is $\binom{k}{j}m^j$. Thereby, for the interpolation of the k-th symmetric subsystem transfer function

$$
\dim\left(\operatorname{span}(V)\right) \geq \sum_{j=1}^{k} \binom{k}{j} m^j \tag{6.24}
$$

holds, under the assumption that all constructed vectors are linear independent. Note that (6.24) simplifies in case of SISO systems to

$$\dim\Big(\operatorname{span}(V)\Big) \geq \sum_{j=1}^{k} \binom{k}{j} = 2^k - 1. \tag{6.25}$$

While (6.24) and (6.25) can easily become pretty large, one needs to note that, at the same time, (6.25) many scalar or matrix interpolation conditions are matched.

In practice, due to the complexity of the symmetric transfer functions often only the first two levels are actively used. Therefore, the following corollary states the special case of restricting Theorem 6.2 to the first two symmetric subsystem transfer functions and formulates the construction of the projection space with the actual matrix-valued functions instead of the recursive terms.

Corollary 6.3 (Simplified symmetric transfer function interpolation):
Let $\mathcal{G}_{\mathrm{Q}}$ be a quadratic-bilinear system, described by its symmetric subsystem transfer functions $\mathcal{G}_{\mathrm{Q,sym},k}$ in (6.14) and (6.15), and $\widehat{\mathcal{G}}_{\mathrm{Q}}$ the reduced-order quadratic-bilinear system constructed by (6.22), with its reduced-order symmetric transfer functions $\widehat{\mathcal{G}}_{\mathrm{Q,sym},k}$. Also, let σ_1, $\sigma_2 \in \mathbb{C}$ be two interpolation points such that the matrix functions $\mathcal{C}$, $\mathcal{K}^{-1}$, $\mathcal{B}$, $\mathcal{N}$, $\mathcal{H}$ and $\widehat{\mathcal{K}}^{-1}$ are defined in these points and their sum. Construct V using

$$\begin{aligned}
V_{1,1} &= \mathcal{K}(\sigma_1)^{-1}\mathcal{B}(\sigma_1),\\
V_{1,2} &= \mathcal{K}(\sigma_2)^{-1}\mathcal{B}(\sigma_2),\\
V_2 &= \mathcal{K}(\sigma_1+\sigma_2)^{-1}\Big(\mathcal{H}(\sigma_1,\sigma_2)(V_{1,1}\otimes V_{1,2}) + \mathcal{H}(\sigma_2,\sigma_1)(V_{1,2}\otimes V_{1,1})\\
&\quad + \mathcal{N}(\sigma_1)(I_m \otimes V_{1,1}) + \mathcal{N}(\sigma_2)(I_m \otimes V_{1,2})\Big),\\
\operatorname{span}(V) &\supseteq \operatorname{span}\left(\begin{bmatrix} V_{1,1} & V_{1,2} & V_2 \end{bmatrix}\right),
\end{aligned}$$

and let W be an arbitrary full-rank truncation matrix of appropriate dimension. Then the symmetric transfer functions of $\widehat{\mathcal{G}}_{\mathrm{Q}}$ interpolate those of $\mathcal{G}_{\mathrm{Q}}$ in the following way:

$$\begin{aligned}
\mathcal{G}_{\mathrm{Q,sym},1}(\sigma_1) &= \widehat{\mathcal{G}}_{\mathrm{Q,sym},1}(\sigma_1),\\
\mathcal{G}_{\mathrm{Q,sym},1}(\sigma_2) &= \widehat{\mathcal{G}}_{\mathrm{Q,sym},1}(\sigma_2),\\
\mathcal{G}_{\mathrm{Q,sym},2}(\sigma_1,\sigma_2) &= \widehat{\mathcal{G}}_{\mathrm{Q,sym},1}(\sigma_1,\sigma_2).
\end{aligned}$$ ◊

Another possibility of limiting the dimension growth of the underlying projection space is used in the literature, e.g., in [30]. A specific choice of interpolation points can be used to construct as many linear dependent vectors as possible such that the dimension of the underlying projection space becomes as small as possible. For symmetric subsystem transfer functions, this can be done by choosing

$$\sigma_1 = \sigma_2 = \ldots = \sigma_k = \sigma,$$

with a single interpolation point $\sigma \in \mathbb{C}$. By this choice, the dimension of the projection space (6.24) for the interpolation of the k-th symmetric subsystem transfer function can be reduced to $\sum_{j=1}^{k} m^j$. Especially in Theorem 6.2, a lot of computations can be avoided since for each basis contribution V_j only m^j columns are needed. Consider the previous example of interpolating the fourth symmetric subsystem transfer function. The computation of the projection space is now restricted to

$$\begin{aligned} V_1 &= \mathcal{S}_{\mathrm{Q,sym},1}(\sigma), & V_2 &= \mathcal{S}_{\mathrm{Q,sym},2}(\sigma,\sigma), \\ V_3 &= \mathcal{S}_{\mathrm{Q,sym},3}(\sigma,\sigma,\sigma), & V_4 &= \mathcal{S}_{\mathrm{Q,sym},4}(\sigma,\sigma,\sigma,\sigma), \end{aligned}$$

which yields the interpolation of

$$\mathcal{G}_{\mathrm{Q,sym},1}(\sigma), \quad \mathcal{G}_{\mathrm{Q,sym},2}(\sigma,\sigma), \quad \mathcal{G}_{\mathrm{Q,sym},3}(\sigma,\sigma,\sigma), \quad \mathcal{G}_{\mathrm{Q,sym},4}(\sigma,\sigma,\sigma,\sigma).$$

In contrast to Proposition 3.2 and the results in Chapter 5, there will be nothing equivalent to Theorem 6.2 for the construction of W. This follows directly from the projection framework (6.22) and the quadratic term, which involves the Kronecker product of two V matrices but only a single W matrix to be multiplied from the left. Still, there are advantageous choices for W to increase the number of matched interpolation conditions, as, for example, used in [30]. A first suggestion is given in the following lemma, which enables the interpolation of the second symmetric subsystem transfer function without evaluating quadratic or bilinear terms.

Lemma 6.4 (Implicit symmetric transfer function interpolation):
Let $\mathcal{G}_{\mathrm{Q}}$ be a quadratic-bilinear system, described by its symmetric subsystem transfer functions $\mathcal{G}_{\mathrm{Q,sym},k}$ in (6.14) and (6.15), $\widehat{\mathcal{G}}_{\mathrm{Q}}$ the reduced-order quadratic-bilinear system, constructed by (6.22) with its reduced-order symmetric transfer functions $\widehat{\mathcal{G}}_{\mathrm{Q,sym},k}$, and $\sigma_1, \sigma_2 \in \mathbb{C}$ two interpolation points such that the matrix functions $\mathcal{C}$, $\mathcal{K}^{-1}$, $\mathcal{B}$, $\mathcal{N}$, $\mathcal{H}$ and $\widehat{\mathcal{K}}^{-1}$ are defined in these points and their sum. Construct V using

$$\operatorname{span}(V) \supseteq \operatorname{span}\left(\begin{bmatrix} \mathcal{K}(\sigma_1)^{-1}\mathcal{B}(\sigma_1) & \mathcal{K}(\sigma_2)^{-1}\mathcal{B}(\sigma_2) \end{bmatrix}\right),$$

and W using

$$\operatorname{span}(W) \supseteq \operatorname{span}\left(\mathcal{K}(\sigma_1+\sigma_2)^{-\mathsf{H}}\mathcal{C}(\sigma_1+\sigma_2)^{\mathsf{H}}\right),$$

and let the two matrices V and W be of appropriate dimensions. Then the symmetric transfer functions of $\widehat{\mathcal{G}}_{\mathrm{Q}}$ interpolate those of $\mathcal{G}_{\mathrm{Q}}$ in the following way:

$$\begin{aligned} \mathcal{G}_{\mathrm{Q,sym},1}(\sigma_1) &= \widehat{\mathcal{G}}_{\mathrm{Q,sym},1}(\sigma_1), & \mathcal{G}_{\mathrm{Q,sym},1}(\sigma_2) &= \widehat{\mathcal{G}}_{\mathrm{Q,sym},1}(\sigma_2), \\ \mathcal{G}_{\mathrm{Q,sym},1}(\sigma_1+\sigma_2) &= \widehat{\mathcal{G}}_{\mathrm{Q,sym},1}(\sigma_1+\sigma_2), & \mathcal{G}_{\mathrm{Q,sym},2}(\sigma_1,\sigma_2) &= \widehat{\mathcal{G}}_{\mathrm{Q,sym},2}(\sigma_1,\sigma_2). \end{aligned}$$

◊

Proof. While the interpolation of the first symmetric subsystem transfer function $\mathcal{G}_{\mathrm{Q,sym},1}$ in the points σ_1, σ_2 and $\sigma_1 + \sigma_2$ follows directly from the construction of $\operatorname{span}(V)$ and $\operatorname{span}(W)$ together with Proposition 3.2, the interpolation of the second symmetric subsystem transfer function still needs to be shown. From the proof of Theorem 6.2, it is already known that

$$\begin{aligned}\widehat{\mathcal{G}}_{\mathrm{Q,sym},2}(\sigma_1,\sigma_2) = \frac{1}{2}\widehat{\mathcal{C}}(\sigma_1+\sigma_2)\widehat{\mathcal{K}}(\sigma_1+\sigma_2)^{-1}W^{\mathsf{H}} \\ \times \Big(\mathcal{H}(\sigma_1,\sigma_2)\big(\mathcal{S}_{\mathrm{Q,sym},1}(\sigma_1)\otimes\mathcal{S}_{\mathrm{Q,sym},1}(\sigma_2)\big) \\ + \mathcal{H}(\sigma_2,\sigma_1)\big(\mathcal{S}_{\mathrm{Q,sym},1}(\sigma_2)\otimes\mathcal{S}_{\mathrm{Q,sym},1}(\sigma_1)\big) \\ + \mathcal{N}(\sigma_{i_1})\big(I_m\otimes\mathcal{S}_{\mathrm{Q,sym},1}(\sigma_{i_1})\big) + \mathcal{N}(\sigma_{i_2})\big(I_m\otimes\mathcal{S}_{\mathrm{Q,sym},1}(\sigma_{i_2})\big)\Big)\end{aligned}$$

holds via the construction of V. Taking a look at the first line, one can see that

$$\begin{aligned}\widehat{\mathcal{C}}(\sigma_1+\sigma_2)\widehat{\mathcal{K}}(\sigma_1+\sigma_2)^{-1}W^{\mathsf{H}} &= \mathcal{C}(\sigma_1+\sigma_2)\mathcal{K}(\sigma_1+\sigma_2)^{-1}\mathcal{K}(\sigma_1+\sigma_2)V\widehat{\mathcal{K}}(\sigma_1+\sigma_2)^{-1}W^{\mathsf{H}} \\ &= \underbrace{\mathcal{C}(\sigma_1+\sigma_2)\mathcal{K}(\sigma_1+\sigma_2)^{-1}}_{=:z,\ \operatorname{span}(z^{\mathsf{H}})\subseteq\operatorname{span}(W)} P_{\mathrm{W}}(\sigma_1+\sigma_2)^{\mathsf{H}} \\ &= \mathcal{C}(\sigma_1+\sigma_2)\mathcal{K}(\sigma_1+\sigma_2)^{-1}\end{aligned}$$

yields the desired interpolation result, where P_{W} is the projector from (3.25) onto the projection space $\operatorname{span}(W)$. □

A generalization of Lemma 6.4 to higher transfer function levels is possible such that the explicit sampling of the k-th symmetric subsystem transfer function can be avoided for the interpolation. For brevity and less practical relevance, these results are omitted. Instead, a related idea from [30] is considered next using the two-sided projection approach to interpolate the first-order partial derivatives of the second symmetric subsystem transfer function.

Theorem 6.5 (Implicit Hermite interpolation of sym. transfer functions): Let $\mathcal{G}_{\mathrm{Q}}$ be a quadratic-bilinear system, described by its symmetric subsystem transfer functions $\mathcal{G}_{\mathrm{Q,sym},k}$ in (6.14) and (6.15), $\widehat{\mathcal{G}}_{\mathrm{Q}}$ the reduced-order quadratic-bilinear system, constructed by (6.22) with its reduced-order symmetric transfer functions $\widehat{\mathcal{G}}_{\mathrm{Q,sym},k}$, and $\sigma_1,\sigma_2\in\mathbb{C}$ interpolation points such that the matrix functions $\mathcal{C}$, $\mathcal{K}^{-1}$, $\mathcal{B}$, $\mathcal{N}$, $\mathcal{H}$ and $\widehat{\mathcal{K}}^{-1}$ are complex differentiable in these points and their sum. Let the following matrices be given

$$\begin{aligned}V_{1,1} &= \mathcal{K}(\sigma_1)^{-1}\mathcal{B}(\sigma_1), \\ V_{1,2} &= \mathcal{K}(\sigma_2)^{-1}\mathcal{B}(\sigma_2), \\ V_2 &= \mathcal{K}(\sigma_1+\sigma_2)^{-1}\Big(\mathcal{H}(\sigma_1,\sigma_2)(V_{1,1}\otimes V_{1,2}) + \mathcal{H}(\sigma_2,\sigma_1)(V_{1,2}\otimes V_{1,1}) \\ &\quad + \mathcal{N}(\sigma_1)(I_m\otimes V_{1,1}) + \mathcal{N}(\sigma_2)(I_m\otimes V_{1,2})\Big),\end{aligned}$$

and, also,

$$\begin{aligned}
W_1 &= \mathcal{K}(\sigma_1+\sigma_2)^{-\mathsf{H}}\mathcal{C}(\sigma_1+\sigma_2)^{\mathsf{H}},\\
W_2 &= \mathcal{K}(\sigma_1)^{-\mathsf{H}}\Big(\overline{\boldsymbol{\mathcal{H}}^{(2)}(\sigma_1,\sigma_2)}(\overline{V_{1,2}}\otimes W_1)+\overline{\boldsymbol{\mathcal{N}}^{(2)}(\sigma_1)}(I_m\otimes W_1)\Big),\\
W_3 &= \mathcal{K}(\sigma_1)^{-\mathsf{H}}\overline{\boldsymbol{\mathcal{H}}^{(3)}(\sigma_2,\sigma_1)}(\overline{V_{1,2}}\otimes W_1),\\
W_4 &= \mathcal{K}(\sigma_2)^{-\mathsf{H}}\Big(\overline{\boldsymbol{\mathcal{H}}^{(2)}(\sigma_2,\sigma_1)}(\overline{V_{1,1}}\otimes W_1)+\overline{\boldsymbol{\mathcal{N}}^{(2)}(\sigma_2)}(I_m\otimes W_1)\Big),\\
W_5 &= \mathcal{K}(\sigma_2)^{-\mathsf{H}}\overline{\boldsymbol{\mathcal{H}}^{(3)}(\sigma_1,\sigma_2)}(\overline{V_{1,1}}\otimes W_1),
\end{aligned}$$

where $\boldsymbol{\mathcal{H}}^{(2)}, \boldsymbol{\mathcal{H}}^{(3)}$ are the 2- and 3-mode matricizations of the tensor corresponding to the quadratic term such that $\boldsymbol{\mathcal{H}}^{(1)} = \mathcal{H}$, and $\boldsymbol{\mathcal{N}}^{(2)}$ is the 2-mode matricization of the tensor corresponding to the bilinear term such that $\boldsymbol{\mathcal{N}}^{(1)} = \mathcal{N}$. Then the following statements hold:

(a) If V and W are constructed such that

$$\operatorname{span}(V) \supseteq \operatorname{span}\left(\begin{bmatrix} V_{1,1} & V_{1,2}\end{bmatrix}\right) \quad \text{and} \quad \operatorname{span}(W) \supseteq \operatorname{span}(W_1),$$

then the following interpolation conditions hold:

$$\begin{aligned}
\mathcal{G}_{\mathrm{Q,sym},1}(\sigma_1) &= \widehat{\mathcal{G}}_{\mathrm{Q,sym},1}(\sigma_1), & \mathcal{G}_{\mathrm{Q,sym},1}(\sigma_2) &= \widehat{\mathcal{G}}_{\mathrm{Q,sym},1}(\sigma_2),\\
\mathcal{G}_{\mathrm{Q,sym},1}(\sigma_1+\sigma_2) &= \widehat{\mathcal{G}}_{\mathrm{Q,sym},1}(\sigma_1+\sigma_2), & \mathcal{G}_{\mathrm{Q,sym},2}(\sigma_1,\sigma_2) &= \widehat{\mathcal{G}}_{\mathrm{Q,sym},2}(\sigma_1,\sigma_2).
\end{aligned} \tag{6.26}$$

(b) If V and W are constructed such that

$$\operatorname{span}(V) \supseteq \operatorname{span}\left(\begin{bmatrix} V_{1,1} & V_{1,2} & V_2\end{bmatrix}\right) \quad \text{and} \quad \operatorname{span}(W) \supseteq \operatorname{span}\left(\begin{bmatrix} W_1 & W_2 & W_3\end{bmatrix}\right),$$

then, additionally to (6.26), the following Hermite interpolation condition holds:

$$\partial_{s_1}\mathcal{G}_{\mathrm{Q,sym},2}(\sigma_1,\sigma_2) = \partial_{s_1}\widehat{\mathcal{G}}_{\mathrm{Q,sym},2}(\sigma_1,\sigma_2).$$

(c) If V and W are constructed such that

$$\operatorname{span}(V) \supseteq \operatorname{span}\left(\begin{bmatrix} V_{1,1} & V_{1,2} & V_2\end{bmatrix}\right) \quad \text{and} \quad \operatorname{span}(W) \supseteq \operatorname{span}\left(\begin{bmatrix} W_1 & W_4 & W_5\end{bmatrix}\right),$$

then, additionally to (6.26), the following Hermite interpolation condition holds:

$$\partial_{s_2}\mathcal{G}_{\mathrm{Q,sym},2}(\sigma_1,\sigma_2) = \partial_{s_2}\widehat{\mathcal{G}}_{\mathrm{Q,sym},2}(\sigma_1,\sigma_2). \qquad \lozenge$$

Proof. First, note that Part (a) is a reminder of Lemma 6.4. Therefore, only Part (b) and (c) are left to be proven. Consider first the derivative with respect to s_1 of the reduced-order model in Part (b), namely

$$\begin{aligned}
\partial_{s_1}\widehat{\mathcal{G}}_{\mathrm{Q,sym},2}(\sigma_1,\sigma_2) = {} & \frac{1}{2}\partial_s\widehat{\mathcal{C}}(\sigma_1+\sigma_2)\widehat{\mathcal{S}}_{\mathrm{Q,sym},2}(\sigma_1,\sigma_2) \\
& - \frac{1}{2}\widehat{\mathcal{C}}(\sigma_1+\sigma_2)\widehat{\mathcal{K}}(\sigma_1+\sigma_2)^{-1}\partial_s\widehat{\mathcal{K}}(\sigma_1+\sigma_2)\widehat{\mathcal{S}}_{\mathrm{Q,sym},2}(\sigma_1,\sigma_2) \\
& + \frac{1}{2}\widehat{\mathcal{C}}(\sigma_1+\sigma_2)\widehat{\mathcal{K}}(\sigma_1+\sigma_2)^{-1} \\
& \times \Big(\partial_{s_1}\widehat{\mathcal{H}}(\sigma_1,\sigma_2)\big(\widehat{\mathcal{S}}_{\mathrm{Q,sym},1}(\sigma_1)\otimes\widehat{\mathcal{S}}_{\mathrm{Q,sym},1}(\sigma_2)\big) \\
& + \widehat{\mathcal{H}}(\sigma_1,\sigma_2)\big(\partial_{s_1}\widehat{\mathcal{S}}_{\mathrm{Q,sym},1}(\sigma_1)\otimes\widehat{\mathcal{S}}_{\mathrm{Q,sym},1}(\sigma_2)\big) \\
& + \partial_{s_2}\widehat{\mathcal{H}}(\sigma_2,\sigma_1)\big(\widehat{\mathcal{S}}_{\mathrm{Q,sym},1}(\sigma_2)\otimes\widehat{\mathcal{S}}_{\mathrm{Q,sym},1}(\sigma_1)\big) \\
& + \widehat{\mathcal{H}}(\sigma_2,\sigma_1)\big(\widehat{\mathcal{S}}_{\mathrm{Q,sym},1}(\sigma_2)\otimes\partial_{s_1}\widehat{\mathcal{S}}_{\mathrm{Q,sym},1}(\sigma_1)\big) \\
& + \partial_s\widehat{\mathcal{N}}(\sigma_1)\big(I_m\otimes\widehat{\mathcal{S}}_{\mathrm{Q,sym},1}(\sigma_1)\big) + \widehat{\mathcal{N}}(\sigma_1)\big(I_m\otimes\partial_{s_1}\widehat{\mathcal{S}}_{\mathrm{Q,sym},1}(\sigma_1)\big)\Big).
\end{aligned}$$

Interpolation of the first two terms in the sum follows directly using the projection approach from the previous proofs with the construction of V and W_1. Only the last term is left and split again into several parts to be considered independent of each other. Start with the terms involving the derivative of the bilinear or quadratic matrix functions, for which

$$\begin{aligned}
& \frac{1}{2}\widehat{\mathcal{C}}(\sigma_1+\sigma_2)\widehat{\mathcal{K}}(\sigma_1+\sigma_2)^{-1}\Big(\partial_{s_1}\widehat{\mathcal{H}}(\sigma_1,\sigma_2)\big(\widehat{\mathcal{S}}_{\mathrm{Q,sym},1}(\sigma_1)\otimes\widehat{\mathcal{S}}_{\mathrm{Q,sym},1}(\sigma_2)\big) \\
& \quad + \partial_{s_2}\widehat{\mathcal{H}}(\sigma_2,\sigma_1)\big(\widehat{\mathcal{S}}_{\mathrm{Q,sym},1}(\sigma_2)\otimes\widehat{\mathcal{S}}_{\mathrm{Q,sym},1}(\sigma_1)\big) + \partial_s\widehat{\mathcal{N}}(\sigma_1)\big(I_m\otimes\widehat{\mathcal{S}}_{\mathrm{Q,sym},1}(\sigma_1)\big)\Big) \\
& = \frac{1}{2}\mathcal{C}(\sigma_1+\sigma_2)V\widehat{\mathcal{K}}(\sigma_1+\sigma_2)^{-1}W^{\mathsf{H}}\Big(\partial_{s_1}\mathcal{H}(\sigma_1,\sigma_2)\big(\mathcal{S}_{\mathrm{Q,sym},1}(\sigma_1)\otimes\mathcal{S}_{\mathrm{Q,sym},1}(\sigma_2)\big) \\
& \quad + \partial_{s_2}\mathcal{H}(\sigma_2,\sigma_1)\big(\mathcal{S}_{\mathrm{Q,sym},1}(\sigma_2)\otimes\mathcal{S}_{\mathrm{Q,sym},1}(\sigma_1)\big) + \partial_s\mathcal{N}(\sigma_1)\big(I_m\otimes\widehat{\mathcal{S}}_{\mathrm{Q,sym},1}(\sigma_1)\big)\Big) \\
& = \frac{1}{2}\mathcal{C}(\sigma_1+\sigma_2)\mathcal{K}(\sigma_1+\sigma_2)^{-1}P_{\mathrm{W}}(\sigma_1+\sigma_2)^{\mathsf{H}}\Big(\partial_{s_1}\mathcal{H}(\sigma_1,\sigma_2)\big(\mathcal{S}_{\mathrm{Q,sym},1}(\sigma_1)\otimes\mathcal{S}_{\mathrm{Q,sym},1}(\sigma_2)\big) \\
& \quad + \partial_{s_2}\mathcal{H}(\sigma_2,\sigma_1)\big(\mathcal{S}_{\mathrm{Q,sym},1}(\sigma_2)\otimes\mathcal{S}_{\mathrm{Q,sym},1}(\sigma_1)\big) + \partial_s\mathcal{N}(\sigma_1)\big(I_m\otimes\widehat{\mathcal{S}}_{\mathrm{Q,sym},1}(\sigma_1)\big)\Big) \\
& = \frac{1}{2}\mathcal{C}(\sigma_1+\sigma_2)\mathcal{K}(\sigma_1+\sigma_2)^{-1}\Big(\partial_{s_1}\mathcal{H}(\sigma_1,\sigma_2)\big(\mathcal{S}_{\mathrm{Q,sym},1}(\sigma_1)\otimes\mathcal{S}_{\mathrm{Q,sym},1}(\sigma_2)\big) \\
& \quad + \partial_{s_2}\mathcal{H}(\sigma_2,\sigma_1)\big(\mathcal{S}_{\mathrm{Q,sym},1}(\sigma_2)\otimes\mathcal{S}_{\mathrm{Q,sym},1}(\sigma_1)\big) + \partial_s\mathcal{N}(\sigma_1)\big(I_m\otimes\widehat{\mathcal{S}}_{\mathrm{Q,sym},1}(\sigma_1)\big)\Big)
\end{aligned}$$

holds, using again the construction of W_1 as well as $V_{1,1}$ and $V_{1,2}$. Now, only the terms with derivatives in the Kronecker products are left. These are gathered into two groups

depending on the position of the partial derivative in the Kronecker product. Consider first

$$\widehat{Z}_1 := \frac{1}{2}\widehat{\mathcal{C}}(\sigma_1+\sigma_2)\widehat{\mathcal{K}}(\sigma_1+\sigma_2)^{-1}\Big(\widehat{\mathcal{H}}(\sigma_2,\sigma_1)\Big(\widehat{\mathcal{S}}_{\mathrm{Q,sym},1}(\sigma_2)\otimes\partial_{s_1}\widehat{\mathcal{S}}_{\mathrm{Q,sym},1}(\sigma_1)\Big) + \widehat{\mathcal{N}}(\sigma_1)\Big(I_m\otimes\partial_{s_1}\widehat{\mathcal{S}}_{\mathrm{Q,sym},1}(\sigma_1)\Big)\Big).$$

Instead of working directly on $\widehat{Z}_1$, this term is seen to be the 1-mode matricization of the tensor $\widehat{\boldsymbol{Z}}_1$ such that, equivalently, the conjugate 2-mode matricization can be considered instead. Together with the identity (2.2) it holds

$$\begin{aligned}
\overline{\widehat{\boldsymbol{Z}}_1^{(2)}} &= \frac{1}{2}\partial_{s_1}\widehat{\mathcal{S}}_{\mathrm{Q,sym},1}(\sigma_1)^{\mathsf{H}}\Big(\overline{\widehat{\boldsymbol{\mathcal{H}}}^{(2)}(\sigma_2,\sigma_1)}\Big(\overline{\widehat{\mathcal{S}}_{\mathrm{Q,sym},1}(\sigma_2)}\otimes\widehat{\mathcal{K}}(\sigma_1+\sigma_2)^{-\mathsf{H}}\widehat{\mathcal{C}}(\sigma_1+\sigma_2)^{\mathsf{H}}\Big)\\
&\quad+\overline{\widehat{\boldsymbol{\mathcal{N}}}^{(2)}(\sigma_1)}\Big(I_m\otimes\widehat{\mathcal{K}}(\sigma_1+\sigma_2)^{-\mathsf{H}}\widehat{\mathcal{C}}(\sigma_1+\sigma_2)^{\mathsf{H}}\Big)\Big)\\
&= \frac{1}{2}\partial_{s_1}\widehat{\mathcal{S}}_{\mathrm{Q,sym},1}(\sigma_1)^{\mathsf{H}}V^{\mathsf{H}}\Big(\overline{\boldsymbol{\mathcal{H}}^{(2)}(\sigma_2,\sigma_1)}\Big(\overline{\mathcal{S}_{\mathrm{Q,sym},1}(\sigma_2)}\otimes\mathcal{K}(\sigma_1+\sigma_2)^{-\mathsf{H}}\mathcal{C}(\sigma_1+\sigma_2)^{\mathsf{H}}\Big)\\
&\quad+\overline{\boldsymbol{\mathcal{N}}^{(2)}(\sigma_1)}\Big(I_m\otimes\mathcal{K}(\sigma_1+\sigma_2)^{-\mathsf{H}}\mathcal{C}(\sigma_1+\sigma_2)^{\mathsf{H}}\Big)\Big)\\
&= \frac{1}{2}\Big(\partial_s\widehat{\mathcal{B}}(\sigma_1)^{\mathsf{H}}-\widehat{\mathcal{B}}(\sigma_1)^{\mathsf{H}}\widehat{\mathcal{K}}(\sigma_1)^{-\mathsf{H}}\Big(\partial_s\widehat{\mathcal{K}}(\sigma_1)^{\mathsf{H}}\Big)\Big)\widehat{\mathcal{K}}(\sigma_1)^{-\mathsf{H}}V^{\mathsf{H}}\\
&\quad\times\Big(\overline{\boldsymbol{\mathcal{H}}^{(2)}(\sigma_2,\sigma_1)}(\overline{V_{1,2}}\otimes W_1)+\overline{\boldsymbol{\mathcal{N}}^{(2)}(\sigma_1)}(I_m\otimes W_1)\Big)\\
&= \frac{1}{2}\Big(\partial_s\mathcal{B}(\sigma_1)^{\mathsf{H}}-\mathcal{B}(\sigma_1)^{\mathsf{H}}\mathcal{K}(\sigma_1)^{-\mathsf{H}}\Big(\partial_s\mathcal{K}(\sigma_1)^{\mathsf{H}}\Big)\Big)P_{\mathrm{W}}(\sigma_1)W_2\\
&= \frac{1}{2}\partial_{s_1}\mathcal{S}_{\mathrm{Q,sym},1}(\sigma_1)^{\mathsf{H}}\Big(\overline{\boldsymbol{\mathcal{H}}^{(2)}(\sigma_2,\sigma_1)}\Big(\overline{\mathcal{S}_{\mathrm{Q,sym},1}(\sigma_2)}\otimes\mathcal{K}(\sigma_1+\sigma_2)^{-\mathsf{H}}\mathcal{C}(\sigma_1+\sigma_2)^{\mathsf{H}}\Big)\\
&\quad+\overline{\boldsymbol{\mathcal{N}}^{(2)}(\sigma_1)}\Big(I_m\otimes\mathcal{K}(\sigma_1+\sigma_2)^{-\mathsf{H}}\mathcal{C}(\sigma_1+\sigma_2)^{\mathsf{H}}\Big)\Big).
\end{aligned}$$

Thereby, for the 1-mode matricization it holds

$$\widehat{Z}_1 = \frac{1}{2}\mathcal{C}(\sigma_1+\sigma_2)\mathcal{K}(\sigma_1+\sigma_2)^{-1}\Big(\mathcal{H}(\sigma_2,\sigma_1)\Big(\mathcal{S}_{\mathrm{Q,sym},1}(\sigma_2)\otimes\partial_{s_1}\mathcal{S}_{\mathrm{Q,sym},1}(\sigma_1)\Big) + \mathcal{N}(\sigma_1)\Big(I_m\otimes\partial_{s_1}\mathcal{S}_{\mathrm{Q,sym},1}(\sigma_1)\Big)\Big).$$

In other words, the term with the reduced-order matrix functions is identical to the same expression using the full-order matrix functions. The only thing left is the term with the partial derivative at the first position in the Kronecker product

$$\widehat{Z}_2 := \frac{1}{2}\widehat{\mathcal{C}}(\sigma_1+\sigma_2)\widehat{\mathcal{K}}(\sigma_1+\sigma_2)^{-1}\widehat{\mathcal{H}}(\sigma_1,\sigma_2)\Big(\partial_{s_1}\widehat{\mathcal{S}}_{\mathrm{Q,sym},1}(\sigma_1)\otimes\widehat{\mathcal{S}}_{\mathrm{Q,sym},1}(\sigma_2)\Big),$$

which is again seen as the 1-mode matricization of another tensor $\widehat{\boldsymbol{Z}}_2$. Now, the 3-mode matricization of this tensor together with the identity (2.3) can be used to show analogously to $\widehat{Z}_1$ that

$$\begin{aligned}\overline{\widehat{\boldsymbol{Z}}_2^{(3)}} &= \frac{1}{2}\partial_{s_1}\widehat{\mathcal{S}}_{\mathrm{Q,sym},1}(\sigma_1)^{\mathsf{H}}\overline{\widehat{\boldsymbol{\mathcal{H}}}^{(3)}(\sigma_1,\sigma_2)}\Big(\overline{\widehat{\mathcal{S}}_{\mathrm{Q,sym},1}(\sigma_2)} \otimes \widehat{\mathcal{K}}(\sigma_1+\sigma_2)^{-\mathsf{H}}\widehat{\mathcal{C}}(\sigma_1+\sigma_2)^{\mathsf{H}}\Big)\\ &= \frac{1}{2}\partial_{s_1}\widehat{\mathcal{S}}_{\mathrm{Q,sym},1}(\sigma_1)^{\mathsf{H}}V^{\mathsf{H}}\overline{\boldsymbol{\mathcal{H}}^{(3)}(\sigma_1,\sigma_2)}\Big(\overline{V_{1,2}} \otimes W_1\Big)\\ &= \frac{1}{2}\partial_{s_1}\mathcal{S}_{\mathrm{Q,sym},1}(\sigma_1)^{\mathsf{H}}\overline{\boldsymbol{\mathcal{H}}^{(3)}(\sigma_1,\sigma_2)}\Big(\overline{\mathcal{S}_{\mathrm{Q,sym},1}(\sigma_2)} \otimes \mathcal{K}(\sigma_1+\sigma_2)^{-\mathsf{H}}\mathcal{C}(\sigma_1+\sigma_2)^{\mathsf{H}}\Big)\end{aligned}$$

holds, where the construction of V_2, W_1 and W_3 was used. Therefore, it also holds that

$$\widehat{Z}_2 = \frac{1}{2}\mathcal{C}(\sigma_1+\sigma_2)\mathcal{K}(\sigma_1+\sigma_2)^{-1}\mathcal{H}(\sigma_1,\sigma_2)\Big(\partial_{s_1}\mathcal{S}_{\mathrm{Q,sym},1}(\sigma_1) \otimes \mathcal{S}_{\mathrm{Q,sym},1}(\sigma_2)\Big),$$

which gives overall the desired Hermite interpolation result with respect to s_1. The interpolation of the partial derivative with respect to s_2 in Part (c) can be obtained the same way, where now instead of W_2 and W_3 the two matrices W_4 and W_5 are used. □

Theorem 6.5 shows the Hermite interpolation conditions in a separate fashion to point out the single basis contributions and their effects. It can be noted that if Theorem 6.5 Parts (b) and (c) hold, then the complete Jacobian is interpolated

$$\nabla\mathcal{G}_{\mathrm{Q,sym},2}(\sigma_1,\sigma_2) = \nabla\widehat{\mathcal{G}}_{\mathrm{Q,sym},2}(\sigma_1,\sigma_2).$$

The main difference of Theorem 6.5 to the two-sided projection results from the literature [30] are the additional matrices constructed for $\operatorname{span}(W)$. This comes from the facts that neither an underlying symmetric tensor for $\mathcal{H}$ was assumed nor the system was restricted to the SISO case. The following theorem considers exactly these special assumptions.

Theorem 6.6 (Implicit Hermite interpolation of sym. SISO TFs):
Let $\mathcal{G}_{\mathrm{Q}}$ be a quadratic-bilinear SISO system, described by its symmetric subsystem transfer functions $\mathcal{G}_{\mathrm{Q,sym},k}$ in (6.14) and (6.15), $\widehat{\mathcal{G}}_{\mathrm{Q}}$ the reduced-order quadratic-bilinear SISO system, constructed by (6.22) with its reduced-order symmetric transfer functions $\widehat{\mathcal{G}}_{\mathrm{Q,sym},k}$, and $\sigma \in \mathbb{C}$ an interpolation point such that the matrix functions $\mathcal{C}$, $\mathcal{K}^{-1}$, $\mathcal{B}$, $\mathcal{N}$, $\mathcal{H}$ and $\widehat{\mathcal{K}}^{-1}$ are complex differentiable in σ and 2σ. Also, let the tensor $\boldsymbol{\mathcal{H}}$, given by its 1-mode matricization $\boldsymbol{\mathcal{H}}^{(1)} = \mathcal{H}$, be symmetric. Construct V using

$$\begin{aligned} v_1 &= \mathcal{K}(\sigma)^{-1}\mathcal{B}(\sigma),\\ v_2 &= \mathcal{K}(2\sigma)^{-1}\Big(\mathcal{H}(\sigma,\sigma)(v_1 \otimes v_1) + \mathcal{N}(\sigma)v_1\Big),\\ \operatorname{span}(V) &\supseteq \operatorname{span}\left(\begin{bmatrix} v_1 & v_2 \end{bmatrix}\right), \end{aligned}$$

and W using

$$\begin{aligned} w_1 &= \mathcal{K}(2\sigma)^{-\mathsf{H}}\mathcal{C}(2\sigma)^{\mathsf{H}}, \\ w_2 &= \mathcal{K}(\sigma)^{-\mathsf{H}}\Big(\overline{\boldsymbol{\mathcal{H}}^{(2)}(\sigma,\sigma)}(\overline{v_1}\otimes w_1) + \frac{1}{2}\mathcal{N}(\sigma)^{\mathsf{H}}w_1\Big), \\ \operatorname{span}(W) &\supseteq \operatorname{span}\left(\begin{bmatrix} w_1 & w_2 \end{bmatrix}\right), \end{aligned}$$

and let V and W be of appropriate dimensions. Then the symmetric transfer functions of $\widehat{\mathcal{G}}_{\mathrm{Q}}$ interpolate those of $\mathcal{G}_{\mathrm{Q}}$ in the following way:

$$\begin{aligned} \mathcal{G}_{\mathrm{Q},\mathrm{sym},1}(\sigma) &= \widehat{\mathcal{G}}_{\mathrm{Q},\mathrm{sym},1}(\sigma), & \mathcal{G}_{\mathrm{Q},\mathrm{sym},1}(2\sigma) &= \widehat{\mathcal{G}}_{\mathrm{Q},\mathrm{sym},1}(2\sigma), \\ \mathcal{G}_{\mathrm{Q},\mathrm{sym},2}(\sigma,\sigma) &= \widehat{\mathcal{G}}_{\mathrm{Q},\mathrm{sym},2}(\sigma,\sigma), & \nabla\mathcal{G}_{\mathrm{Q},\mathrm{sym},2}(\sigma,\sigma) &= \nabla\widehat{\mathcal{G}}_{\mathrm{Q},\mathrm{sym},2}(\sigma,\sigma). \end{aligned} \qquad \lozenge$$

Proof. The simple interpolation conditions follow directly from Theorem 6.5 since these are independent of the special choice of interpolation points and the number of inputs and outputs of the system. The partial derivatives can be derived the same way as in the proof of Theorem 6.5. Consider first the partial derivative with respect to s_1 and collect the terms with partial derivative of $\widehat{\mathcal{S}}_{\mathrm{Q},\mathrm{sym},1}(\sigma)$ into

$$\begin{aligned} \widehat{Z} := \frac{1}{2}\widehat{\mathcal{C}}(2\sigma)\widehat{\mathcal{K}}(2\sigma)^{-1}\Big(&\mathcal{H}(\sigma,\sigma)\big(\partial_{s_1}\widehat{\mathcal{S}}_{\mathrm{Q},\mathrm{sym},1}(\sigma)\otimes\widehat{\mathcal{S}}_{\mathrm{Q},\mathrm{sym},1}(\sigma)\big) \\ &+ \mathcal{H}(\sigma,\sigma)\big(\widehat{\mathcal{S}}_{\mathrm{Q},\mathrm{sym},1}(\sigma)\otimes\partial_{s_1}\widehat{\mathcal{S}}_{\mathrm{Q},\mathrm{sym},1}(\sigma)\big) + \widehat{\mathcal{N}}(\sigma)\partial_{s_1}\widehat{\mathcal{S}}_{\mathrm{Q},\mathrm{sym},1}(\sigma)\Big). \end{aligned}$$

This expression can be simplified with the assumption of $\boldsymbol{\mathcal{H}}$ being a symmetric tensor and using (2.4) with the SISO system assumption such that

$$\widehat{Z} = \widehat{\mathcal{C}}(2\sigma)\widehat{\mathcal{K}}(2\sigma)^{-1}\Big(\widehat{\mathcal{H}}(\sigma,\sigma)\big(\widehat{\mathcal{S}}_{\mathrm{Q},\mathrm{sym},1}(\sigma)\otimes\partial_{s_1}\widehat{\mathcal{S}}_{\mathrm{Q},\mathrm{sym},1}(\sigma)\big) + \frac{1}{2}\widehat{\mathcal{N}}(\sigma)\partial_{s_1}\widehat{\mathcal{S}}_{\mathrm{Q},\mathrm{sym},1}(\sigma)\Big)$$

holds. Considering now the underlying tensor $\widehat{\boldsymbol{Z}}^{(1)} = \widehat{Z}$ and its 2-mode matricization, together with (2.2) it holds

$$\begin{aligned} \overline{\widehat{\boldsymbol{Z}}^{(2)}} &= \partial_{s_1}\widehat{\mathcal{S}}_{\mathrm{Q},\mathrm{sym},1}(\sigma)^{\mathsf{H}}\Big(\overline{\widehat{\boldsymbol{\mathcal{H}}}^{(2)}(\sigma,\sigma)}\big(\overline{\widehat{\mathcal{S}}_{\mathrm{Q},\mathrm{sym},1}(\sigma)}\otimes\widehat{\mathcal{K}}(2\sigma)^{-\mathsf{H}}\widehat{\mathcal{C}}(2\sigma)^{\mathsf{H}}\big) \\ &\quad + \frac{1}{2}\widehat{\mathcal{N}}(\sigma)^{\mathsf{H}}\widehat{\mathcal{K}}(2\sigma)^{-\mathsf{H}}\widehat{\mathcal{C}}(2\sigma)^{\mathsf{H}}\Big) \\ &= \partial_{s_1}\widehat{\mathcal{S}}_{\mathrm{Q},\mathrm{sym},1}(\sigma)^{\mathsf{H}}V^{\mathsf{H}}\Big(\overline{\boldsymbol{\mathcal{H}}^{(2)}(\sigma,\sigma)}\big(\overline{V_1}\otimes w_1\big) + \frac{1}{2}\mathcal{N}(\sigma)^{\mathsf{H}}w_1\Big) \\ &= \Big(\partial_s\mathcal{B}(\sigma)^{\mathsf{H}} - \mathcal{B}(\sigma)^{\mathsf{H}}\mathcal{K}(\sigma)^{-\mathsf{H}}\big(\partial_s\mathcal{K}(\sigma)^{\mathsf{H}}\big)\Big)P_{\mathrm{W}}(\sigma)w_2 \\ &= \partial_{s_1}\mathcal{S}_{\mathrm{Q},\mathrm{sym},1}(\sigma)^{\mathsf{H}}\Big(\overline{\boldsymbol{\mathcal{H}}^{(2)}(\sigma,\sigma)}\big(\overline{\mathcal{S}_{\mathrm{Q},\mathrm{sym},1}(\sigma)}\otimes\mathcal{K}(2\sigma)^{-\mathsf{H}}\mathcal{C}(2\sigma)^{\mathsf{H}}\big) \\ &\quad + \frac{1}{2}\mathcal{N}(\sigma)^{\mathsf{H}}\mathcal{K}(2\sigma)^{-\mathsf{H}}\mathcal{C}(2\sigma)^{\mathsf{H}}\Big), \end{aligned}$$

which yields the desired interpolation of $\partial_{s_1}\mathcal{G}_{\mathrm{Q,sym},2}(\sigma,\sigma)$. Due to the symmetry of the transfer function with respect to the frequency arguments, the exact same expression Z can be used to proof the interpolation of the partial derivative with respect to s_2, which gives the interpolation of the full Jacobi matrix. □

6.4.3 Interpolating structured regular subsystem transfer functions

The regular transfer functions for quadratic-bilinear systems were developed to decrease the number of terms evaluated in the frequency domain representation. Therefore, one could expect a similar reduction of frequency-dependent terms and, consequently, of the projection space dimensions in the corresponding structured interpolation approach. The following theorem is a translation of Theorem 6.2 to the case of regular transfer functions.

Theorem 6.7 (Regular transfer function interpolation via V):
Let $\mathcal{G}_{\mathrm{Q}}$ be a quadratic-bilinear system, described by its regular subsystem transfer functions $\mathcal{G}_{\mathrm{Q,reg},k}$ in (6.18) and (6.19), and $\widehat{\mathcal{G}}_{\mathrm{Q}}$ the reduced-order quadratic-bilinear system constructed by (6.22), with its reduced-order regular transfer functions $\widehat{\mathcal{G}}_{\mathrm{Q,reg},k}$. Also, let $\sigma_1, \ldots, \sigma_k \in \mathbb{C}$ be a set of interpolation points such that the matrix functions $\mathcal{C}$, $\mathcal{K}^{-1}$, $\mathcal{B}$, $\mathcal{N}$, $\mathcal{H}$ and $\widehat{\mathcal{K}}^{-1}$ are defined in these points and differences between them. Construct V using

$$\begin{aligned} V_j &= \begin{bmatrix} \mathcal{S}_{\mathrm{Q,reg},j}(\sigma_1,\ldots\sigma_j) & \mathcal{S}_{\mathrm{Q,reg},j}(\sigma_{1+i_j}-\sigma_{i_j},\ldots,\sigma_{j+i_j}-\sigma_{i_j}) \end{bmatrix}, \\ \operatorname{span}(V) &\supseteq \operatorname{span}\left(\begin{bmatrix} V_1 & \ldots & V_k \end{bmatrix} \right), \end{aligned}$$

with $1 \leq j \leq k$; $1 \leq i_j \leq k-j$ and with the recursive terms from (6.19), and let W be an arbitrary full-rank truncation matrix of appropriate dimension. Then the regular transfer functions of $\widehat{\mathcal{G}}_{\mathrm{Q}}$ interpolate those of $\mathcal{G}_{\mathrm{Q}}$ in the following way:

$$\begin{aligned} \mathcal{G}_{\mathrm{Q,reg},j}(\sigma_1,\ldots\sigma_j) &= \widehat{\mathcal{G}}_{\mathrm{Q,reg},j}(\sigma_1,\ldots\sigma_j), \\ \mathcal{G}_{\mathrm{Q,reg},j}(\sigma_{1+i_j}-\sigma_{i_j},\ldots,\sigma_{j+i_j}-\sigma_{i_j}) &= \widehat{\mathcal{G}}_{\mathrm{Q,reg},j}(\sigma_{1+i_j}-\sigma_{i_j},\ldots,\sigma_{j+i_j}-\sigma_{i_j}), \end{aligned}$$

for $1 \leq j \leq k$ and $1 \leq i_j \leq k-j$. ◊

In the construction of the projection space in Theorem 6.7 a special notation is used for the concatenation of several recursive terms. Therefore, an illustrative example is shown first before continuing with the proof of Theorem 6.7. Consider the interpolation of $\mathcal{G}_{\mathrm{Q,reg},4}$ in the interpolation points σ_1, σ_2, σ_3, σ_4. The constructed matrices for the

projection space are then given by

$$\begin{aligned}
V_1 &= \Big[\mathcal{S}_{\mathrm{Q,reg},1}(\sigma_1),\ \mathcal{S}_{\mathrm{Q,reg},1}(\sigma_2-\sigma_1),\ \mathcal{S}_{\mathrm{Q,reg},1}(\sigma_3-\sigma_2),\ \mathcal{S}_{\mathrm{Q,reg},1}(\sigma_4-\sigma_3)\Big],\\
V_2 &= \Big[\mathcal{S}_{\mathrm{Q,reg},2}(\sigma_1,\sigma_2),\ \mathcal{S}_{\mathrm{Q,reg},2}(\sigma_2-\sigma_1,\sigma_3-\sigma_1),\ \mathcal{S}_{\mathrm{Q,reg},2}(\sigma_3-\sigma_2,\sigma_4-\sigma_2)\Big],\\
V_3 &= \Big[\mathcal{S}_{\mathrm{Q,reg},3}(\sigma_1,\sigma_2,\sigma_3),\ \mathcal{S}_{\mathrm{Q,reg},3}(\sigma_2-\sigma_1,\sigma_3-\sigma_1,\sigma_4-\sigma_1)\Big],\\
V_4 &= \mathcal{S}_{\mathrm{Q,reg},4}(\sigma_1,\sigma_2,\sigma_3,\sigma_4),
\end{aligned}$$

which yields the interpolation of

$$\begin{aligned}
&\mathcal{G}_{\mathrm{Q,reg},1}(\sigma_1),\ \mathcal{G}_{\mathrm{Q,reg},1}(\sigma_2-\sigma_1),\ \mathcal{G}_{\mathrm{Q,reg},1}(\sigma_3-\sigma_2),\ \mathcal{G}_{\mathrm{Q,reg},1}(\sigma_4-\sigma_3),\\
&\mathcal{G}_{\mathrm{Q,reg},2}(\sigma_1,\sigma_2),\ \mathcal{G}_{\mathrm{Q,reg},2}(\sigma_2-\sigma_1,\sigma_3-\sigma_1),\ \mathcal{G}_{\mathrm{Q,reg},2}(\sigma_3-\sigma_2,\sigma_4-\sigma_2),\\
&\mathcal{G}_{\mathrm{Q,reg},3}(\sigma_1,\sigma_2,\sigma_3),\ \mathcal{G}_{\mathrm{Q,reg},3}(\sigma_2-\sigma_1,\sigma_3-\sigma_1,\sigma_4-\sigma_1),\\
&\mathcal{G}_{\mathrm{Q,reg},4}(\sigma_1,\sigma_2,\sigma_3,\sigma_4).
\end{aligned}$$

Proof of Theorem 6.7. As in previous proofs, it is enough to show the interpolation of the second regular subsystem transfer function as the rest follows via induction over the transfer function level k. For simplicity, only the interpolation in the point (σ_1,σ_2) is considered. The other combinations of interpolation points follow analogously. It holds

$$\begin{aligned}
\widehat{\mathcal{G}}_{\mathrm{Q,reg},2}(\sigma_1,\sigma_2) &= \widehat{\mathcal{C}}(\sigma_2)\widehat{\mathcal{K}}(\sigma_2)^{-1}\Big(\widehat{\mathcal{H}}(\sigma_2-\sigma_1,\sigma_1)\big(\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma_2-\sigma_1)\otimes\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma_1)\big)\\
&\quad+\widehat{\mathcal{N}}(\sigma_1)\big(I_m\otimes\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma_1)\big)\Big)\\
&= \widehat{\mathcal{C}}(\sigma_2)\widehat{\mathcal{K}}(\sigma_2)^{-1}\Big(W^{\mathsf{H}}\mathcal{H}(\sigma_2-\sigma_1,\sigma_1)\big(\mathcal{S}_{\mathrm{Q,reg},1}(s_2-\sigma_1)\otimes\mathcal{S}_{\mathrm{Q,reg},1}(\sigma_1)\big)\\
&\quad+W^{\mathsf{H}}\mathcal{N}(\sigma_1)\big(I_m\otimes\mathcal{S}_{\mathrm{Q,reg},1}(\sigma_1)\big)\Big)\\
&= \mathcal{C}(\sigma_2)\underbrace{V\widehat{\mathcal{K}}(\sigma_2)^{-1}W^{\mathsf{H}}\mathcal{K}(\sigma_2)}_{=P_{\mathrm{V}}(\sigma_2)}\mathcal{K}(\sigma_2)^{-1}\\
&\quad\times\Big(\mathcal{H}(\sigma_2-\sigma_1,\sigma_1)\big(\mathcal{S}_{\mathrm{Q,reg},1}(\sigma_2-\sigma_1)\otimes\mathcal{S}_{\mathrm{Q,reg},1}(\sigma_1)\big)\\
&\quad+\mathcal{N}(\sigma_1)\big(I_m\otimes\mathcal{S}_{\mathrm{Q,reg},1}(\sigma_1)\big)\Big)\\
&= \mathcal{C}(\sigma_2)P_{\mathrm{V}}(\sigma_2)\mathcal{S}_{\mathrm{Q,reg},2}(\sigma_1,\sigma_2)\\
&= \mathcal{G}_{\mathrm{Q,reg},2}(\sigma_1,\sigma_2),
\end{aligned}$$

where P_{V} is the projector onto $\operatorname{span}(V)$ from (3.24) and using

$$\operatorname{span}\big(\mathcal{S}_{\mathrm{Q,reg},2}(\sigma_1,\sigma_2)\big)\subseteq\operatorname{span}(V).$$

The same ideas are then used to show the interpolation in the differences of the interpolation points $(\sigma_2-\sigma_1,\sigma_3-\sigma_1),\ldots,(\sigma_{k-1}-\sigma_{k-2},\sigma_k-\sigma_{k-2})$ and the complete result follows via induction over k. □

Counting the computed vectors and interpolation conditions shows that the matrix V_j contributes $(k-j+1)m^j$ columns to the projection space construction, i.e., for the interpolation of the k-th regular subsystem transfer function

$$\dim\Big(\operatorname{span}(V)\Big) \geq \sum_{j=1}^{k}(k-j+1)m^j \tag{6.27}$$

holds, if all computed vectors are linearly independent. This dimension (6.27) is significantly smaller than (6.24). It becomes easier to see in the case of SISO systems, since (6.27) simplifies to

$$\dim\Big(\operatorname{span}(V)\Big) \geq \sum_{j=1}^{k}(k-j+1) = \sum_{j=1}^{k} j = \frac{k(k+1)}{2}. \tag{6.28}$$

Therefore, the number of frequency-dependent terms in the regular transfer function case is only growing quadratically (6.28) with the transfer function level in contrast to the exponential growth in the symmetric transfer function case (6.25). On the other hand, due to the less compute columns for the projection space, also less interpolation conditions are matched.

Similar to the symmetric case, in practice, only the first two regular subsystem transfer functions are of actual interest. Therefore, the following corollary states a simplified version of Theorem 6.7 for this special case.

Corollary 6.8 (Simplified regular transfer function interpolation):
Let $\mathcal{G}_{\mathrm{Q}}$ be a quadratic-bilinear system, described by its regular subsystem transfer functions $\mathcal{G}_{\mathrm{Q},\mathrm{reg},k}$ in (6.18) and (6.19), and $\widehat{\mathcal{G}}_{\mathrm{Q}}$ the reduced-order quadratic-bilinear system constructed by (6.22), with its reduced-order regular transfer functions $\widehat{\mathcal{G}}_{\mathrm{Q},\mathrm{reg},k}$. Also, let $\sigma_1, \sigma_2 \in \mathbb{C}$ be two interpolation points such that the matrix functions $\mathcal{C}$, $\mathcal{K}^{-1}$, $\mathcal{B}$, $\mathcal{N}$, $\mathcal{H}$ and $\widehat{\mathcal{K}}^{-1}$ are defined in these points and in $\sigma_2 - \sigma_1$. Construct V using

$$\begin{aligned}
V_{1,1} &= \mathcal{K}(\sigma_1)^{-1}\mathcal{B}(\sigma_1),\\
V_{1,2} &= \mathcal{K}(\sigma_2-\sigma_1)^{-1}\mathcal{B}(\sigma_2-\sigma_1),\\
V_2 &= \mathcal{C}(\sigma_2)\mathcal{K}(\sigma_2)^{-1}\Big(\mathcal{H}(\sigma_2-\sigma_1,\sigma_1)(V_{1,2}\otimes V_{1,1}) + \mathcal{N}(\sigma_1)(I_m \otimes V_{1,1})\Big),\\
\operatorname{span}(V) &\supseteq \operatorname{span}\left(\begin{bmatrix} V_{1,1} & V_{1,2} & V_2 \end{bmatrix}\right),
\end{aligned}$$

and let W be an arbitrary full-rank truncation matrix of appropriate dimension. Then the regular transfer functions of $\widehat{\mathcal{G}}_{\mathrm{Q}}$ interpolate those of $\mathcal{G}_{\mathrm{Q}}$ in the following way:

$$\begin{aligned}
\mathcal{G}_{\mathrm{Q},\mathrm{reg},1}(\sigma_1) &= \widehat{\mathcal{G}}_{\mathrm{Q},\mathrm{reg},1}(\sigma_1),\\
\mathcal{G}_{\mathrm{Q},\mathrm{reg},1}(\sigma_2-\sigma_1) &= \widehat{\mathcal{G}}_{\mathrm{Q},\mathrm{reg},1}(\sigma_2-\sigma_1),\\
\mathcal{G}_{\mathrm{Q},\mathrm{reg},2}(\sigma_1,\sigma_2) &= \widehat{\mathcal{G}}_{\mathrm{Q},\mathrm{reg},1}(\sigma_1,\sigma_2).
\end{aligned}$$

◊

While the growth of the number of computed terms for the interpolation of regular and symmetric transfer functions are in principle completely different, see (6.24) and (6.27), this only holds for higher-level transfer functions ($k \geq 3$). Corollaries 6.3 and 6.8 are both computing 3 terms for the interpolation of the second symmetric and regular subsystem transfer functions.

As in the symmetric case, for the regular subsystem transfer functions the choice of interpolation points can be used to reduce the dimension of the projection space significantly. In contrast to the previous section, the interpolation points need to be chosen to be integer multiples of each other:

$$\sigma_1 = \sigma, \quad \sigma_2 = 2\sigma, \quad \ldots, \quad \sigma_k = k\sigma, \tag{6.29}$$

for a single interpolation point $\sigma \in \mathbb{C}$. This is also the choice made in [4] for unstructured regular transfer functions. Using this particular choice of interpolation points reduces the dimension of the projection space (6.27) necessary for the interpolation of the k-th regular subsystem transfer function to $\sum_{j=1}^{k} m^j$. In terms of the previous example, the constructed matrices can be reduced to

$$\begin{aligned} V_1 &= \mathcal{S}_{\mathrm{Q,reg},1}(\sigma), & V_2 &= \mathcal{S}_{\mathrm{Q,reg},2}(\sigma, 2\sigma), \\ V_3 &= \mathcal{S}_{\mathrm{Q,reg},3}(\sigma, 2\sigma, 3\sigma), & V_4 &= \mathcal{S}_{\mathrm{Q,reg},4}(\sigma, 2\sigma, 3\sigma, 4\sigma), \end{aligned}$$

since all other matrices are identical to these four and, thereby, span the same subspace. The cost for the reduction of the dimension of the projection space comes with the decrease of the number of matched interpolation conditions. In this example, only four matched interpolation conditions are left

$$\mathcal{G}_{\mathrm{Q,reg},1}(\sigma), \quad \mathcal{G}_{\mathrm{Q,reg},2}(\sigma, 2\sigma), \quad \mathcal{G}_{\mathrm{Q,reg},3}(\sigma, 2\sigma, 3\sigma), \quad \mathcal{G}_{\mathrm{Q,reg},4}(\sigma, 2\sigma, 3\sigma, 4\sigma).$$

Remark 6.9 (Interpolation point selection in the purely bilinear case):
The suggested choice of interpolation points (6.29) for regular subsystem transfer functions to reduce the number of computed terms differs significantly from the default suggestion in case of regular bilinear transfer functions; see Chapter 5. There, the interpolation points for higher regular subsystem transfer functions are proposed to be chosen as for symmetric transfer functions

$$\sigma_1 = \ldots = \sigma_k = \sigma.$$

This choice is still possible and efficient for quadratic-bilinear systems, but comes with the cost of forced interpolation in 0 due to the difference of the interpolation points needed in the quadratic terms. Therefore, the matrix functions need to exist in 0. Also, setting one of the frequency arguments in $\mathcal{H}$ permanently to 0 easily hides information of the quadratic term, e.g., in (6.23), H_{vp} and H_{vv} are always multiplied with 0 and never taken into account for the projection space construction. ◊

The next lemma states an equivalent to Lemma 6.4 for the interpolation of the second regular subsystem transfer functions without evaluating the nonlinear terms.

Lemma 6.10 (Implicit regular transfer function interpolation):
Let $\mathcal{G}_{\mathrm{Q}}$ be a quadratic-bilinear system, described by its regular subsystem transfer functions $\mathcal{G}_{\mathrm{Q,reg},k}$ in (6.18) and (6.19), $\widehat{\mathcal{G}}_{\mathrm{Q}}$ the reduced-order quadratic-bilinear system constructed by (6.22), with its reduced-order regular transfer functions $\widehat{\mathcal{G}}_{\mathrm{Q,reg},k}$, and $\sigma_1, \sigma_2 \in \mathbb{C}$ two interpolation points such that the matrix functions $\mathcal{C}$, $\mathcal{K}^{-1}$, $\mathcal{B}$, $\mathcal{N}$, $\mathcal{H}$ and $\widehat{\mathcal{K}}^{-1}$ are defined in these points and their difference $\sigma_2 - \sigma_1$. Construct V using

$$\operatorname{span}(V) \supseteq \operatorname{span}\left(\begin{bmatrix}\mathcal{K}(\sigma_1)^{-1}\mathcal{B}(\sigma_1) & \mathcal{K}(\sigma_2-\sigma_1)^{-1}\mathcal{B}(\sigma_2-\sigma_1)\end{bmatrix}\right),$$

and W using

$$\operatorname{span}(W) \supseteq \operatorname{span}\left(\mathcal{K}(\sigma_2)^{-\mathsf{H}}\mathcal{C}(\sigma_2)^{\mathsf{H}}\right),$$

and let the two matrices V and W be of appropriate dimensions. Then the regular transfer functions of $\widehat{\mathcal{G}}_{\mathrm{Q}}$ interpolate those of $\mathcal{G}_{\mathrm{Q}}$ in the following way:

$$\begin{aligned}\mathcal{G}_{\mathrm{Q,reg},1}(\sigma_1) &= \widehat{\mathcal{G}}_{\mathrm{Q,reg},1}(\sigma_1), & \mathcal{G}_{\mathrm{Q,reg},1}(\sigma_2) &= \widehat{\mathcal{G}}_{\mathrm{Q,reg},1}(\sigma_2),\\ \mathcal{G}_{\mathrm{Q,reg},1}(\sigma_2-\sigma_1) &= \widehat{\mathcal{G}}_{\mathrm{Q,reg},1}(\sigma_2-\sigma_1), & \mathcal{G}_{\mathrm{Q,reg},2}(\sigma_1,\sigma_2) &= \widehat{\mathcal{G}}_{\mathrm{Q,reg},2}(\sigma_1,\sigma_2).\end{aligned} \qquad \lozenge$$

Proof. The three interpolation conditions for the first subsystem transfer function follow directly from Corollary 6.8 and Proposition 3.2. Left to be proven is the interpolation of the second transfer function level via the two-sided projection. It holds

$$\begin{aligned}\widehat{\mathcal{G}}_{\mathrm{Q,reg},2}(\sigma_1,\sigma_2) &= \widehat{\mathcal{C}}(\sigma_2)\widehat{\mathcal{K}}(\sigma_2)^{-1}\Big(\widehat{\mathcal{H}}(\sigma_2-\sigma_1,\sigma_1)\big(\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma_2-\sigma_1)\otimes\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma_1)\big)\\ &\quad + \widehat{\mathcal{N}}(\sigma_1)\big(I_m\otimes\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma_1)\big)\Big)\\ &= \widehat{\mathcal{C}}(\sigma_2)\widehat{\mathcal{K}}(\sigma_2)^{-1}W^{\mathsf{H}}\Big(\mathcal{H}(\sigma_2-\sigma_1,\sigma_1)\big(\mathcal{S}_{\mathrm{Q,reg},1}(\sigma_2-\sigma_1)\otimes\mathcal{S}_{\mathrm{Q,reg},1}(\sigma_1)\big)\\ &\quad + \mathcal{N}(\sigma_1)\big(I_m\otimes\mathcal{S}_{\mathrm{Q,reg},1}(\sigma_1)\big)\Big)\\ &= \mathcal{C}(\sigma_2)\mathcal{K}(\sigma_2)^{-1}\underbrace{\mathcal{K}(\sigma_2)V\widehat{\mathcal{K}}(\sigma_2)^{-1}W^{\mathsf{H}}}_{=P_{\mathrm{W}}^{\mathsf{H}}(\sigma_2)}\\ &\quad \times\Big(\mathcal{H}(\sigma_2-\sigma_1,\sigma_1)\big(\mathcal{S}_{\mathrm{Q,reg},1}(\sigma_2-\sigma_1)\otimes\mathcal{S}_{\mathrm{Q,reg},1}(\sigma_1)\big)\\ &\quad + \mathcal{N}(\sigma_1)\big(I_m\otimes\mathcal{S}_{\mathrm{Q,reg},1}(\sigma_1)\big)\Big)\\ &= \mathcal{G}_{\mathrm{Q,reg},2}(\sigma_1,\sigma_2)\end{aligned}$$

with P_{W} the projector (3.25) onto $\operatorname{span}(W)$. □

A generalization of Lemma 6.10 for higher transfer function levels in the sense of Theorem 6.7 is omitted here due to its similarity to the more practical result of Lemma 6.10.

Making further use of the two-sided projection allows to implicitly interpolate not only the regular subsystem transfer functions but also their partial derivatives. These results are given in the following theorem.

Theorem 6.11 (Implicit Hermite interpolation of reg. transfer functions): Let $\mathcal{G}_{\mathrm{Q}}$ be a quadratic-bilinear system, described by its regular subsystem transfer functions $\mathcal{G}_{\mathrm{Q},\mathrm{reg},k}$ in (6.18) and (6.19), $\widehat{\mathcal{G}}_{\mathrm{Q}}$ the reduced-order quadratic-bilinear system constructed by (6.22), with its reduced-order regular transfer functions $\widehat{\mathcal{G}}_{\mathrm{Q},\mathrm{reg},k}$, and $\sigma_1, \sigma_2 \in \mathbb{C}$ two interpolation points such that the matrix functions $\mathcal{C}$, $\mathcal{K}^{-1}$, $\mathcal{B}$, $\mathcal{N}$, $\mathcal{H}$ and $\widehat{\mathcal{K}}^{-1}$ are complex differentiable in these points and their difference $\sigma_2 - \sigma_1$. Let the following matrices be given

$$\begin{aligned} V_{1,1} &= \mathcal{K}(\sigma_1)^{-1}\mathcal{B}(\sigma_1), \\ V_{1,2} &= \mathcal{K}(\sigma_2 - \sigma_1)^{-1}\mathcal{B}(\sigma_2 - \sigma_1), \\ V_2 &= \mathcal{K}(\sigma_2)^{-1}\Big(\mathcal{H}(\sigma_2 - \sigma_1, \sigma_1)(V_{1,2} \otimes V_{1,1}) + \mathcal{N}(\sigma_1)(I_m \otimes V_{1,1})\Big), \end{aligned}$$

and, also,

$$\begin{aligned} W_1 &= \mathcal{K}(\sigma_2)^{-\mathsf{H}}\mathcal{C}(\sigma_2)^{\mathsf{H}}, \\ W_2 &= \mathcal{K}(\sigma_1)^{-\mathsf{H}}\Big(\overline{\boldsymbol{\mathcal{H}}^{(2)}(\sigma_2 - \sigma_1, \sigma_1)}(\overline{V_{1,2}} \otimes W_1) + \overline{\boldsymbol{\mathcal{N}}^{(2)}(\sigma_1)}(I_m \otimes W_1)\Big), \\ W_3 &= \mathcal{K}(\sigma_2 - \sigma_1)^{-\mathsf{H}}\overline{\boldsymbol{\mathcal{H}}^{(3)}(\sigma_2 - \sigma_1, \sigma_1)}(\overline{V_{1,1}} \otimes W_1), \end{aligned}$$

where $\boldsymbol{\mathcal{H}}^{(2)}, \boldsymbol{\mathcal{H}}^{(3)}$ are the 2- and 3-mode matricizations of the tensor corresponding to the quadratic term such that $\boldsymbol{\mathcal{H}}^{(1)} = \mathcal{H}$, and $\boldsymbol{\mathcal{N}}^{(2)}$ is the 2-mode matricization of the tensor corresponding to the bilinear term such that $\boldsymbol{\mathcal{N}}^{(1)} = \mathcal{N}$. Then the following statements hold:

(a) If V and W are constructed such that

$$\operatorname{span}(V) \supseteq \operatorname{span}\left(\begin{bmatrix} V_{1,1} & V_{1,2} \end{bmatrix}\right) \quad \text{and} \quad \operatorname{span}(W) \supseteq \operatorname{span}(W_1),$$

then the following interpolation conditions hold:

$$\begin{aligned} \mathcal{G}_{\mathrm{Q},\mathrm{reg},1}(\sigma_1) &= \widehat{\mathcal{G}}_{\mathrm{Q},\mathrm{reg},1}(\sigma_1), & \mathcal{G}_{\mathrm{Q},\mathrm{reg},1}(\sigma_2) &= \widehat{\mathcal{G}}_{\mathrm{Q},\mathrm{reg},1}(\sigma_2), \\ \mathcal{G}_{\mathrm{Q},\mathrm{reg},1}(\sigma_2 - \sigma_1) &= \widehat{\mathcal{G}}_{\mathrm{Q},\mathrm{reg},1}(\sigma_2 - \sigma_1), & \mathcal{G}_{\mathrm{Q},\mathrm{reg},2}(\sigma_1, \sigma_2) &= \widehat{\mathcal{G}}_{\mathrm{Q},\mathrm{reg},2}(\sigma_1, \sigma_2). \end{aligned} \tag{6.30}$$

(b) If V and W are constructed such that

$$\operatorname{span}(V) \supseteq \operatorname{span}\left(\begin{bmatrix} V_{1,1} & V_{1,2} \end{bmatrix}\right) \quad \text{and} \quad \operatorname{span}(W) \supseteq \operatorname{span}\left(\begin{bmatrix} W_1 & W_2 & W_3 \end{bmatrix}\right),$$

then, additionally to (6.30), the following Hermite interpolation condition holds:

$$\partial_{s_1}\mathcal{G}_{\mathrm{Q},\mathrm{reg},2}(\sigma_1, \sigma_2) = \partial_{s_1}\widehat{\mathcal{G}}_{\mathrm{Q},\mathrm{reg},2}(\sigma_1, \sigma_2).$$

(c) If V and W are constructed such that

$$\operatorname{span}(V) \supseteq \operatorname{span}\left(\begin{bmatrix} V_{1,1} & V_{1,2} & V_2 \end{bmatrix}\right) \quad \text{and} \quad \operatorname{span}(W) \supseteq \operatorname{span}\left(\begin{bmatrix} W_1 & W_3 \end{bmatrix}\right),$$

then, additionally to (6.30), the following Hermite interpolation condition holds:

$$\partial_{s_2} \mathcal{G}_{\mathrm{Q,reg},2}(\sigma_1, \sigma_2) = \partial_{s_2} \widehat{\mathcal{G}}_{\mathrm{Q,reg},2}(\sigma_1, \sigma_2). \qquad \Diamond$$

Proof. Part (a) of the theorem is only resuming the results of Lemma 6.10 with the construction of $V_{1,1}$, $V_{1,2}$ and W_1.

To prove Part (b), consider the partial derivative of the second reduced-order regular subsystem transfer function in the interpolation points, which is given by

$$\begin{aligned}
\partial_{s_1} \widehat{\mathcal{G}}_{\mathrm{Q,reg},2}(\sigma_1, \sigma_2) = \widehat{\mathcal{C}}(\sigma_2)\widehat{\mathcal{K}}(\sigma_2)^{-1}\Big(&\Big(\partial_{s_2}\widehat{\mathcal{H}}(\sigma_2 - \sigma_1, \sigma_1) - \partial_{s_1}\widehat{\mathcal{H}}(\sigma_2 - \sigma_1, \sigma_1)\Big) \\
&\times \Big(\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma_2 - \sigma_1) \otimes \widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma_1)\Big) \\
&- \widehat{\mathcal{H}}(\sigma_2 - \sigma_1, \sigma_1)\Big(\partial_{s_1}\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma_2 - \sigma_1) \otimes \widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma_1)\Big) \\
&+ \widehat{\mathcal{H}}(\sigma_2 - \sigma_1, \sigma_1)\Big(\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma_2 - \sigma_1) \otimes \partial_{s_1}\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma_1)\Big) \\
&+ \partial_s\widehat{\mathcal{N}}(\sigma_1)\Big(I_m \otimes \widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma_1)\Big) + \widehat{\mathcal{N}}(\sigma_1)\Big(I_m \otimes \partial_{s_1}\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma_1)\Big)\Big).
\end{aligned}$$

The terms above need to be grouped according to the occurrence of partial derivatives in the Kronecker products and also with respect to the position of the partial derivatives in the Kronecker products. For the terms with no derivatives in the Kronecker products, it can be shown that

$$\begin{aligned}
&\widehat{\mathcal{C}}(\sigma_2)\widehat{\mathcal{K}}(\sigma_2)^{-1}\Big(\Big(\partial_{s_2}\widehat{\mathcal{H}}(\sigma_2 - \sigma_1, \sigma_1) - \partial_{s_1}\widehat{\mathcal{H}}(\sigma_2 - \sigma_1, \sigma_1)\Big) \\
&\quad \times \Big(\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma_2 - \sigma_1) \otimes \widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma_1)\Big) + \partial_s\widehat{\mathcal{N}}(\sigma_1)\Big(I_m \otimes \widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma_1)\Big)\Big) \\
&= \widehat{\mathcal{C}}(\sigma_2)\widehat{\mathcal{K}}(\sigma_2)^{-1}W^{\mathsf{H}}\Big(\Big(\partial_{s_2}\mathcal{H}(\sigma_2 - \sigma_1, \sigma_1) - \partial_{s_1}\mathcal{H}(\sigma_2 - \sigma_1, \sigma_1)\Big) \\
&\quad \times \Big(V\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma_2 - \sigma_1) \otimes V\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma_1)\Big) + \partial_s\mathcal{N}(\sigma_1)\Big(I_m \otimes V\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma_1)\Big)\Big) \\
&= \mathcal{C}(\sigma_2)\mathcal{K}(\sigma_2)^{-1}\Big(\Big(\partial_{s_2}\mathcal{H}(\sigma_2 - \sigma_1, \sigma_1) - \partial_{s_1}\mathcal{H}(\sigma_2 - \sigma_1, \sigma_1)\Big) \\
&\quad \times \Big(\mathcal{S}_{\mathrm{Q,reg},1}(\sigma_2 - \sigma_1) \otimes \mathcal{S}_{\mathrm{Q,reg},1}(\sigma_1)\Big) + \partial_s\mathcal{N}(\sigma_1)\Big(I_m \otimes \mathcal{S}_{\mathrm{Q,reg},1}(\sigma_1)\Big)\Big)
\end{aligned}$$

holds, by using the following identities

$$\begin{aligned}
\widehat{\mathcal{C}}(\sigma_2)\widehat{\mathcal{K}}(\sigma_2)^{-1}W^{\mathsf{H}} &= \mathcal{C}(\sigma_2)\mathcal{K}(\sigma_2)^{-1}, \\
V\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma_2 - \sigma_1) &= \mathcal{S}_{\mathrm{Q,reg},1}(\sigma_2 - \sigma_1), \\
V\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma_1) &= \mathcal{S}_{\mathrm{Q,reg},1}(\sigma_1),
\end{aligned}$$

which can be shown following the ideas from the previous proofs and the correct application of the projectors P_{V} and P_{W} from (3.24) and (3.25). Next, the term with the derivative at the first position in the Kronecker product is considered

$$\widehat{Z}_1 := -\widehat{\mathcal{C}}(\sigma_2)\widehat{\mathcal{K}}(\sigma_2)^{-1}\widehat{\mathcal{H}}(\sigma_2-\sigma_1,\sigma_1)\Big(\partial_{s_1}\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma_2-\sigma_1)\otimes\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma_1)\Big).$$

Associating the tensor $\widehat{\boldsymbol{Z}}_1$ with this term such that $\widehat{\boldsymbol{Z}}_1^{(1)} = \widehat{Z}_1$ holds, allows to use other matricizations equivalently. For the conjugate of the 3-mode matricization of $\widehat{\boldsymbol{Z}}_1$ together with (2.3) and the identities from above, it holds

$$\begin{aligned}
-\overline{\widehat{\boldsymbol{Z}}_1^{(3)}} &= \partial_{s_1}\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma_2-\sigma_1)^{\mathsf{H}}\overline{\widehat{\boldsymbol{\mathcal{H}}}^{(3)}(\sigma_2-\sigma_1,\sigma_1)}\Big(\overline{\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma_1)}\otimes\widehat{\mathcal{K}}(\sigma_2)^{-\mathsf{H}}\widehat{\mathcal{C}}(\sigma_2)^{\mathsf{H}}\Big)\\
&= -\partial_{s_1}\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma_2-\sigma_1)^{\mathsf{H}}V^{\mathsf{H}}\overline{\boldsymbol{\mathcal{H}}^{(3)}(\sigma_2-\sigma_1,\sigma_1)}\Big(\overline{V\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma_1)}\otimes W\widehat{\mathcal{K}}(\sigma_2)^{-\mathsf{H}}\widehat{\mathcal{C}}(\sigma_2)^{\mathsf{H}}\Big)\\
&= -\Big(\partial_s\widehat{\mathcal{B}}(\sigma_2-\sigma_1)^{\mathsf{H}}-\widehat{\mathcal{B}}(\sigma_2-\sigma_1)^{\mathsf{H}}\widehat{\mathcal{K}}(\sigma_2-\sigma_1)^{-\mathsf{H}}\partial_s\widehat{\mathcal{K}}(\sigma_2-\sigma_1)^{\mathsf{H}}\Big)\widehat{\mathcal{K}}(\sigma_2-\sigma_1)^{-\mathsf{H}}V^{\mathsf{H}}\\
&\quad\times\overline{\boldsymbol{\mathcal{H}}^{(3)}(\sigma_2-\sigma_1,\sigma_1)}\Big(\overline{\mathcal{S}_{\mathrm{Q,reg},1}(\sigma_1)}\otimes\mathcal{K}(\sigma_2)^{-\mathsf{H}}\mathcal{C}(\sigma_2)^{\mathsf{H}}\Big)\\
&= -\Big(\partial_s\mathcal{B}(\sigma_2-\sigma_1)^{\mathsf{H}}-\mathcal{S}_{\mathrm{Q,reg},1}(\sigma_2-\sigma_1)^{\mathsf{H}}\partial_s\mathcal{K}(\sigma_2-\sigma_1)^{\mathsf{H}}\Big)\\
&\quad\times\underbrace{W\widehat{\mathcal{K}}(\sigma_2-\sigma_1)^{-\mathsf{H}}V^{\mathsf{H}}\mathcal{K}(\sigma_2-\sigma_1)^{\mathsf{H}}}_{=P_{\mathrm{W}}(\sigma_2-\sigma_1)}\\
&\quad\times\underbrace{\mathcal{K}(\sigma_2-\sigma_1)^{-\mathsf{H}}\overline{\boldsymbol{\mathcal{H}}^{(3)}(\sigma_2-\sigma_1,\sigma_1)}\Big(\overline{\mathcal{S}_{\mathrm{Q,reg},1}(\sigma_1)}\otimes\mathcal{K}(\sigma_2)^{-\mathsf{H}}\mathcal{C}(\sigma_2)^{\mathsf{H}}\Big)}_{=W_3}\\
&= -\partial_{s_1}\mathcal{S}_{\mathrm{Q,reg},1}(\sigma_2-\sigma_1)^{\mathsf{H}}\overline{\boldsymbol{\mathcal{H}}^{(3)}(\sigma_2-\sigma_1,\sigma_1)}\Big(\overline{\mathcal{S}_{\mathrm{Q,reg},1}(\sigma_1)}\otimes\widehat{\mathcal{K}}(\sigma_2)^{-\mathsf{H}}\mathcal{C}(\sigma_2)^{\mathsf{H}}\Big),
\end{aligned}$$

and, therefore,

$$\widehat{Z}_1 := -\mathcal{C}(\sigma_2)\mathcal{K}(\sigma_2)^{-1}\mathcal{H}(\sigma_2-\sigma_1,\sigma_1)\Big(\partial_{s_1}\mathcal{S}_{\mathrm{Q,reg},1}(\sigma_2-\sigma_1)\otimes\mathcal{S}_{\mathrm{Q,reg},1}(\sigma_1)\Big).$$

Now, only the terms where the partial derivative enters in the second argument of the Kronecker products are left. Consider

$$\begin{aligned}
\widehat{Z}_2 := \widehat{\mathcal{C}}(\sigma_2)\widehat{\mathcal{K}}(\sigma_2)^{-1}\Big(&\widehat{\mathcal{H}}(\sigma_2-\sigma_1,\sigma_1)\Big(\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma_2-\sigma_1)\otimes\partial_{s_1}\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma_1)\Big)\\
&+\widehat{\mathcal{N}}(\sigma_1)\Big(I_m\otimes\partial_{s_1}\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma_1)\Big)\Big),
\end{aligned}$$

which is also re-interpreted as a tensor $\widehat{\boldsymbol{Z}}_2$ by $\widehat{\boldsymbol{Z}}_2^{(1)} = \widehat{Z}_2$. This time its 2-mode matricization is considered such that, with the identities from above and (2.2), it holds

$$\begin{aligned}
\overline{\widehat{\boldsymbol{Z}}_2^{(3)}} = \partial_{s_1}\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma_1)^{\mathsf{H}}\Big(&\overline{\widehat{\boldsymbol{\mathcal{H}}}^{(2)}(\sigma_2-\sigma_1,\sigma_1)}\Big(\overline{\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma_2-\sigma_1)}\otimes\widehat{\mathcal{K}}(\sigma_2)^{-\mathsf{H}}\widehat{\mathcal{C}}(\sigma_2)^{\mathsf{H}}\Big)\\
&+\overline{\widehat{\boldsymbol{\mathcal{N}}}^{(2)}(\sigma_1)}\Big(I_m\otimes\widehat{\mathcal{K}}(\sigma_2)^{-\mathsf{H}}\widehat{\mathcal{C}}(\sigma_2)^{\mathsf{H}}\Big)\Big)
\end{aligned}$$

$$
\begin{aligned}
&= \partial_{s_1}\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma_1)^{\mathsf{H}} V^{\mathsf{H}} \Big(\overline{\boldsymbol{\mathcal{H}}^{(2)}(\sigma_2-\sigma_1,\sigma_1)} \Big(\overline{V\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma_2-\sigma_1)} \otimes W\widehat{\mathcal{K}}(\sigma_2)^{-\mathsf{H}}\widehat{\mathcal{C}}(\sigma_2)^{\mathsf{H}} \Big) \\
&\quad + \overline{\boldsymbol{\mathcal{N}}^{(2)}(\sigma_1)} \Big(I_m \otimes W\widehat{\mathcal{K}}(\sigma_2)^{-\mathsf{H}}\widehat{\mathcal{C}}(\sigma_2)^{\mathsf{H}} \Big) \Big) \\
&= \Big(\partial_s \mathcal{B}(\sigma_1)^{\mathsf{H}} - \mathcal{S}_{\mathrm{Q,reg},1}(\sigma_1)^{\mathsf{H}} \partial_s \mathcal{K}(\sigma_1)^{\mathsf{H}} \Big) W\widehat{\mathcal{K}}(\sigma_1)^{-1} V^{\mathsf{H}} \\
&\quad \times \Big(\overline{\boldsymbol{\mathcal{H}}^{(2)}(\sigma_2-\sigma_1,\sigma_1)} \Big(\overline{V\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma_2-\sigma_1)} \otimes W\widehat{\mathcal{K}}(\sigma_2)^{-\mathsf{H}}\widehat{\mathcal{C}}(\sigma_2)^{\mathsf{H}} \Big) \\
&\quad + \overline{\boldsymbol{\mathcal{N}}^{(2)}(\sigma_1)} \Big(I_m \otimes W\widehat{\mathcal{K}}(\sigma_2)^{-\mathsf{H}}\widehat{\mathcal{C}}(\sigma_2)^{\mathsf{H}} \Big) \Big) \\
&= \Big(\partial_s \mathcal{B}(\sigma_1)^{\mathsf{H}} - \mathcal{S}_{\mathrm{Q,reg},1}(\sigma_1)^{\mathsf{H}} \partial_s \mathcal{K}(\sigma_1)^{\mathsf{H}} \Big) P_{\mathrm{W}}(\sigma_1) W_2 \\
&= \Big(\partial_s \mathcal{B}(\sigma_1)^{\mathsf{H}} - \mathcal{S}_{\mathrm{Q,reg},1}(\sigma_1)^{\mathsf{H}} \partial_s \mathcal{K}(\sigma_1)^{\mathsf{H}} \Big) W_2.
\end{aligned}
$$

Consequently, one obtains

$$
\begin{aligned}
\widehat{Z}_2 = \mathcal{C}(\sigma_2)\mathcal{K}(\sigma_2)^{-1} \Big(&\mathcal{H}(\sigma_2-\sigma_1,\sigma_1) \Big(\mathcal{S}_{\mathrm{Q,reg},1}(\sigma_2-\sigma_1) \otimes \partial_{s_1} \mathcal{S}_{\mathrm{Q,reg},1}(\sigma_1) \Big) \\
&+ \mathcal{N}(\sigma_1) \Big(I_m \otimes \partial_{s_1} \mathcal{S}_{\mathrm{Q,reg},1}(\sigma_1) \Big) \Big),
\end{aligned}
$$

which then yields the desired interpolation condition for the partial derivative with respect to s_1.

For Part (c), the partial derivative of the second reduced-order regular subsystem transfer function with respect to s_2 in the interpolation points is considered

$$
\begin{aligned}
\partial_{s_2} \widehat{\mathcal{G}}_{\mathrm{Q,reg},2}(\sigma_1,\sigma_2) = {} & \partial_s \widehat{\mathcal{C}}(\sigma_2) \widehat{\mathcal{S}}_{\mathrm{Q,reg},2}(\sigma_1,\sigma_2) - \widehat{\mathcal{C}}(\sigma_2)\widehat{\mathcal{K}}(\sigma_2)^{-1} \partial_s \widehat{\mathcal{K}}(\sigma_2) \widehat{\mathcal{S}}_{\mathrm{Q,reg},2}(\sigma_1,\sigma_2) \\
& + \widehat{\mathcal{C}}(\sigma_2)\widehat{\mathcal{K}}(\sigma_2)^{-1} \\
& \times \Big(\partial_{s_1} \widehat{\mathcal{H}}(\sigma_2-\sigma_1,\sigma_1) \Big(\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma_2-\sigma_1) \otimes \widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma_1) \Big) \\
& + \widehat{\mathcal{H}}(\sigma_2-\sigma_1,\sigma_1) \Big(\partial_{s_1} \widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma_2-\sigma_1) \otimes \widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma_1) \Big) \Big).
\end{aligned}
$$

The single terms in this derivative are then grouped into two with respect to occurrence of the partial derivatives in the Kronecker products. For the terms without the differentiation in the Kronecker products, one can quickly show that

$$
\begin{aligned}
&\partial_s \widehat{\mathcal{C}}(\sigma_2) \widehat{\mathcal{S}}_{\mathrm{Q,reg},2}(\sigma_1,\sigma_2) - \widehat{\mathcal{C}}(\sigma_2)\widehat{\mathcal{K}}(\sigma_2)^{-1} \partial_s \widehat{\mathcal{K}}(\sigma_2) \widehat{\mathcal{S}}_{\mathrm{Q,reg},2}(\sigma_1,\sigma_2) \\
&\quad + \widehat{\mathcal{C}}(\sigma_2)\widehat{\mathcal{K}}(\sigma_2)^{-1} \partial_{s_1} \widehat{\mathcal{H}}(\sigma_2-\sigma_1,\sigma_1) \Big(\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma_2-\sigma_1) \otimes \widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma_1) \Big) \\
&= \partial_s \mathcal{C}(\sigma_2) V \widehat{\mathcal{S}}_{\mathrm{Q,reg},2}(\sigma_1,\sigma_2) - \widehat{\mathcal{C}}(\sigma_2)\widehat{\mathcal{K}}(\sigma_2)^{-1} W^{\mathsf{H}} \partial_s \mathcal{K}(\sigma_2) V \widehat{\mathcal{S}}_{\mathrm{Q,reg},2}(\sigma_1,\sigma_2) \\
&\quad + \widehat{\mathcal{C}}(\sigma_2)\widehat{\mathcal{K}}(\sigma_2)^{-1} W^{\mathsf{H}} \partial_{s_1} \mathcal{H}(\sigma_2-\sigma_1,\sigma_1) \Big(V\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma_2-\sigma_1) \otimes V\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma_1) \Big) \\
&= \partial_s \mathcal{C}(\sigma_2) \mathcal{S}_{\mathrm{Q,reg},2}(\sigma_1,\sigma_2) - \mathcal{C}(\sigma_2)\mathcal{K}(\sigma_2)^{-1} \partial_s \mathcal{K}(\sigma_2) \mathcal{S}_{\mathrm{Q,reg},2}(\sigma_1,\sigma_2) \\
&\quad + \mathcal{C}(\sigma_2)\mathcal{K}(\sigma_2)^{-1} \partial_{s_1} \mathcal{H}(\sigma_2-\sigma_1,\sigma_1) \Big(\mathcal{S}_{\mathrm{Q,reg},1}(\sigma_2-\sigma_1) \otimes \mathcal{S}_{\mathrm{Q,reg},1}(\sigma_1) \Big)
\end{aligned}
$$

holds, where the identities from the proof of Part (b) were used as well as

$$V\widehat{\mathcal{S}}_{\mathrm{Q,reg},2}(\sigma_1,\sigma_2) = \mathcal{S}_{\mathrm{Q,reg},2}(\sigma_1,\sigma_2).$$

Only a single term with derivative in the Kronecker product is left. One can observe that this leftover term is, in fact, $\widehat{Z}_1$ without the minus sign in front, which was already proven to be interpolated by the use of W_3. Therefore, the interpolation of the partial derivative with respect to s_2 holds. □

Similar to Theorem 6.5, in case that Parts (b) and (c) of Theorem 6.11 are fulfilled, the complete Jacobian is interpolated

$$\nabla\mathcal{G}_{\mathrm{Q,reg},2}(\sigma_1,\sigma_2) = \nabla\widehat{\mathcal{G}}_{\mathrm{Q,reg},2}(\sigma_1,\sigma_2).$$

Further comparing the two theorems for implicit Hermite interpolation, one major difference stands out. In Theorem 6.11, less conditions are imposed on the left projection space $\operatorname{span}(W)$ than in Theorem 6.5 for the interpolation of the full Jacobi matrix, while for the simple interpolation in both transfer function concepts via two-sided projection the same number of conditions are needed. This roots in the larger number of frequency-dependent terms in symmetric transfer functions, which leads to more terms involved in the differentiation than in case of regular transfer functions.

In the literature, most results are available for the special case of SISO systems with symmetric quadratic tensors and the particular choice of interpolation points (6.29); see [4]. In a similar fashion, Theorem 6.11 simplifies significantly as it will be shown in the following theorem.

Theorem 6.12 (Implicit Hermite interpolation of reg. SISO TFs):
Let $\mathcal{G}_{\mathrm{Q}}$ be a quadratic-bilinear SISO system, described by its regular subsystem transfer functions $\mathcal{G}_{\mathrm{Q,reg},k}$ in (6.18) and (6.19), $\widehat{\mathcal{G}}_{\mathrm{Q}}$ the reduced-order quadratic-bilinear SISO system constructed by (6.22), with its reduced-order regular transfer functions $\widehat{\mathcal{G}}_{\mathrm{Q,reg},k}$, and let $\sigma \in \mathbb{C}$ be an interpolation point such that the matrix functions $\mathcal{C}$, $\mathcal{K}^{-1}$, $\mathcal{B}$, $\mathcal{N}$, $\mathcal{H}$ and $\widehat{\mathcal{K}}^{-1}$ are complex differentiable in σ and 2σ. Also, let the tensor $\boldsymbol{\mathcal{H}}$, given by its 1-mode matricization $\boldsymbol{\mathcal{H}}^{(1)} = \mathcal{H}$, be symmetric. Given the following vectors:

$$\begin{aligned} v_1 &= \mathcal{K}(\sigma)^{-1}\mathcal{B}(\sigma), \\ v_2 &= \mathcal{K}(2\sigma)^{-1}\Big(\mathcal{H}(\sigma,\sigma)(v_1\otimes v_1) + \mathcal{N}(\sigma)v_1\Big), \end{aligned}$$

and, also,

$$\begin{aligned} w_1 &= \mathcal{K}(2\sigma)^{-\mathsf{H}}\mathcal{C}(2\sigma)^{\mathsf{H}}, \\ w_2 &= \mathcal{K}(\sigma)^{-\mathsf{H}}\mathcal{N}(\sigma)^{\mathsf{H}}w_1, \\ w_3 &= \mathcal{K}(\sigma)^{-\mathsf{H}}\overline{\boldsymbol{\mathcal{H}}^{(2)}(\sigma,\sigma)}(\overline{v_1}\otimes w_1), \end{aligned}$$

then, the following statements hold:

(a) If V and W are constructed such that

$$\operatorname{span}(V) \supseteq \operatorname{span}(v_1) \quad \text{and} \quad \operatorname{span}(W) \supseteq \operatorname{span}(w_1),$$

then the following interpolation conditions hold:

$$\begin{aligned} \mathcal{G}_{\mathrm{Q,reg},1}(\sigma) &= \widehat{\mathcal{G}}_{\mathrm{Q,reg},1}(\sigma), \\ \mathcal{G}_{\mathrm{Q,reg},1}(2\sigma) &= \widehat{\mathcal{G}}_{\mathrm{Q,reg},1}(2\sigma), \\ \mathcal{G}_{\mathrm{Q,reg},2}(\sigma, 2\sigma) &= \widehat{\mathcal{G}}_{\mathrm{Q,reg},2}(\sigma, 2\sigma). \end{aligned} \tag{6.31}$$

(b) If V and W are constructed such that

$$\operatorname{span}(V) \supseteq \operatorname{span}(v_1) \quad \text{and} \quad \operatorname{span}(W) \supseteq \operatorname{span}\left(\begin{bmatrix} w_1 & w_2 \end{bmatrix}\right),$$

then, additionally to (6.31), the following Hermite interpolation condition holds:

$$\partial_{s_1} \mathcal{G}_{\mathrm{Q,reg},2}(\sigma, 2\sigma) = \partial_{s_1} \mathcal{G}_{\mathrm{Q,reg},2}(\sigma, 2\sigma).$$

(c) If V and W are constructed such that

$$\operatorname{span}(V) \supseteq \operatorname{span}\left(\begin{bmatrix} v_1 & v_2 \end{bmatrix}\right) \quad \text{and} \quad \operatorname{span}(W) \supseteq \operatorname{span}\left(\begin{bmatrix} w_1 & w_3 \end{bmatrix}\right),$$

then, additionally to (6.31), the following Hermite interpolation condition holds:

$$\partial_{s_2} \mathcal{G}_{\mathrm{Q,reg},2}(\sigma, 2\sigma) = \partial_{s_2} \mathcal{G}_{\mathrm{Q,reg},2}(\sigma, 2\sigma). \qquad \lozenge$$

Proof. Part (a) is a direct consequence of Theorem 6.11. For Part (b), the partial derivative of the second regular subsystem transfer function with respect to s_1 simplifies significantly. By making use of the symmetric tensor $\boldsymbol{\mathcal{H}}$, the SISO system and consequently (2.4), and the special choice of interpolation points, it holds

$$\begin{aligned} \partial_{s_1} \widehat{\mathcal{G}}_{\mathrm{Q,reg},2}(\sigma, 2\sigma) = \widehat{\mathcal{C}}(2\sigma)\widehat{\mathcal{K}}(2\sigma)^{-1} \Big(& \Big(\partial_{s_2} \widehat{\mathcal{H}}(\sigma,\sigma) - \partial_{s_1} \widehat{\mathcal{H}}(\sigma,\sigma)\Big)\Big(\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma) \otimes \widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma)\Big) \\ & - \widehat{\mathcal{H}}(\sigma,\sigma)\Big(\partial_{s_1} \widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma) \otimes \widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma)\Big) \\ & + \widehat{\mathcal{H}}(\sigma,\sigma)\Big(\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma) \otimes \partial_{s_1} \widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma)\Big) \\ & + \partial_s \widehat{\mathcal{N}}(\sigma) \widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma) + \widehat{\mathcal{N}}(\sigma) \partial_{s_1} \widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma) \Big) \\ = \widehat{\mathcal{C}}(2\sigma)\widehat{\mathcal{K}}(2\sigma)^{-1} \Big(& \Big(\partial_{s_2} \widehat{\mathcal{H}}(\sigma,\sigma) - \partial_{s_1} \widehat{\mathcal{H}}(\sigma,\sigma)\Big)\Big(\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma) \otimes \widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma)\Big) \\ & + \partial_s \widehat{\mathcal{N}}(\sigma) \widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma) + \widehat{\mathcal{N}}(\sigma) \partial_{s_1} \widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma) \Big). \end{aligned}$$

Note that due to the symmetric tensor, the terms with derivatives in the Kronecker products, which were multiplied with the quadratic term, vanished. Grouping the remaining terms into two yields

$$\begin{aligned}
&\widehat{\mathcal{C}}(2\sigma)\widehat{\mathcal{K}}(2\sigma)^{-1}\Big(\big(\partial_{s_2}\widehat{\mathcal{H}}(\sigma,\sigma)-\partial_{s_1}\widehat{\mathcal{H}}(\sigma,\sigma)\big)\big(\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma)\otimes\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma)\big)\\
&\quad+\partial_s\widehat{\mathcal{N}}(\sigma)\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma)\Big)\\
&=\widehat{\mathcal{C}}(2\sigma)\widehat{\mathcal{K}}(2\sigma)^{-1}W^{\mathsf{H}}\Big(\big(\partial_{s_2}\mathcal{H}(\sigma,\sigma)-\partial_{s_1}\mathcal{H}(\sigma,\sigma)\big)\big(V\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma)\otimes V\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma)\big)\\
&\quad+\partial_s\mathcal{N}(\sigma)V\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma)\Big)\\
&=\mathcal{C}(2\sigma)\mathcal{K}(2\sigma)^{-1}\Big(\big(\partial_{s_2}\mathcal{H}(\sigma,\sigma)-\partial_{s_1}\mathcal{H}(\sigma,\sigma)\big)\big(\mathcal{S}_{\mathrm{Q,reg},1}(\sigma)\otimes\mathcal{S}_{\mathrm{Q,reg},1}(\sigma)\big)\\
&\quad+\partial_s\mathcal{N}(\sigma)\mathcal{S}_{\mathrm{Q,reg},1}(\sigma)\Big),
\end{aligned}$$

by using similar identities to those in the proof of Theorem 6.11. On the other hand, one can show that

$$\begin{aligned}
&\widehat{\mathcal{C}}(2\sigma)\widehat{\mathcal{K}}(2\sigma)^{-1}\widehat{\mathcal{N}}(\sigma)\partial_{s_1}\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma)\\
&=\widehat{\mathcal{C}}(2\sigma)\widehat{\mathcal{K}}(2\sigma)^{-1}\widehat{\mathcal{N}}(\sigma)\widehat{\mathcal{K}}(\sigma)^{-1}\big(\partial_s\widehat{\mathcal{B}}(\sigma)-\partial_s\widehat{\mathcal{K}}(\sigma)\widehat{\mathcal{K}}(\sigma)^{-1}\widehat{\mathcal{B}}(\sigma)\big)\\
&=\mathcal{C}(2\sigma)V\widehat{\mathcal{K}}(2\sigma)^{-1}W^{\mathsf{H}}\mathcal{N}(\sigma)V\widehat{\mathcal{K}}(\sigma)^{-1}W^{\mathsf{H}}\big(\partial_s\mathcal{B}(\sigma)-\partial_s\mathcal{K}(\sigma)V\widehat{\mathcal{K}}(\sigma)^{-1}\widehat{\mathcal{B}}(\sigma)\big)\\
&=w_1^{\mathsf{H}}P_{\mathrm{W}}(2\sigma)^{\mathsf{H}}\mathcal{N}(\sigma)\mathcal{K}(\sigma)^{-1}P_{\mathrm{W}}(\sigma)^{\mathsf{H}}\big(\partial_s\mathcal{B}(\sigma)-\partial_s\mathcal{K}(\sigma)P_{\mathrm{V}}(\sigma)v_1\big)\\
&=w_2^{\mathsf{H}}P_{\mathrm{W}}(\sigma)^{\mathsf{H}}\partial_{s_1}\mathcal{S}_{\mathrm{Q,reg},1}(\sigma)\\
&=\mathcal{C}(2\sigma)\mathcal{K}(2\sigma)^{-1}\mathcal{N}(\sigma)\partial_{s_1}\mathcal{S}_{\mathrm{Q,reg},1}(\sigma)
\end{aligned}$$

holds, with P_{V} and P_{W} the projectors from (3.24) and (3.25), respectively, which gives the desired interpolation result.

Part (c) can be shown in an analogous fashion. The partial derivative of the second reduced-order regular subsystem transfer function with respect to s_2 in the interpolation points is given by

$$\begin{aligned}
\partial_{s_2}\widehat{\mathcal{G}}_{\mathrm{Q,reg},2}(\sigma,2\sigma)=\partial_s\widehat{\mathcal{C}}(2\sigma)\widehat{\mathcal{S}}_{\mathrm{Q,reg},2}(\sigma,2\sigma)-\widehat{\mathcal{C}}(2\sigma)\widehat{\mathcal{K}}(2\sigma)^{-1}\partial_s\widehat{\mathcal{K}}(2\sigma)\widehat{\mathcal{S}}_{\mathrm{Q,reg},2}(\sigma,2\sigma)\\
+\widehat{\mathcal{C}}(2\sigma)\widehat{\mathcal{K}}(2\sigma)^{-1}\Big(\partial_{s_1}\widehat{\mathcal{H}}(\sigma,\sigma)\big(\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma)\otimes\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma)\big)\\
+\widehat{\mathcal{H}}(\sigma,\sigma)\big(\partial_{s_1}\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma)\otimes\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma)\big)\Big).
\end{aligned}$$

For the first two terms, it holds

$$\begin{aligned}
&\partial_s \widehat{\mathcal{C}}(2\sigma)\widehat{\mathcal{S}}_{\mathrm{Q,reg},2}(\sigma,2\sigma) - \widehat{\mathcal{C}}(2\sigma)\widehat{\mathcal{K}}(2\sigma)^{-1}\partial_s\widehat{\mathcal{K}}(2\sigma)\widehat{\mathcal{S}}_{\mathrm{Q,reg},2}(\sigma,2\sigma)\\
&= \partial_s\mathcal{C}(2\sigma)V\widehat{\mathcal{S}}_{\mathrm{Q,reg},2}(\sigma,2\sigma) - \mathcal{C}(2\sigma)V\widehat{\mathcal{K}}(2\sigma)^{-1}W^{\mathsf{H}}\partial_s\mathcal{K}(2\sigma)V\widehat{\mathcal{S}}_{\mathrm{Q,reg},2}(\sigma,2\sigma)\\
&= \partial_s\mathcal{C}(2\sigma)P_{\mathrm{V}}(2\sigma)v_2 - w_1^{\mathsf{H}}P_{\mathrm{W}}(2\sigma)^{\mathsf{H}}\partial_s\mathcal{K}(2\sigma)P_{\mathrm{V}}(\sigma)v_2\\
&= \partial_s\mathcal{C}(2\sigma)\mathcal{S}_{\mathrm{Q,reg},2}(\sigma,2\sigma) - \mathcal{C}(2\sigma)\mathcal{K}(2\sigma)^{-1}\partial_s\mathcal{K}(2\sigma)\mathcal{S}_{\mathrm{Q,reg},2}(\sigma,2\sigma).
\end{aligned}$$

Similarly, one can show that

$$\begin{aligned}
&\widehat{\mathcal{C}}(2\sigma)\widehat{\mathcal{K}}(2\sigma)^{-1}\partial_{s_1}\widehat{\mathcal{H}}(\sigma,\sigma)\Big(\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma)\otimes\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma)\Big)\\
&= \mathcal{C}(2\sigma)V\widehat{\mathcal{K}}(2\sigma)^{-1}W\partial_{s_1}\mathcal{H}(\sigma,\sigma)\Big(V\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma)\otimes V\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma)\Big)\\
&= w_1P_{\mathrm{W}}(2\sigma)\partial_{s_1}\mathcal{H}(\sigma,\sigma)\Big(P_{\mathrm{V}}(\sigma)v_1\otimes P_{\mathrm{V}}(\sigma)v_1\Big)\\
&= \mathcal{C}(2\sigma)\mathcal{K}(2\sigma)^{-1}\partial_{s_1}\mathcal{H}(\sigma,\sigma)\Big(\mathcal{S}_{\mathrm{Q,reg},1}(\sigma)\otimes\mathcal{S}_{\mathrm{Q,reg},1}(\sigma)\Big)
\end{aligned}$$

holds. The term left contains the matrix function for the quadratic components and a derivative in the associated Kronecker product

$$\widehat{Z} := \widehat{\mathcal{C}}(2\sigma)\widehat{\mathcal{K}}(2\sigma)^{-1}\widehat{\mathcal{H}}(\sigma,\sigma)\Big(\partial_{s_1}\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma)\otimes\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma)\Big).$$

Therefore, it is considered again with the underlying tensor $\widehat{\boldsymbol{Z}}^{(1)} = \widehat{Z}$ such that

$$\begin{aligned}
\overline{\widehat{\boldsymbol{Z}}^{(2)}} &= \partial_{s_1}\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma)^{\mathsf{H}}\overline{\widehat{\boldsymbol{\mathcal{H}}}^{(2)}(\sigma,\sigma)}\Big(\overline{\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma)}\otimes\widehat{\mathcal{K}}(2\sigma)^{-\mathsf{H}}\widehat{\mathcal{C}}(2\sigma)^{\mathsf{H}}\Big)\\
&= \partial_{s_1}\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma)^{\mathsf{H}}V^{\mathsf{H}}\overline{\boldsymbol{\mathcal{H}}^{(2)}(\sigma,\sigma)}\Big(\overline{V\widehat{\mathcal{S}}_{\mathrm{Q,reg},1}(\sigma)}\otimes W\widehat{\mathcal{K}}(2\sigma)^{-\mathsf{H}}\widehat{\mathcal{C}}(2\sigma)^{\mathsf{H}}\Big)\\
&= \Big(\partial_s\mathcal{B}(\sigma) - \partial\mathcal{K}(\sigma)\mathcal{K}(\sigma)^{-1}\mathcal{B}(\sigma)\Big)^{\mathsf{H}}P_{\mathrm{W}}(\sigma)\mathcal{K}(2\sigma)^{-\mathsf{H}}\overline{\boldsymbol{\mathcal{H}}^{(2)}(\sigma,\sigma)}\\
&\quad\times\Big(\overline{P_{\mathrm{V}}(\sigma)v_1}\otimes P_{\mathrm{W}}(2\sigma)w_1\Big)\\
&= \Big(\partial_s\mathcal{B}(\sigma) - \partial\mathcal{K}(\sigma)\mathcal{K}(\sigma)^{-1}\mathcal{B}(\sigma)\Big)^{\mathsf{H}}P_{\mathrm{W}}(\sigma)w_3\\
&= \partial_{s_1}\mathcal{S}_{\mathrm{Q,reg},1}(\sigma)^{\mathsf{H}}\overline{\boldsymbol{\mathcal{H}}^{(2)}(\sigma,\sigma)}\Big(\overline{\mathcal{S}_{\mathrm{Q,reg},1}(\sigma)}\otimes\mathcal{K}(2\sigma)^{-\mathsf{H}}\mathcal{C}(2\sigma)^{\mathsf{H}}\Big)
\end{aligned}$$

holds, where (2.2) and (2.4) and the projectors (3.24) and (3.25) were used. This gives finally the desired interpolation of the partial derivative with respect to s_2. □

Comparing Theorem 6.12 with its equivalent version in case of symmetric transfer functions Theorem 6.6 shows some interesting differences. While in both theorems for the right projection space one vector less is needed than before, the change of the left projection space differs a lot. In case of symmetric transfer functions it was possible to gather four of the previous conditions on the left projection space into a single one thanks to the structural symmetry of the partial derivatives, which became even identical

by the specific choice of interpolation points $\sigma_1 = \sigma_2 = \sigma$. This was not possible in case of regular transfer functions. For these, the derivatives simplify in the sense that in one case, there are no derivatives in the Kronecker products left, and, in the other case, no bilinear term is involved anymore. This big difference in the partial derivatives makes it impossible to combine the two associated conditions on the left projection space.

6.4.4 Interpolating structured generalized transfer functions

At last, the structured generalized transfer functions are considered. Therefore, a similar interpolation theory as in the previous two sections for symmetric and regular transfer functions is developed, starting with the following theorem of interpolation by right projection.

Theorem 6.13 (Generalized transfer function interpolation via V):
Let $\mathcal{G}_{\mathrm{Q}}$ be a quadratic-bilinear system, described by its generalized transfer functions $\mathcal{G}^{\gamma}_{\mathrm{Q,gen},k}$ in (6.20) and (6.21), and $\widehat{\mathcal{G}}_{\mathrm{Q}}$ the reduced-order quadratic-bilinear system constructed by (6.22), with its reduced-order generalized transfer functions $\widehat{\mathcal{G}}^{\gamma}_{\mathrm{Q,gen},k}$. Also, let $\sigma_1, \ldots, \sigma_k \in \mathbb{C}$ be a set of interpolation points such that the matrix functions $\mathcal{C}$, $\mathcal{K}^{-1}$, $\mathcal{B}$, $\mathcal{N}$, $\mathcal{H}$ and $\widehat{\mathcal{K}}^{-1}$ are defined in these points, and let γ be an appropriate nested tuple. Construct V using the following recursive concatenation of matrices:

$$V(\gamma, \sigma_1, \ldots, \sigma_j) = \begin{cases} \Big[\Gamma(\gamma, \sigma_j)\Big], & \text{if } \gamma = (\mathrm{B}), \\ \Big[\Gamma(\gamma, \sigma_1, \ldots, \sigma_j),\ V(\gamma_{j-1}, \sigma_1, \ldots, \sigma_{j-1})\Big], & \text{if } \gamma = (\mathrm{N}, \gamma_{j-1}), \\ \Big[\Gamma(\gamma, \sigma_1, \ldots, \sigma_j),\ V(\gamma_{j-1}, \sigma_\ell, \ldots, \sigma_{j-1}), & \\ \quad V(\gamma_{j-2}, \sigma_1, \ldots, \sigma_{\ell-1})\Big], & \text{if } \gamma = (\mathrm{H}, \gamma_{j-1}, \gamma_{j-2}), \end{cases}$$

$$\operatorname{span}(V) \supseteq \operatorname{span}\Big(V(\gamma, \sigma_1, \ldots, \sigma_k)\Big),$$

and let W be an arbitrary full-rank truncation matrix of appropriate dimension. Then the generalized transfer functions of $\widehat{\mathcal{G}}_{\mathrm{Q}}$ interpolate those of $\mathcal{G}_{\mathrm{Q}}$ in the following way:

$$\mathcal{G}^{\gamma}_{\mathrm{Q,gen},j}(\sigma_1, \ldots, \sigma_j) = \widehat{\mathcal{G}}^{\gamma}_{\mathrm{Q,gen},j}(\sigma_1, \ldots, \sigma_j),$$

and, additionally, if $\gamma = (\mathrm{N}, \gamma_{j-1})$:

$$\mathcal{G}^{\gamma_{j-1}}_{\mathrm{Q,gen},j-1}(\sigma_1, \ldots, \sigma_{j-1}) = \widehat{\mathcal{G}}^{\gamma_{j-1}}_{\mathrm{Q,gen},j-1}(\sigma_1, \ldots, \sigma_{j-1}),$$

or, additionally, if $\gamma = (\mathrm{H}, \gamma_{j-1}, \gamma_{j-2})$:

$$\mathcal{G}^{\gamma_{j-1}}_{\mathrm{Q,gen},j-\ell+1}(\sigma_\ell, \ldots, \sigma_{j-1}) = \widehat{\mathcal{G}}^{\gamma_{j-1}}_{\mathrm{Q,gen},j-\ell+1}(\sigma_\ell, \ldots, \sigma_{j-1}),$$
$$\mathcal{G}^{\gamma_{j-2}}_{\mathrm{Q,gen},\ell-1}(\sigma_1, \ldots, \sigma_{\ell-1}) = \widehat{\mathcal{G}}^{\gamma_{j-2}}_{\mathrm{Q,gen},\ell-1}(\sigma_1, \ldots, \sigma_{\ell-1}),$$

for $j = k, k-1, \ldots, 1$. ◊

Proof. The proof follows exactly the same projection arguments as used in the proofs of Theorems 6.2 and 6.7, and the recursive construction of $\operatorname{span}(V)$. □

As for the other two transfer function concepts, for illustration of Theorem 6.13, a higher-level example is considered. Here, the aim will be the interpolation of $\mathcal{G}^{\gamma}_{\mathrm{Q,gen},4}$, with $\gamma = (\mathrm{H},(\mathrm{N},(\mathrm{B})),(\mathrm{B}))$, in the interpolation points $\sigma_1, \sigma_2, \sigma_3, \sigma_4$. By the construction formula of the projection space in Theorem 6.13, the matrix function $V(\gamma, \sigma_1, \sigma_2, \sigma_3, \sigma_4)$ will construct the following sub-matrices in concatenation (ordered by their occurrence in the construction formula):

$$\begin{aligned} &\Gamma((\mathrm{H},(\mathrm{N},(\mathrm{B})),(\mathrm{B})), \sigma_1, \sigma_2, \sigma_3, \sigma_4), \\ &\Gamma((\mathrm{N},(\mathrm{B})), \sigma_2, \sigma_3), \\ &\Gamma((\mathrm{B}), \sigma_1), \\ &\Gamma((\mathrm{B}), \sigma_2). \end{aligned} \tag{6.32}$$

Reversing the order yields the following practical construction of the projection space

$$\begin{aligned} V_1 &= \mathcal{K}(\sigma_2)^{-1}\mathcal{B}(\sigma_2), \\ V_2 &= \mathcal{K}(\sigma_1)^{-1}\mathcal{B}(\sigma_1), \\ V_3 &= \mathcal{K}(\sigma_3)^{-1}\mathcal{N}(\sigma_2)(I_m \otimes V_1), \\ V_4 &= \mathcal{K}(\sigma_4)^{-1}\mathcal{H}(\sigma_3, \sigma_1)(V_3 \otimes V_2), \end{aligned}$$

such that

$$\operatorname{span}(V) \supseteq \operatorname{span}\left(\begin{bmatrix} V_1 & V_2 & V_3 & V_4 \end{bmatrix}\right).$$

Using now V for model reduction by projection yields the interpolation of the following function values ordered according to (6.32):

$$\mathcal{G}^{(\mathrm{H},(\mathrm{N},(\mathrm{B})),(\mathrm{B}))}_{\mathrm{Q,gen},4}(\sigma_1, \sigma_2, \sigma_3, \sigma_4), \quad \mathcal{G}^{(\mathrm{N},(\mathrm{B}))}_{\mathrm{Q,gen},2}(\sigma_2, \sigma_3), \quad \mathcal{G}^{(\mathrm{B})}_{\mathrm{Q,gen},1}(\sigma_1), \quad \mathcal{G}^{(\mathrm{B})}_{\mathrm{Q,gen},1}(\sigma_2).$$

In contrast to the previous transfer functions (6.14) and (6.18), the number of frequency-dependent terms is only growing linearly with the transfer function level since no additional linear combinations are involved. This allows to match k interpolation conditions, when constructing the projection space for a k-th level generalized transfer function. In the SISO system case, this only needs the computation of k vectors compared to (6.25) and (6.28). This gives a lot of additional freedom for choosing interpolation points and transfer function levels in model order reduction. There is unfortunately no direct way of computing the potential dimension of the projection space as for the other transfer function concepts since here, the dimension strongly depends on the nested tuple

γ. In principle, the number of columns to be computed can be recursively determined by

$$f(\gamma) = \begin{cases} m, & \text{if } \gamma = (\mathrm{B}), \\ \dim_2\Big(\Gamma(\gamma,.)\Big) + f(\gamma_{j-1}), & \text{if } \gamma = (\mathrm{N},\gamma_{j-1}), \\ \dim_2\Big(\Gamma(\gamma,.)\Big) + f(\gamma_{j-1}) + f(\gamma_{j-2}), & \text{if } \gamma = (\mathrm{H},\gamma_{j-1},\gamma_{j-2}), \end{cases}$$

where $\dim_2(X)$ gives the column dimension of a matrix X. For the example above, this function resolves as follows

$$\begin{aligned} f((\mathrm{H},(\mathrm{N},(\mathrm{B})),(\mathrm{B}))) &= m^3 + f((\mathrm{N},(\mathrm{B}))) + f((\mathrm{B}))) \\ &= m^3 + m^2 + f((\mathrm{B})) + m \\ &= m^3 + m^2 + 2m. \end{aligned}$$

Due to the complexity of the recursion formulae in Theorem 6.13, a more practical version of the theorem that restricts to the interpolation of the generalized transfer functions of second level for the bilinear term and third level for the quadratic term is outlined below. Note the difference to Corollaries 6.3 and 6.8, where already the second transfer function level contains the quadratic term.

Corollary 6.14 (Simplified generalized transfer function interpolation):
Let $\mathcal{G}_{\mathrm{Q}}$ be a quadratic-bilinear system, described by its generalized transfer functions $\mathcal{G}^{\gamma}_{\mathrm{Q},\mathrm{gen},k}$ in (6.20) and (6.21), and $\widehat{\mathcal{G}}_{\mathrm{Q}}$ the reduced-order quadratic-bilinear system constructed by (6.22), with its reduced-order generalized transfer functions $\widehat{\mathcal{G}}^{\gamma}_{\mathrm{Q},\mathrm{gen},k}$. Also, let $\sigma_1, \sigma_2, \sigma_3 \in \mathbb{C}$ be three interpolation points such that the matrix functions $\mathcal{C}$, $\mathcal{K}^{-1}$, $\mathcal{B}$, $\mathcal{N}$, $\mathcal{H}$ and $\widehat{\mathcal{K}}^{-1}$ are defined in these points. Construct V using

$$\begin{aligned} V_1 &= \mathcal{K}(\sigma_1)^{-1}\mathcal{B}(\sigma_1), \\ V_2 &= \mathcal{K}(\sigma_2)^{-1}\mathcal{B}(\sigma_2), \\ V_3 &= \mathcal{K}(\sigma_2)^{-1}\mathcal{N}(\sigma_1)(I_m \otimes V_1), \\ V_4 &= \mathcal{K}(\sigma_3)^{-1}\mathcal{H}(\sigma_2,\sigma_1)\Big(V_2 \otimes V_1\Big), \\ \operatorname{span}(V) &\supseteq \operatorname{span}\left(\begin{bmatrix} V_1 & V_2 & V_3 & V_4 \end{bmatrix}\right), \end{aligned}$$

and let W be an arbitrary full-rank truncation matrix of appropriate dimension. Then the generalized transfer functions of $\widehat{\mathcal{G}}_{\mathrm{Q}}$ interpolate those of $\mathcal{G}_{\mathrm{Q}}$ in the following way:

$$\begin{aligned} \mathcal{G}^{(\mathrm{B})}_{\mathrm{Q},\mathrm{gen},1}(\sigma_1) &= \widehat{\mathcal{G}}^{(\mathrm{B})}_{\mathrm{Q},\mathrm{gen},1}(\sigma_1), \\ \mathcal{G}^{(\mathrm{B})}_{\mathrm{Q},\mathrm{gen},1}(\sigma_2) &= \widehat{\mathcal{G}}^{(\mathrm{B})}_{\mathrm{Q},\mathrm{gen},1}(\sigma_2), \\ \mathcal{G}^{(\mathrm{N},(\mathrm{B}))}_{\mathrm{Q},\mathrm{gen},2}(\sigma_1,\sigma_2) &= \widehat{\mathcal{G}}^{(\mathrm{N},(\mathrm{B}))}_{\mathrm{Q},\mathrm{gen},2}(\sigma_1,\sigma_2), \\ \mathcal{G}^{(\mathrm{H},(\mathrm{B}),(\mathrm{B}))}_{\mathrm{Q},\mathrm{gen},3}(\sigma_1,\sigma_2,\sigma_3) &= \widehat{\mathcal{G}}^{(\mathrm{H},(\mathrm{B}),(\mathrm{B}))}_{\mathrm{Q},\mathrm{gen},3}(\sigma_1,\sigma_2,\sigma_3). \end{aligned}$$ ◊

Like for the previous two transfer function concepts, it is possible to produce the interpolation results of Corollary 6.14 and even more in an implicit way, without evaluating the nonlinear terms at all. The idea is to make use of the two-sided projection and the second basis matrix W.

Lemma 6.15 (Implicit generalized transfer function interpolation):
Let $\mathcal{G}_{\mathrm{Q}}$ be a quadratic-bilinear system, described by its generalized transfer functions $\mathcal{G}^{\gamma}_{\mathrm{Q},\mathrm{gen},k}$ in (6.20) and (6.21), $\widehat{\mathcal{G}}_{\mathrm{Q}}$ the reduced-order quadratic-bilinear system constructed by (6.22), with its reduced-order generalized transfer functions $\widehat{\mathcal{G}}^{\gamma}_{\mathrm{Q},\mathrm{gen},k}$, and $\sigma_1, \sigma_2, \sigma_3 \in \mathbb{C}$ three interpolation points such that the matrix functions $\mathcal{C}$, $\mathcal{K}^{-1}$, $\mathcal{B}$, $\mathcal{N}$, $\mathcal{H}$ and $\widehat{\mathcal{K}}^{-1}$ are defined in all these points and complex differentiable in σ_2. Construct V using

$$\operatorname{span}(V) \supseteq \operatorname{span}\left(\begin{bmatrix}\mathcal{K}(\sigma_1)^{-1}\mathcal{B}(\sigma_1) & \mathcal{K}(\sigma_2)^{-1}\mathcal{B}(\sigma_2)\end{bmatrix}\right),$$

and W using

$$\operatorname{span}(W) \supseteq \operatorname{span}\left(\begin{bmatrix}\mathcal{K}(\sigma_2)^{-\mathsf{H}}\mathcal{C}(\sigma_2)^{\mathsf{H}} & \mathcal{K}(\sigma_3)^{-\mathsf{H}}\mathcal{C}(\sigma_3)^{\mathsf{H}}\end{bmatrix}\right),$$

and let the two matrices V and W be of appropriate dimensions. Then the generalized transfer functions of $\widehat{\mathcal{G}}_{\mathrm{Q}}$ interpolate those of $\mathcal{G}_{\mathrm{Q}}$ in the following way:

$$\begin{aligned}
\mathcal{G}^{(\mathrm{B})}_{\mathrm{Q},\mathrm{gen},1}(\sigma_1) &= \widehat{\mathcal{G}}^{(\mathrm{B})}_{\mathrm{Q},\mathrm{gen},1}(\sigma_1),\\
\mathcal{G}^{(\mathrm{N},(\mathrm{B}))}_{\mathrm{Q},\mathrm{gen},2}(\sigma_1,\sigma_2) &= \widehat{\mathcal{G}}^{(\mathrm{N},(\mathrm{B}))}_{\mathrm{Q},\mathrm{gen},2}(\sigma_1,\sigma_2),\\
\mathcal{G}^{(\mathrm{H},(\mathrm{B}),(\mathrm{B}))}_{\mathrm{Q},\mathrm{gen},3}(\sigma_1,\sigma_2,\sigma_3) &= \widehat{\mathcal{G}}^{(\mathrm{H},(\mathrm{B}),(\mathrm{B}))}_{\mathrm{Q},\mathrm{gen},3}(\sigma_1,\sigma_2,\sigma_3),
\end{aligned}$$

and, additionally,

$$\begin{aligned}
\mathcal{G}^{(\mathrm{B})}_{\mathrm{Q},\mathrm{gen},1}(\sigma_2) &= \widehat{\mathcal{G}}^{(\mathrm{B})}_{\mathrm{Q},\mathrm{gen},1}(\sigma_2), & \mathcal{G}^{(\mathrm{B})}_{\mathrm{Q},\mathrm{gen},1}(\sigma_3) &= \widehat{\mathcal{G}}^{(\mathrm{B})}_{\mathrm{Q},\mathrm{gen},1}(\sigma_3),\\
\nabla\mathcal{G}^{(\mathrm{B})}_{\mathrm{Q},\mathrm{gen},1}(\sigma_2) &= \nabla\widehat{\mathcal{G}}^{(\mathrm{B})}_{\mathrm{Q},\mathrm{gen},1}(\sigma_2), & \mathcal{G}^{(\mathrm{N},(\mathrm{B}))}_{\mathrm{Q},\mathrm{gen},2}(\sigma_1,\sigma_3) &= \widehat{\mathcal{G}}^{(\mathrm{N},(\mathrm{B}))}_{\mathrm{Q},\mathrm{gen},2}(\sigma_1,\sigma_3),\\
\mathcal{G}^{(\mathrm{N},(\mathrm{B}))}_{\mathrm{Q},\mathrm{gen},2}(\sigma_2,\sigma_2) &= \widehat{\mathcal{G}}^{(\mathrm{N},(\mathrm{B}))}_{\mathrm{Q},\mathrm{gen},2}(\sigma_2,\sigma_2), & \mathcal{G}^{(\mathrm{N},(\mathrm{B}))}_{\mathrm{Q},\mathrm{gen},2}(\sigma_2,\sigma_3) &= \widehat{\mathcal{G}}^{(\mathrm{N},(\mathrm{B}))}_{\mathrm{Q},\mathrm{gen},2}(\sigma_2,\sigma_3),
\end{aligned}$$

and

$$\mathcal{G}^{(\mathrm{H},(\mathrm{B}),(\mathrm{B}))}_{\mathrm{Q},\mathrm{gen},3}(\omega_{3,j_3}) = \widehat{\mathcal{G}}^{(\mathrm{H},(\mathrm{B}),(\mathrm{B}))}_{\mathrm{Q},\mathrm{gen},3}(\omega_{3,j_3}), \qquad j_3 = 1,\ldots,7,$$

in the interpolation points

$$\begin{aligned}
&\omega_{3,1} = (\sigma_1,\sigma_1,\sigma_2), \quad \omega_{3,2} = (\sigma_2,\sigma_2,\sigma_2), \quad \omega_{3,3} = (\sigma_1,\sigma_1,\sigma_3), \quad \omega_{3,4} = (\sigma_2,\sigma_2,\sigma_3),\\
&\omega_{3,5} = (\sigma_1,\sigma_2,\sigma_2), \quad \omega_{3,6} = (\sigma_2,\sigma_1,\sigma_2), \quad \omega_{3,7} = (\sigma_2,\sigma_1,\sigma_3).
\end{aligned}$$

◊

Proof. The interpolation conditions for the first-level transfer functions follow from Proposition 3.2 and for the second-level generalized transfer functions from Section 5.4. The interpolation of the third-level generalized transfer function is proven exactly the same way with the construction of appropriate projectors as done in the proofs of Lemmas 6.4 and 6.10. □

It was shown in [92] that for first-order quadratic-bilinear SISO systems (2.35) with the additional assumption of an underlying symmetric tensor for the quadratic term, the left truncation matrix W could be used to implicitly interpolate different higher-level generalized transfer functions in specific interpolation points. Also, in case of purely bilinear systems without the quadratic nonlinearity, the generalized and regular transfer functions are the same and yield a lot richer interpolation theory in the structure-preserving setting (Chapter 5), e.g., allowing for high-order Hermite interpolation or equivalent results for the left truncation matrix W as for V in terms of construction and interpolation conditions. The following theorem generalizes the interpolation of generalized transfer functions to the implicit matching of Hermite interpolation conditions.

Theorem 6.16 (Implicit Hermite interpolation of gen. transfer functions): Let $\mathcal{G}_{\mathrm{Q}}$ be a quadratic-bilinear system, described by its generalized transfer functions $\mathcal{G}^{\gamma}_{\mathrm{Q,gen},k}$ in (6.20) and (6.21), $\widehat{\mathcal{G}}_{\mathrm{Q}}$ the reduced-order quadratic-bilinear system constructed by (6.22), with its reduced-order generalized transfer functions $\widehat{\mathcal{G}}^{\gamma}_{\mathrm{Q,gen},k}$, and $\sigma_1, \sigma_2, \sigma_3 \in \mathbb{C}$ three interpolation points such that the matrix functions $\mathcal{C}$, $\mathcal{K}^{-1}$, $\mathcal{B}$, $\mathcal{N}$, $\mathcal{H}$ and $\widehat{\mathcal{K}}^{-1}$ are complex differentiable in these points. Let the following matrices be given

$$\begin{aligned}
V_1 &= \mathcal{K}(\sigma_1)^{-1}\mathcal{B}(\sigma_1),\\
V_2 &= \mathcal{K}(\sigma_2)^{-1}\mathcal{N}(\sigma_1)(I_m \otimes V_1),\\
V_3 &= \mathcal{K}(\sigma_2)^{-1}\mathcal{B}(\sigma_2),\\
V_4 &= \mathcal{K}(\sigma_3)^{-1}\mathcal{H}(\sigma_2,\sigma_1)(V_3 \otimes V_1),
\end{aligned}$$

and, also,

$$\begin{aligned}
W_1 &= \mathcal{K}(\sigma_1)^{-\mathsf{H}}\mathcal{C}(\sigma_1)^{\mathsf{H}},\\
W_2 &= \mathcal{K}(\sigma_2)^{-\mathsf{H}}\mathcal{C}(\sigma_2)^{\mathsf{H}},\\
W_3 &= \mathcal{K}(\sigma_1)^{-\mathsf{H}}\overline{\boldsymbol{\mathcal{N}}^{(2)}(\sigma_1)}(I_m \otimes W_2),\\
W_4 &= \mathcal{K}(\sigma_3)^{-\mathsf{H}}\mathcal{C}(\sigma_3)^{\mathsf{H}},\\
W_5 &= \mathcal{K}(\sigma_1)^{-\mathsf{H}}\overline{\boldsymbol{\mathcal{H}}(\sigma_2,\sigma_1)^{(2)}}(\overline{V_3} \otimes W_4),\\
W_6 &= \mathcal{K}(\sigma_2)^{-\mathsf{H}}\overline{\boldsymbol{\mathcal{H}}(\sigma_2,\sigma_1)^{(3)}}(\overline{V_1} \otimes W_4),
\end{aligned}$$

where $\boldsymbol{\mathcal{H}}^{(2)}, \boldsymbol{\mathcal{H}}^{(3)}$ are the 2- and 3-mode matricizations of the tensor corresponding to the quadratic term such that $\boldsymbol{\mathcal{H}}^{(1)} = \mathcal{H}$, and $\boldsymbol{\mathcal{N}}^{(2)}$ is the 2-mode matricization of the tensor

corresponding to the bilinear term such that $\boldsymbol{\mathcal{N}}^{(1)} = \mathcal{N}$. Then the following statements hold:

(a) If V and W are constructed such that

$$\operatorname{span}(V) \supseteq \operatorname{span}(V_1) \quad \text{and} \quad \operatorname{span}(W) \supseteq \operatorname{span}(W_1),$$

then the following interpolation conditions are fulfilled:

$$\mathcal{G}^{(\mathrm{B})}_{\mathrm{Q,gen},1}(\sigma_1) = \widehat{\mathcal{G}}^{(\mathrm{B})}_{\mathrm{Q,gen},1}(\sigma_1), \quad \nabla\mathcal{G}^{(\mathrm{B})}_{\mathrm{Q,gen},1}(\sigma_1) = \nabla\widehat{\mathcal{G}}^{(\mathrm{B})}_{\mathrm{Q,gen},1}(\sigma_1),$$

and, additionally,

$$\begin{aligned}\mathcal{G}^{(\mathrm{N},(\mathrm{B}))}_{\mathrm{Q,gen},2}(\sigma_1,\sigma_1) &= \widehat{\mathcal{G}}^{(\mathrm{N},(\mathrm{B}))}_{\mathrm{Q,gen},2}(\sigma_1,\sigma_1),\\ \mathcal{G}^{(\mathrm{H},(\mathrm{B}),(\mathrm{B}))}_{\mathrm{Q,gen},3}(\sigma_1,\sigma_1,\sigma_1) &= \mathcal{G}^{(\mathrm{H},(\mathrm{B}),(\mathrm{B}))}_{\mathrm{Q,gen},3}(\sigma_1,\sigma_1,\sigma_1).\end{aligned}$$

(b) If V and W are constructed such that

$$\operatorname{span}(V) \supseteq \operatorname{span}\left(\begin{bmatrix} V_1 & V_2 \end{bmatrix}\right) \quad \text{and} \quad \operatorname{span}(W) \supseteq \operatorname{span}\left(\begin{bmatrix} W_2 & W_3 \end{bmatrix}\right),$$

then the following interpolation conditions are fulfilled:

$$\mathcal{G}^{(\mathrm{N},(\mathrm{B}))}_{\mathrm{Q,gen},2}(\sigma_1,\sigma_2) = \widehat{\mathcal{G}}^{(\mathrm{N},(\mathrm{B}))}_{\mathrm{Q,gen},2}(\sigma_1,\sigma_2), \quad \nabla\mathcal{G}^{(\mathrm{N},(\mathrm{B}))}_{\mathrm{Q,gen},2}(\sigma_1,\sigma_2) = \nabla\widehat{\mathcal{G}}^{(\mathrm{N},(\mathrm{B}))}_{\mathrm{Q,gen},2}(\sigma_1,\sigma_2),$$

and, additionally,

$$\begin{aligned}\mathcal{G}^{\gamma_{1,j_1}}_{\mathrm{Q,gen},1}(\omega_{1,j_1}) &= \widehat{\mathcal{G}}^{\gamma_{1,j_1}}_{\mathrm{Q,gen},1}(\omega_{1,j_1}), && j_1 = 1,2,\\ \mathcal{G}^{\gamma_{3,j_3}}_{\mathrm{Q,gen},3}(\omega_{3,j_3}) &= \widehat{\mathcal{G}}^{\gamma_{3,j_3}}_{\mathrm{Q,gen},3}(\omega_{3,j_3}), && j_3 = 1,2,3,\\ \mathcal{G}^{\gamma_{4,j_4}}_{\mathrm{Q,gen},4}(\omega_{4,j_4}) &= \widehat{\mathcal{G}}^{\gamma_{4,j_4}}_{\mathrm{Q,gen},4}(\omega_{4,j_4}), && j_4 = 1,2,3,4,\\ \mathcal{G}^{\gamma_{5,j_5}}_{\mathrm{Q,gen},5}(\omega_{5,j_5}) &= \widehat{\mathcal{G}}^{\gamma_{5,j_5}}_{\mathrm{Q,gen},5}(\omega_{5,j_5}), && j_5 = 1,2,3,\\ \mathcal{G}^{\gamma_6}_{\mathrm{Q,gen},6}(\omega_6) &= \widehat{\mathcal{G}}^{\gamma_6}_{\mathrm{Q,gen},6}(\omega_6),\end{aligned}$$

with

$$\begin{aligned}\gamma_{1,1} &= (\mathrm{B}), & \omega_{1,1} &= \sigma_1,\\ \gamma_{1,2} &= (\mathrm{B}), & \omega_{1,2} &= \sigma_2,\\ \gamma_{3,1} &= (\mathrm{N},(\mathrm{N},(\mathrm{B}))), & \omega_{3,1} &= (\sigma_1,\sigma_2,\sigma_2),\\ \gamma_{3,2} &= (\mathrm{N},(\mathrm{N},(\mathrm{B}))), & \omega_{3,2} &= (\sigma_1,\sigma_1,\sigma_2),\\ \gamma_{3,3} &= (\mathrm{H},(\mathrm{B}),(\mathrm{B})), & \omega_{3,3} &= (\sigma_1,\sigma_1,\sigma_2),\\ \gamma_{4,1} &= (\mathrm{N},(\mathrm{N},(\mathrm{N},(\mathrm{B})))), & \omega_{4,1} &= (\sigma_1,\sigma_2,\sigma_1,\sigma_2),\\ \gamma_{4,2} &= (\mathrm{H},(\mathrm{N},(\mathrm{B})),(\mathrm{B})), & \omega_{4,2} &= (\sigma_1,\sigma_1,\sigma_2,\sigma_2),\end{aligned}$$

$$
\begin{aligned}
\gamma_{4,3} &= (\mathrm{H},(\mathrm{B}),(\mathrm{N},(\mathrm{B}))), & \omega_{4,3} &= (\sigma_1,\sigma_2,\sigma_1,\sigma_2),\\
\gamma_{4,4} &= (\mathrm{N},(\mathrm{H},(\mathrm{B}),(\mathrm{B}))), & \omega_{4,4} &= (\sigma_1,\sigma_1,\sigma_1,\sigma_2),\\
\gamma_{5,1} &= (\mathrm{H},(\mathrm{N},(\mathrm{B})),(\mathrm{N},(\mathrm{B}))), & \omega_{5,1} &= (\sigma_1,\sigma_2,\sigma_1,\sigma_2,\sigma_2),\\
\gamma_{5,2} &= (\mathrm{N},(\mathrm{H},(\mathrm{N},(\mathrm{B})),(\mathrm{B}))), & \omega_{5,2} &= (\sigma_1,\sigma_1,\sigma_2,\sigma_1,\sigma_2),\\
\gamma_{5,3} &= (\mathrm{N},(\mathrm{H},(\mathrm{B}),(\mathrm{N},(\mathrm{B})))), & \omega_{5,3} &= (\sigma_1,\sigma_2,\sigma_1,\sigma_1,\sigma_2),\\
\gamma_{6} &= (\mathrm{N},(\mathrm{H},(\mathrm{N},(\mathrm{B})),(\mathrm{N},(\mathrm{B})))), & \omega_{6} &= (\sigma_1,\sigma_2,\sigma_1,\sigma_2,\sigma_1,\sigma_2).
\end{aligned}
$$

(c) If V and W are constructed such that

$$
\operatorname{span}(V) \supseteq \operatorname{span}\left(\begin{bmatrix} V_1 & V_3 & V_4 \end{bmatrix}\right) \quad \text{and} \quad \operatorname{span}(W) \supseteq \operatorname{span}\left(\begin{bmatrix} W_4 & W_5 & W_6 \end{bmatrix}\right),
$$

then the following interpolation conditions are fulfilled:

$$
\begin{aligned}
\mathcal{G}_{\mathrm{Q,gen},3}^{(\mathrm{H},(\mathrm{B}),(\mathrm{B}))}(\sigma_1,\sigma_2,\sigma_3) &= \mathcal{G}_{\mathrm{Q,gen},3}^{(\mathrm{H},(\mathrm{B}),(\mathrm{B}))}(\sigma_1,\sigma_2,\sigma_3),\\
\nabla\mathcal{G}_{\mathrm{Q,gen},3}^{(\mathrm{H},(\mathrm{B}),(\mathrm{B}))}(\sigma_1,\sigma_2,\sigma_3) &= \nabla\mathcal{G}_{\mathrm{Q,gen},3}^{(\mathrm{H},(\mathrm{B}),(\mathrm{B}))}(\sigma_1,\sigma_2,\sigma_3),
\end{aligned}
$$

and, additionally,

$$
\begin{aligned}
\mathcal{G}_{\mathrm{Q,gen},1}^{\gamma_{1,j_1}}(\omega_{1,j_1}) &= \widehat{\mathcal{G}}_{\mathrm{Q,gen},1}^{\gamma_{1,j_1}}(\omega_{1,j_1}), & j_1 &= 1,2,3,\\
\mathcal{G}_{\mathrm{Q,gen},2}^{\gamma_{2,j_2}}(\omega_{2,j_2}) &= \widehat{\mathcal{G}}_{\mathrm{Q,gen},2}^{\gamma_{2,j_2}}(\omega_{2,j_2}), & j_2 &= 1,2,\\
\mathcal{G}_{\mathrm{Q,gen},3}^{\gamma_{3,j_3}}(\omega_{3,j_3}) &= \widehat{\mathcal{G}}_{\mathrm{Q,gen},3}^{\gamma_{3,j_3}}(\omega_{3,j_3}), & j_3 &= 1,2,3\\
\mathcal{G}_{\mathrm{Q,gen},4}^{\gamma_{4,j_4}}(\omega_{4,j_4}) &= \widehat{\mathcal{G}}_{\mathrm{Q,gen},4}^{\gamma_{4,j_4}}(\omega_{4,j_4}), & j_4 &= 1,\ldots,5,\\
\mathcal{G}_{\mathrm{Q,gen},5}^{\gamma_{5,j_5}}(\omega_{5,j_5}) &= \widehat{\mathcal{G}}_{\mathrm{Q,gen},5}^{\gamma_{5,j_5}}(\omega_{5,j_5}), & j_5 &= 1,\ldots,12,\\
\mathcal{G}_{\mathrm{Q,gen},6}^{\gamma_{6,j_6}}(\omega_{6,j_6}) &= \widehat{\mathcal{G}}_{\mathrm{Q,gen},6}^{\gamma_{6,j_6}}(\omega_{6,j_6}), & j_6 &= 1,2,
\end{aligned}
$$

with

$$
\begin{aligned}
\gamma_{1,1} &= (\mathrm{B}), & \omega_{1,1} &= \sigma_1,\\
\gamma_{1,2} &= (\mathrm{B}), & \omega_{1,2} &= \sigma_2,\\
\gamma_{1,3} &= (\mathrm{B}), & \omega_{1,3} &= \sigma_3,\\
\gamma_{2,1} &= (\mathrm{N},(\mathrm{B})), & \omega_{2,1} &= (\sigma_1,\sigma_3),\\
\gamma_{2,2} &= (\mathrm{N},(\mathrm{B})), & \omega_{2,2} &= (\sigma_2,\sigma_3),\\
\gamma_{3,1} &= (\mathrm{H},(\mathrm{B}),(\mathrm{B})), & \omega_{3,1} &= (\sigma_2,\sigma_1,\sigma_3),\\
\gamma_{3,2} &= (\mathrm{H},(\mathrm{B}),(\mathrm{B})), & \omega_{3,2} &= (\sigma_1,\sigma_1,\sigma_3),\\
\gamma_{3,3} &= (\mathrm{H},(\mathrm{B}),(\mathrm{B})), & \omega_{3,3} &= (\sigma_2,\sigma_2,\sigma_3),\\
\gamma_{4,1} &= (\mathrm{N},(\mathrm{H},(\mathrm{B}),(\mathrm{B}))), & \omega_{4,1} &= (\sigma_1,\sigma_2,\sigma_3,\sigma_3),\\
\gamma_{4,2} &= (\mathrm{H},(\mathrm{N},(\mathrm{B})),(\mathrm{B})), & \omega_{4,2} &= (\sigma_1,\sigma_1,\sigma_2,\sigma_3),
\end{aligned}
$$

$$
\begin{aligned}
\gamma_{4,3} &= (\mathrm{H},(\mathrm{N},(\mathrm{B})),(\mathrm{B})), & \omega_{4,3} &= (\sigma_1,\sigma_2,\sigma_2,\sigma_3),\\
\gamma_{4,4} &= (\mathrm{H},(\mathrm{B}),(\mathrm{N},(\mathrm{B}))), & \omega_{4,4} &= (\sigma_1,\sigma_1,\sigma_2,\sigma_3),\\
\gamma_{4,5} &= (\mathrm{H},(\mathrm{B}),(\mathrm{N},(\mathrm{B}))), & \omega_{4,5} &= (\sigma_2,\sigma_1,\sigma_2,\sigma_3),\\
\gamma_{5,1} &= (\mathrm{H},(\mathrm{H},(\mathrm{B}),(\mathrm{B})),(\mathrm{B})), & \omega_{5,1} &= (\sigma_1,\sigma_1,\sigma_2,\sigma_3,\sigma_3),\\
\gamma_{5,2} &= (\mathrm{H},(\mathrm{H},(\mathrm{B}),(\mathrm{B})),(\mathrm{B})), & \omega_{5,2} &= (\sigma_2,\sigma_1,\sigma_2,\sigma_3,\sigma_3),\\
\gamma_{5,3} &= (\mathrm{H},(\mathrm{H},(\mathrm{B}),(\mathrm{B})),(\mathrm{B})), & \omega_{5,3} &= (\sigma_1,\sigma_1,\sigma_1,\sigma_2,\sigma_3),\\
\gamma_{5,4} &= (\mathrm{H},(\mathrm{H},(\mathrm{B}),(\mathrm{B})),(\mathrm{B})), & \omega_{5,4} &= (\sigma_1,\sigma_2,\sigma_1,\sigma_2,\sigma_3),\\
\gamma_{5,5} &= (\mathrm{H},(\mathrm{H},(\mathrm{B}),(\mathrm{B})),(\mathrm{B})), & \omega_{5,5} &= (\sigma_1,\sigma_1,\sigma_2,\sigma_2,\sigma_3),\\
\gamma_{5,6} &= (\mathrm{H},(\mathrm{H},(\mathrm{B}),(\mathrm{B})),(\mathrm{B})), & \omega_{5,6} &= (\sigma_1,\sigma_2,\sigma_2,\sigma_2,\sigma_3),\\
\gamma_{5,7} &= (\mathrm{H},(\mathrm{B}),(\mathrm{H},(\mathrm{B}),(\mathrm{B}))), & \omega_{5,7} &= (\sigma_1,\sigma_2,\sigma_3,\sigma_1,\sigma_3),\\
\gamma_{5,8} &= (\mathrm{H},(\mathrm{B}),(\mathrm{H},(\mathrm{B}),(\mathrm{B}))), & \omega_{5,8} &= (\sigma_1,\sigma_2,\sigma_3,\sigma_2,\sigma_3),\\
\gamma_{5,9} &= (\mathrm{H},(\mathrm{B}),(\mathrm{H},(\mathrm{B}),(\mathrm{B}))), & \omega_{5,9} &= (\sigma_1,\sigma_1,\sigma_1,\sigma_2,\sigma_3),\\
\gamma_{5,10} &= (\mathrm{H},(\mathrm{B}),(\mathrm{H},(\mathrm{B}),(\mathrm{B}))), & \omega_{5,10} &= (\sigma_2,\sigma_1,\sigma_1,\sigma_2,\sigma_3),\\
\gamma_{5,11} &= (\mathrm{H},(\mathrm{B}),(\mathrm{H},(\mathrm{B}),(\mathrm{B}))), & \omega_{5,11} &= (\sigma_1,\sigma_2,\sigma_1,\sigma_2,\sigma_3),\\
\gamma_{5,12} &= (\mathrm{H},(\mathrm{B}),(\mathrm{H},(\mathrm{B}),(\mathrm{B}))), & \omega_{5,12} &= (\sigma_2,\sigma_2,\sigma_1,\sigma_2,\sigma_3),\\
\gamma_{6,1} &= (\mathrm{H},(\mathrm{N},(\mathrm{H},(\mathrm{B}),(\mathrm{B}))),(\mathrm{B})), & \omega_{6,1} &= (\sigma_1,\sigma_1,\sigma_2,\sigma_3,\sigma_2,\sigma_3),\\
\gamma_{6,2} &= (\mathrm{H},(\mathrm{B}),(\mathrm{N},(\mathrm{H},(\mathrm{B}),(\mathrm{B})))), & \omega_{6,2} &= (\sigma_1,\sigma_2,\sigma_3,\sigma_1,\sigma_2,\sigma_3). \qquad \lozenge
\end{aligned}
$$

Proof. The first-level transfer function interpolation in Part (a) is an extract of Proposition 3.2, and the interpolation of the second-level transfer function with the bilinear term in Part (b) can be found in Corollary 5.11. In Part (c), only the interpolation of third- and higher-level generalized transfer functions needs to be proven since the first- and second-level interpolation results follow again from Proposition 3.2 and Section 5.4. The higher-level results are omitted for brevity since they follow the same arguments as used for the third-level transfer function. With the three matrices V_1, V_3 and W_4, the projection spaces corresponding to V and W satisfy the conditions in Lemma 6.15 such that

$$
\mathcal{G}_{\mathrm{Q,gen},3}^{(\mathrm{H},(\mathrm{B}),(\mathrm{B}))}(\sigma_1,\sigma_2,\sigma_3) = \mathcal{G}_{\mathrm{Q,gen},3}^{(\mathrm{H},(\mathrm{B}),(\mathrm{B}))}(\sigma_1,\sigma_2,\sigma_3)
$$

holds. Only the interpolation of the partial derivatives in the three frequency arguments is left to be proven since the Jacobian is given by

$$
\nabla \mathcal{G}_{\mathrm{Q,gen},3}^{(\mathrm{H},(\mathrm{B}),(\mathrm{B}))} = \begin{bmatrix} \partial_{s_1}\mathcal{G}_{\mathrm{Q,gen},3}^{(\mathrm{H},(\mathrm{B}),(\mathrm{B}))} & \partial_{s_2}\mathcal{G}_{\mathrm{Q,gen},3}^{(\mathrm{H},(\mathrm{B}),(\mathrm{B}))} & \partial_{s_3}\mathcal{G}_{\mathrm{Q,gen},3}^{(\mathrm{H},(\mathrm{B}),(\mathrm{B}))} \end{bmatrix}.
$$

The derivative with respect to s_3 is considered first, since this frequency argument does neither enter the quadratic term nor the corresponding Kronecker product. For the

reduced-order transfer function, this derivative is given by

$$\begin{aligned}\partial_{s_3}\widehat{\mathcal{G}}^{(\mathrm{H},(\mathrm{B}),(\mathrm{B}))}_{\mathrm{Q,gen},3}(\sigma_1,\sigma_2,\sigma_3) &= \partial_s\widehat{\mathcal{C}}(\sigma_3)\widehat{\mathcal{K}}(\sigma_3)^{-1}\widehat{\mathcal{H}}(\sigma_2,\sigma_1)\Big(\widehat{\mathcal{K}}(\sigma_2)^{-1}\widehat{\mathcal{B}}(\sigma_2)\otimes\widehat{\mathcal{K}}(\sigma_1)^{-1}\widehat{\mathcal{B}}(\sigma_1)\Big)\\ &\quad-\widehat{\mathcal{C}}(\sigma_3)\widehat{\mathcal{K}}(\sigma_3)^{-1}\partial_s\widehat{\mathcal{K}}(\sigma_3)\widehat{\mathcal{K}}(\sigma_3)^{-1}\widehat{\mathcal{H}}(\sigma_2,\sigma_1)\Big(\widehat{\mathcal{K}}(\sigma_2)^{-1}\widehat{\mathcal{B}}(\sigma_2)\\ &\quad\otimes\widehat{\mathcal{K}}(\sigma_1)^{-1}\widehat{\mathcal{B}}(\sigma_1)\Big).\end{aligned}$$

For the two terms in the subtraction, it holds

$$\begin{aligned}&\partial_s\widehat{\mathcal{C}}(\sigma_3)\widehat{\mathcal{K}}(\sigma_3)^{-1}\widehat{\mathcal{H}}(\sigma_2,\sigma_1)\Big(\widehat{\mathcal{K}}(\sigma_2)^{-1}\widehat{\mathcal{B}}(\sigma_2)\otimes\widehat{\mathcal{K}}(\sigma_1)^{-1}\widehat{\mathcal{B}}(\sigma_1)\Big)\\ &=\partial_s\mathcal{C}(\sigma_3)V\widehat{\mathcal{K}}(\sigma_3)^{-1}W^{\mathsf{H}}\mathcal{H}(\sigma_2,\sigma_1)\Big(V\widehat{\mathcal{K}}(\sigma_2)^{-1}W^{\mathsf{H}}\mathcal{B}(\sigma_2)\otimes V\widehat{\mathcal{K}}(\sigma_1)^{-1}W^{\mathsf{H}}\mathcal{B}(\sigma_1)\Big)\\ &=\partial_s\mathcal{C}(\sigma_3)P_{\mathrm{V}}(\sigma_3)\mathcal{K}(\sigma_3)^{-1}\mathcal{H}(\sigma_2,\sigma_1)\Big(P_{\mathrm{V}}(\sigma_2)V_3\otimes P_{\mathrm{V}}(\sigma_1)V_1\Big)\\ &=\partial_s\mathcal{C}(\sigma_3)P_{\mathrm{V}}(\sigma_3)V_4\\ &=\partial_s\mathcal{C}(\sigma_3)\mathcal{K}(\sigma_3)^{-1}\mathcal{H}(\sigma_2,\sigma_1)\Big(\mathcal{K}(\sigma_2)^{-1}\mathcal{B}(\sigma_2)\otimes\mathcal{K}(\sigma_1)^{-1}\mathcal{B}(\sigma_1)\Big),\end{aligned}$$

and

$$\begin{aligned}&\widehat{\mathcal{C}}(\sigma_3)\widehat{\mathcal{K}}(\sigma_3)^{-1}\partial_s\widehat{\mathcal{K}}(\sigma_3)\widehat{\mathcal{K}}(\sigma_3)^{-1}\widehat{\mathcal{H}}(\sigma_2,\sigma_1)\Big(\widehat{\mathcal{K}}(\sigma_2)^{-1}\widehat{\mathcal{B}}(\sigma_2)\otimes\widehat{\mathcal{K}}(\sigma_1)^{-1}\widehat{\mathcal{B}}(\sigma_1)\Big)\\ &=\mathcal{C}(\sigma_3)V\widehat{\mathcal{K}}(\sigma_3)^{-1}W^{\mathsf{H}}\partial_s\mathcal{K}(\sigma_3)V\widehat{\mathcal{K}}(\sigma_3)^{-1}W^{\mathsf{H}}\mathcal{H}(\sigma_2,\sigma_1)\\ &\quad\times\Big(V\widehat{\mathcal{K}}(\sigma_2)^{-1}\widehat{\mathcal{B}}(\sigma_2)\otimes V\widehat{\mathcal{K}}(\sigma_1)^{-1}\widehat{\mathcal{B}}(\sigma_1)\Big)\\ &=W_4^{\mathsf{H}}P_{\mathrm{W}}(\sigma_3)^{\mathsf{H}}\partial_s\mathcal{K}(\sigma_3)P_{\mathrm{V}}(\sigma_3)V_4\\ &=\mathcal{C}(\sigma_3)\mathcal{K}(\sigma_3)^{-1}\partial_s\mathcal{K}(\sigma_3)\mathcal{K}(\sigma_3)^{-1}\mathcal{H}(\sigma_2,\sigma_1)\Big(\mathcal{K}(\sigma_2)^{-1}\mathcal{B}(\sigma_2)\otimes\mathcal{K}(\sigma_1)^{-1}\mathcal{B}(\sigma_1)\Big),\end{aligned}$$

with the projectors (3.24) and (3.25) onto $\operatorname{span}(V)$ and $\operatorname{span}(W)$, respectively, which yields the interpolation of the partial derivative with respect to s_3. The other two partial derivatives are similar to each other in their structure but need different parts of the projection spaces for the interpolation. Consider first the derivative with respect to s_1, given by

$$\begin{aligned}\partial_{s_1}\widehat{\mathcal{G}}^{(\mathrm{H},(\mathrm{B}),(\mathrm{B}))}_{\mathrm{Q,gen},3}(\sigma_1,\sigma_2,\sigma_3) &= \widehat{\mathcal{C}}(\sigma_3)\widehat{\mathcal{K}}(\sigma_3)^{-1}\Big(\partial_{s_2}\widehat{\mathcal{H}}(\sigma_2,\sigma_1)\Big(\widehat{\mathcal{K}}(\sigma_2)^{-1}\widehat{\mathcal{B}}(\sigma_2)\otimes\widehat{\mathcal{K}}(\sigma_1)^{-1}\widehat{\mathcal{B}}(\sigma_1)\Big)\\ &\quad+\widehat{\mathcal{H}}(\sigma_2,\sigma_1)\Big(\widehat{\mathcal{K}}(\sigma_2)^{-1}\widehat{\mathcal{B}}(\sigma_2)\otimes\partial_s(\widehat{\mathcal{K}}^{-1}\widehat{\mathcal{B}})(\sigma_1)\Big)\Big).\end{aligned}$$

The two resulting terms in the sum need to be handled independently of each other. For the first one, it holds

$$\begin{aligned}&\widehat{\mathcal{C}}(\sigma_3)\widehat{\mathcal{K}}(\sigma_3)^{-1}\partial_{s_2}\widehat{\mathcal{H}}(\sigma_2,\sigma_1)\Big(\widehat{\mathcal{K}}(\sigma_2)^{-1}\widehat{\mathcal{B}}(\sigma_2)\otimes\widehat{\mathcal{K}}(\sigma_1)^{-1}\widehat{\mathcal{B}}(\sigma_1)\Big)\\ &=\mathcal{C}(\sigma_3)V\widehat{\mathcal{K}}(\sigma_3)^{-1}W^{\mathsf{H}}\partial_{s_2}\mathcal{H}(\sigma_2,\sigma_1)\Big(V\widehat{\mathcal{K}}(\sigma_2)^{-1}W^{\mathsf{H}}\mathcal{B}(\sigma_2)\otimes V\widehat{\mathcal{K}}(\sigma_1)^{-1}W^{\mathsf{H}}\mathcal{B}(\sigma_1)\Big)\\ &=W_4^{\mathsf{H}}P_{\mathrm{W}}(\sigma_3)\partial_{s_2}\mathcal{H}(\sigma_2,\sigma_1)\Big(P_{\mathrm{V}}(\sigma_2)V_3\otimes P_{\mathrm{V}}(\sigma_1)V_1\Big)\\ &=\mathcal{C}(\sigma_3)\mathcal{K}(\sigma_3)^{-1}\partial_{s_2}\mathcal{H}(\sigma_2,\sigma_1)\Big(\mathcal{K}(\sigma_2)^{-1}\mathcal{B}(\sigma_2)\otimes\mathcal{K}(\sigma_1)^{-1}\mathcal{B}(\sigma_1)\Big).\end{aligned}$$

The second term needs to be treated in a different way, therefore, let it be denoted by $\widehat{Z}_1$, and let $\widehat{\boldsymbol{Z}}_1$ be the associated tensor such that

$$\widehat{\boldsymbol{Z}}_1^{(1)} = \widehat{Z}_1 = \widehat{\mathcal{C}}(\sigma_3)\widehat{\mathcal{K}}(\sigma_3)^{-1}\widehat{\mathcal{H}}(\sigma_2,\sigma_1)\Big(\widehat{\mathcal{K}}(\sigma_2)^{-1}\widehat{\mathcal{B}}(\sigma_2) \otimes \partial_s(\widehat{\mathcal{K}}^{-1}\widehat{\mathcal{B}})(\sigma_1)\Big).$$

With the 2-mode matricization of this tensor and (2.2), it holds

$$\begin{aligned}
\overline{\widehat{\boldsymbol{Z}}_1^{(2)}} &= \partial_s(\widehat{\mathcal{K}}^{-1}\widehat{\mathcal{B}})^{\mathsf{H}}(\sigma_1)\overline{\widehat{\boldsymbol{\mathcal{H}}}^{(2)}(\sigma_2,\sigma_1)}\Big(\overline{\widehat{\mathcal{K}}(\sigma_2)^{-1}\widehat{\mathcal{B}}(\sigma_2)} \otimes \widehat{\mathcal{K}}(\sigma_3)^{-\mathsf{H}}\widehat{\mathcal{C}}(\sigma_3)^{\mathsf{H}}\Big) \\
&= \partial_s(\widehat{\mathcal{K}}^{-1}\widehat{\mathcal{B}})^{\mathsf{H}}(\sigma_1)V^{\mathsf{H}}\overline{\boldsymbol{\mathcal{H}}^{(2)}(\sigma_2,\sigma_1)}\Big(\overline{V\widehat{\mathcal{K}}(\sigma_2)^{-1}W^{\mathsf{H}}\mathcal{B}(\sigma_2)} \otimes W\widehat{\mathcal{K}}(\sigma_3)^{-\mathsf{H}}V^{\mathsf{H}}\mathcal{C}(\sigma_3)^{\mathsf{H}}\Big) \\
&= \partial_s(\widehat{\mathcal{K}}^{-1}\widehat{\mathcal{B}})^{\mathsf{H}}(\sigma_1)V^{\mathsf{H}}\overline{\boldsymbol{\mathcal{H}}^{(2)}(\sigma_2,\sigma_1)}\Big(\overline{P_{\mathrm{V}}(\sigma_2)V_3} \otimes P_{\mathrm{W}}(\sigma_3)W_4\Big) \\
&= \Big(\partial_s\widehat{\mathcal{B}}(\sigma_1)^{\mathsf{H}} - \widehat{\mathcal{B}}(\sigma_1)^{\mathsf{H}}\widehat{\mathcal{K}}^{-1}(\sigma_1)\partial_s\widehat{\mathcal{K}}(\sigma_1)\Big)\widehat{\mathcal{K}}(\sigma_1)^{-1}V^{\mathsf{H}}\mathcal{K}(\sigma_1)^{\mathsf{H}}W_5 \\
&= \Big(\partial_s\mathcal{B}(\sigma_1)^{\mathsf{H}} - V_1\partial_s\mathcal{K}(\sigma_1)\Big)P_{\mathrm{W}}(\sigma_1)W_5 \\
&= \partial_s(\mathcal{K}^{-1}\mathcal{B})^{\mathsf{H}}(\sigma_1)\overline{\boldsymbol{\mathcal{H}}^{(2)}(\sigma_2,\sigma_1)}\Big(\overline{\mathcal{K}(\sigma_2)^{-1}\mathcal{B}(\sigma_2)} \otimes \mathcal{K}(\sigma_3)^{-\mathsf{H}}\mathcal{C}(\sigma_3)^{\mathsf{H}}\Big).
\end{aligned}$$

This gives the interpolation of the partial derivative with respect to s_1. The interpolation with respect to s_2 follows analogously to s_1 but now by using W_6. □

Theorem 6.16 shows additionally to matching Hermite interpolation conditions the interpolation of generalized transfer functions up to level 6 in an implicit way. This strongly motivates the restriction of evaluations for the projection spaces to the lower transfer function levels, since a lot more can be matched implicitly. This advantage comes from idea of generalized transfer functions to only use products of the involved system terms. In comparison, the regular and symmetric transfer functions contain linear combinations of these terms, which complicates corresponding interpolation approaches.

As for the other two transfer function concepts, the subspace conditions in Theorem 6.16 simplify under some additional assumptions, namely, a special selection of interpolation points, the system being SISO and the quadratic term having an underlying symmetric tensor. Since these simplifications are less significant than in Theorems 6.5 and 6.11 and quickly follow from Theorem 6.16, they are only stated in the following corollary without further proof.

Corollary 6.17 (Implicit Hermite interpolation of gen. SISO TFs):
Let $\mathcal{G}_{\mathrm{Q}}$ be a quadratic-bilinear SISO system, described by its generalized transfer functions $\mathcal{G}^{\gamma}_{\mathrm{Q,gen},k}$ in (6.20) and (6.21), $\widehat{\mathcal{G}}_{\mathrm{Q}}$ the reduced-order quadratic-bilinear SISO system constructed by (6.22), with its reduced-order generalized transfer functions $\widehat{\mathcal{G}}^{\gamma}_{\mathrm{Q,gen},k}$, and $\sigma \in \mathbb{C}$ an interpolation point such that the matrix functions $\mathcal{C}$, $\mathcal{K}^{-1}$, $\mathcal{B}$, $\mathcal{N}$, $\mathcal{H}$ and $\widehat{\mathcal{K}}^{-1}$ are complex differentiable in this point. Also, let the tensor $\boldsymbol{\mathcal{H}}$, given by its 1-mode

matricization $\boldsymbol{\mathcal{H}}^{(1)} = \mathcal{H}$, be symmetric. Let the following vectors be given

$$\begin{aligned} v_1 &= \mathcal{K}(\sigma)^{-1}\mathcal{B}(\sigma), \\ v_2 &= \mathcal{K}(\sigma)^{-1}\mathcal{N}(\sigma)v_1, \\ v_3 &= \mathcal{K}(\sigma)^{-1}\mathcal{H}(\sigma,\sigma)(v_1 \otimes v_1), \end{aligned}$$

and, also,

$$\begin{aligned} w_1 &= \mathcal{K}(\sigma)^{-\mathsf{H}}\mathcal{C}(\sigma)^{\mathsf{H}}, \\ w_2 &= \mathcal{K}(\sigma)^{-\mathsf{H}}\mathcal{N}(\sigma)^{\mathsf{H}}w_1, \\ w_3 &= \mathcal{K}(\sigma)^{-\mathsf{H}}\overline{\boldsymbol{\mathcal{H}}(\sigma,\sigma)^{(2)}}(\overline{v_1} \otimes w_1). \end{aligned}$$

Then the following statements hold:

(a) If V and W are constructed such that

$$\operatorname{span}(V) \supseteq \operatorname{span}(v_1) \quad \text{and} \quad \operatorname{span}(W) \supseteq \operatorname{span}(w_1),$$

then the following interpolation conditions are fulfilled:

$$\begin{aligned} \mathcal{G}^{(\mathrm{B})}_{\mathrm{Q,gen},1}(\sigma) &= \widehat{\mathcal{G}}^{(\mathrm{B})}_{\mathrm{Q,gen},1}(\sigma), & \nabla\mathcal{G}^{(\mathrm{B})}_{\mathrm{Q,gen},1}(\sigma) &= \nabla\widehat{\mathcal{G}}^{(\mathrm{B})}_{\mathrm{Q,gen},1}(\sigma), \\ \mathcal{G}^{(\mathrm{N},(\mathrm{B}))}_{\mathrm{Q,gen},2}(\sigma,\sigma) &= \widehat{\mathcal{G}}^{(\mathrm{N},(\mathrm{B}))}_{\mathrm{Q,gen},2}(\sigma,\sigma), & \mathcal{G}^{(\mathrm{H},(\mathrm{B}),(\mathrm{B}))}_{\mathrm{Q,gen},3}(\sigma,\sigma,\sigma) &= \mathcal{G}^{(\mathrm{H},(\mathrm{B}),(\mathrm{B}))}_{\mathrm{Q,gen},3}(\sigma,\sigma,\sigma). \end{aligned} \tag{6.33}$$

(b) If V and W are constructed such that

$$\operatorname{span}(V) \supseteq \operatorname{span}\left(\begin{bmatrix} v_1 & v_2 \end{bmatrix}\right) \quad \text{and} \quad \operatorname{span}(W) \supseteq \operatorname{span}\left(\begin{bmatrix} w_1 & w_2 \end{bmatrix}\right),$$

then, additionally to (6.33), the following interpolation condition is fulfilled:

$$\nabla\mathcal{G}^{(\mathrm{N},(\mathrm{B}))}_{\mathrm{Q,gen},2}(\sigma,\sigma) = \nabla\widehat{\mathcal{G}}^{(\mathrm{N},(\mathrm{B}))}_{\mathrm{Q,gen},2}(\sigma,\sigma),$$

and, additionally,

$$\begin{aligned} \mathcal{G}^{(\mathrm{N},(\mathrm{N},(\mathrm{B})))}_{\mathrm{Q,gen},3}(\sigma,\sigma,\sigma) &= \widehat{\mathcal{G}}^{(\mathrm{N},(\mathrm{N},(\mathrm{B})))}_{\mathrm{Q,gen},3}(\sigma,\sigma,\sigma), \\ \partial_{s_2}\mathcal{G}^{(\mathrm{N},(\mathrm{N},(\mathrm{B})))}_{\mathrm{Q,gen},3}(\sigma,\sigma,\sigma) &= \partial_{s_2}\widehat{\mathcal{G}}^{(\mathrm{N},(\mathrm{N},(\mathrm{B})))}_{\mathrm{Q,gen},3}(\sigma,\sigma,\sigma), \\ \mathcal{G}^{(\mathrm{N},(\mathrm{N},(\mathrm{N},(\mathrm{B}))))}_{\mathrm{Q,gen},4}(\sigma,\sigma,\sigma,\sigma) &= \widehat{\mathcal{G}}^{(\mathrm{N},(\mathrm{N},(\mathrm{N},(\mathrm{B}))))}_{\mathrm{Q,gen},4}(\sigma,\sigma,\sigma,\sigma), \\ \mathcal{G}^{(\mathrm{H},(\mathrm{B}),(\mathrm{N},(\mathrm{B})))}_{\mathrm{Q,gen},4}(\sigma,\sigma,\sigma,\sigma) &= \widehat{\mathcal{G}}^{(\mathrm{H},(\mathrm{B}),(\mathrm{N},(\mathrm{B})))}_{\mathrm{Q,gen},4}(\sigma,\sigma,\sigma,\sigma), \\ \mathcal{G}^{(\mathrm{N},(\mathrm{H},(\mathrm{B}),(\mathrm{B})))}_{\mathrm{Q,gen},4}(\sigma,\sigma,\sigma,\sigma) &= \widehat{\mathcal{G}}^{(\mathrm{N},(\mathrm{H},(\mathrm{B}),(\mathrm{B})))}_{\mathrm{Q,gen},4}(\sigma,\sigma,\sigma,\sigma), \\ \mathcal{G}^{(\mathrm{H},(\mathrm{N},(\mathrm{B})),(\mathrm{N},(\mathrm{B})))}_{\mathrm{Q,gen},5}(\sigma,\sigma,\sigma,\sigma,\sigma) &= \widehat{\mathcal{G}}^{(\mathrm{H},(\mathrm{N},(\mathrm{B})),(\mathrm{N},(\mathrm{B})))}_{\mathrm{Q,gen},5}(\sigma,\sigma,\sigma,\sigma,\sigma), \\ \mathcal{G}^{(\mathrm{N},(\mathrm{H},(\mathrm{B}),(\mathrm{N},(\mathrm{B}))))}_{\mathrm{Q,gen},5}(\sigma,\sigma,\sigma,\sigma,\sigma) &= \widehat{\mathcal{G}}^{(\mathrm{N},(\mathrm{H},(\mathrm{B}),(\mathrm{N},(\mathrm{B}))))}_{\mathrm{Q,gen},5}(\sigma,\sigma,\sigma,\sigma,\sigma), \\ \mathcal{G}^{(\mathrm{N},(\mathrm{H},(\mathrm{N},(\mathrm{B})),(\mathrm{N},(\mathrm{B}))))}_{\mathrm{Q,gen},6}(\sigma,\sigma,\sigma,\sigma,\sigma,\sigma) &= \widehat{\mathcal{G}}^{(\mathrm{N},(\mathrm{H},(\mathrm{N},(\mathrm{B})),(\mathrm{N},(\mathrm{B}))))}_{\mathrm{Q,gen},6}(\sigma,\sigma,\sigma,\sigma,\sigma,\sigma). \end{aligned}$$

(c) If V and W are constructed such that

$$\operatorname{span}(V) \supseteq \operatorname{span}\left(\begin{bmatrix} v_1 & v_3 \end{bmatrix}\right) \quad \text{and} \quad \operatorname{span}(W) \supseteq \operatorname{span}\left(\begin{bmatrix} w_1 & w_3 \end{bmatrix}\right),$$

then, additionally to (6.33), the following interpolation condition is fulfilled:

$$\nabla \mathcal{G}^{(\mathrm{H},(\mathrm{B}),(\mathrm{B}))}_{\mathrm{Q},\mathrm{gen},3}(\sigma,\sigma,\sigma) = \nabla \mathcal{G}^{(\mathrm{H},(\mathrm{B}),(\mathrm{B}))}_{\mathrm{Q},\mathrm{gen},3}(\sigma,\sigma,\sigma),$$

and, additionally,

$$\begin{aligned}
\mathcal{G}^{(\mathrm{N},(\mathrm{H},(\mathrm{B}),(\mathrm{B})))}_{\mathrm{Q},\mathrm{gen},4}(\sigma,\sigma,\sigma,\sigma) &= \widehat{\mathcal{G}}^{(\mathrm{N},(\mathrm{H},(\mathrm{B}),(\mathrm{B})))}_{\mathrm{Q},\mathrm{gen},4}(\sigma,\sigma,\sigma,\sigma),\\
\mathcal{G}^{(\mathrm{H},(\mathrm{B}),(\mathrm{N},(\mathrm{B})))}_{\mathrm{Q},\mathrm{gen},4}(\sigma,\sigma,\sigma,\sigma) &= \widehat{\mathcal{G}}^{(\mathrm{H},(\mathrm{B}),(\mathrm{N},(\mathrm{B})))}_{\mathrm{Q},\mathrm{gen},4}(\sigma,\sigma,\sigma,\sigma),\\
\mathcal{G}^{(\mathrm{H},(\mathrm{B}),(\mathrm{H},(\mathrm{B}),(\mathrm{B})))}_{\mathrm{Q},\mathrm{gen},5}(\sigma,\sigma,\sigma,\sigma,\sigma) &= \widehat{\mathcal{G}}^{(\mathrm{H},(\mathrm{B}),(\mathrm{H},(\mathrm{B}),(\mathrm{B})))}_{\mathrm{Q},\mathrm{gen},5}(\sigma,\sigma,\sigma,\sigma,\sigma),\\
\partial_{s_3}\mathcal{G}^{(\mathrm{H},(\mathrm{B}),(\mathrm{H},(\mathrm{B}),(\mathrm{B})))}_{\mathrm{Q},\mathrm{gen},5}(\sigma,\sigma,\sigma,\sigma,\sigma) &= \partial_{s_3}\widehat{\mathcal{G}}^{(\mathrm{H},(\mathrm{B}),(\mathrm{H},(\mathrm{B}),(\mathrm{B})))}_{\mathrm{Q},\mathrm{gen},5}(\sigma,\sigma,\sigma,\sigma,\sigma),\\
\mathcal{G}^{(\mathrm{H},(\mathrm{B}),(\mathrm{N},(\mathrm{H},(\mathrm{B}),(\mathrm{B}))))}_{\mathrm{Q},\mathrm{gen},6}(\sigma,\sigma,\sigma,\sigma,\sigma,\sigma) &= \widehat{\mathcal{G}}^{(\mathrm{H},(\mathrm{B}),(\mathrm{N},(\mathrm{H},(\mathrm{B}),(\mathrm{B}))))}_{\mathrm{Q},\mathrm{gen},6}(\sigma,\sigma,\sigma,\sigma,\sigma,\sigma). \qquad \lozenge
\end{aligned}$$

In Corollary 6.17, formally less interpolation conditions are matched than in Theorem 6.16. This comes from the symmetry of $\boldsymbol{\mathcal{H}}$ and the single interpolation point σ, i.e., transfer functions that involve the quadratic term $\mathcal{H}$ are independent of the application order in the quadratic term and the associated Kronecker product, i.e.,

$$\Gamma\Big((\mathrm{H},\gamma_2,\gamma_3),\sigma,\ldots,\sigma\Big) = \Gamma\Big((\mathrm{H},\gamma_3,\gamma_2),\sigma,\ldots,\sigma\Big).$$

In other words, interpolation conditions in Corollary 6.17 that contain quadratic and bilinear terms actually count for two, e.g., from

$$\mathcal{G}^{(\mathrm{H},(\mathrm{B}),(\mathrm{N},(\mathrm{B})))}_{\mathrm{Q},\mathrm{gen},4}(\sigma,\sigma,\sigma,\sigma) = \widehat{\mathcal{G}}^{(\mathrm{H},(\mathrm{B}),(\mathrm{N},(\mathrm{B})))}_{\mathrm{Q},\mathrm{gen},4}(\sigma,\sigma,\sigma,\sigma),$$

it follows that

$$\mathcal{G}^{(\mathrm{H},(\mathrm{N},(\mathrm{B})),(\mathrm{B}))}_{\mathrm{Q},\mathrm{gen},4}(\sigma,\sigma,\sigma,\sigma) = \widehat{\mathcal{G}}^{(\mathrm{H},(\mathrm{N},(\mathrm{B})),(\mathrm{B}))}_{\mathrm{Q},\mathrm{gen},4}(\sigma,\sigma,\sigma,\sigma)$$

also holds. The choices in writing Corollary 6.17 remind of the results in [92] and the simplified representation of generalized transfer functions therein.

6.5 Numerical experiments

In this section, the developed theory from Section 6.4 is tested in numerical experiments with the Toda lattice model from Section 1.3.3. Therefore, the QBDAE version from Section 6.2.1 as well as the QBODE version from Section 6.2.2 of the Toda lattice are

considered separately. The model is used to describe a crystal structure with three different particle types and the system parameters are set to be

$$\begin{aligned} m_1 = m_4 = ... = 1, \quad & m_2 = m_5 = ... = 4, \quad & m_3 = m_6 = ... = 2, \\ \gamma_1 = \gamma_4 = ... = 0.1, \quad & \gamma_2 = \gamma_5 = ... = 0.2, \quad & \gamma_3 = \gamma_6 = ... = 0.3, \\ k_1 = k_4 = ... = 1, \quad & k_2 = k_5 = ... = 0.5, \quad & k_3 = k_6 = ... = 1.5, \end{aligned}$$

with $n_2 = 5\,000$ particles in the crystal. The system is modeled to be SISO with excitation of the first five particles as input and measurement of the summed velocities of the 8-th til 10-th particle as output.

For the model reduction, only one-sided projections are used to preserve mechanical components of the quadratic-bilinearizations. On the one hand, Theorem 6.2 is used for the interpolation of the first and second symmetric subsystem transfer functions $\mathcal{G}_{\mathrm{Q,sym},k}$ in identical interpolation points $\sigma_1 = \sigma_2 = \sigma$, and further denoted by SymInt. On the other hand, Theorem 6.13 is applied for the interpolation of the first-, second- and third-level generalized transfer functions with the nested tuples $\gamma_1 = (\mathrm{B})$, $\gamma_2 = (\mathrm{N}, (\mathrm{B}))$ and $\gamma_3 = (\mathrm{H}, (\mathrm{B}), (\mathrm{B}))$ and identical interpolation points $\sigma_1 = \sigma_2 = \sigma_3 = \sigma$, further on denoted as GenInt. The case of regular subsystem transfer functions is omitted since for the default choice (6.29), the constructed right projection space is identical to the one for the symmetric transfer function interpolation. Also, the alternative choice described in Remark 6.9 cannot be used since the quadratic matrix pencils of the linear system parts in QBDAE and QBODE have eigenvalues in 0 and, therefore, cannot be evaluated in that point. The reduced-order models are then generated similar to Section 5.5.4 with either logarithmically equidistant interpolation points in the frequency range $[10^{-2}, 10^{2}]$ rad/s, denoted by (equi.), or by an oversampling procedure and truncation via the pivoted QR decomposition to the appropriate dimension, denoted by (avg.).

A foreseeable problem will be the preservation of stability. A standard truncation of the block structures in the spring and damper matrices in (6.6) and (6.8) via an arbitrary basis matrix will very likely corrupt the relation of the original state and substitution variables. This might introduce unstable eigenvalues in the linear system components and lead to unstable simulations using the reduced-order models. An approach to tackle this problem by preserving the block structure of the system matrices is known as *split congruence transformation*; see, e.g., [88]. Therefore, let $V \in \mathbb{R}^{2n_2 \times r}$ be any constructed truncation matrix. In the split congruence transformation, the basis matrix is separated into the parts corresponding to the different types of state variables and then rearranged in block diagonal form for the one-sided projection, i.e., the new truncation matrix $\widetilde{V} \in \mathbb{R}^{2n_2 \times 2r}$ is constructed such that

$$\widetilde{V} = \begin{bmatrix} V_1 & 0 \\ 0 & V_2 \end{bmatrix}, \quad \text{with } V = \begin{bmatrix} V_1 \\ V_2 \end{bmatrix}. \tag{6.34}$$

Table 6.1: Relative approximation errors for the QBDAE Toda lattice example with reduced orders $r_2 = 36$ or $r_2 = 72$.

Method	L_2	L_∞	$\mathcal{H}_\infty^{(1)}$	$\mathcal{H}_\infty^{(2,\mathrm{sym})}$	$\mathcal{H}_\infty^{(2,\mathrm{reg})}$
SymInt(equi.)	∞	∞	8.623e-02	4.621e-01	4.612e-01
SymInt2(equi.)	9.906e-03	1.360e-02	5.269e-02	4.675e-01	5.454e-01
SymInt(avg.)	∞	∞	1.317e-01	6.446e-01	1.257e+01
SymInt2(avg.)	1.338e-01	2.899e-01	4.642e-02	3.563e-01	3.382e-01
GenInt(equi.)	∞	∞	3.024e-01	3.185e+00	2.497e+00
GenInt2(equi.)	4.874e-03	1.137e-02	2.109e-02	2.076e-01	2.297e-01
GenInt(avg.)	∞	∞	2.356e-01	1.969e+00	2.558e+00
GenInt2(avg.)	3.302e-02	5.547e-02	8.171e-03	1.095e-01	1.041e-01

An important observation is that by construction of (6.34), it holds

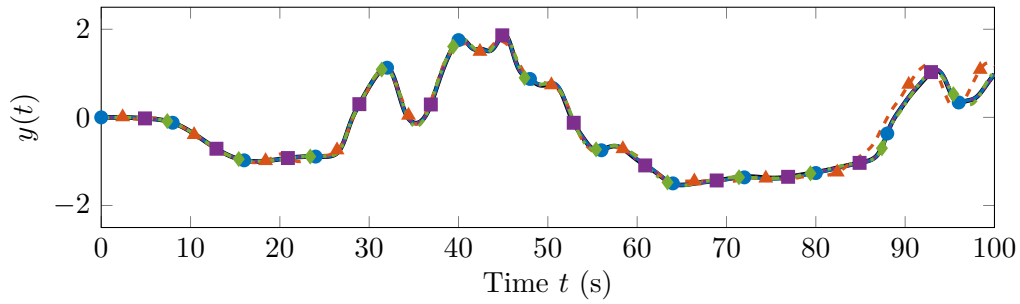
$$\operatorname{span}(V) \subseteq \operatorname{span}(\widetilde{V}),$$

and, therefore, transfer function interpolation can be preserved. But note that the constructed reduced-order model will be twice as large as when matching only the interpolation conditions due to the additional preservation of the block structure. For all model reduction methods above, a split congruence transformation-based version is also computed and denoted with an additional 2 in the name. For example, SymInt2(equi.) will denote the reduced-order model that interpolates the symmetric transfer functions in logarithmically equidistant points and was computed via (6.34).

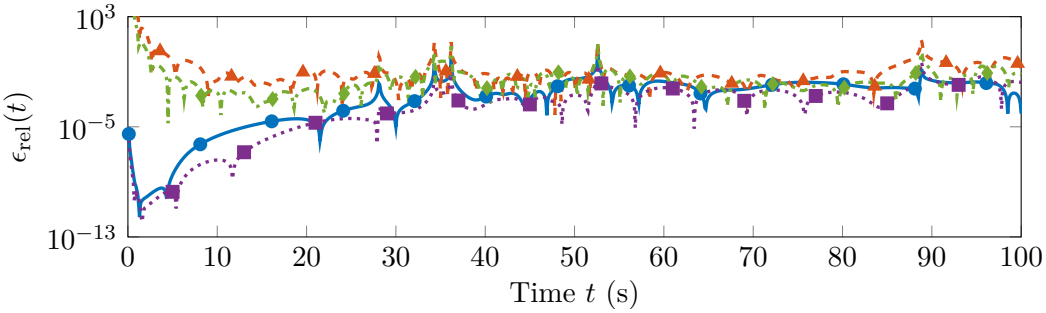
For the comparisons in the two following sections, the reduced-order models are computed for a fixed chosen order and the relative approximation errors are given in the approximate norms in time and frequency domain based on (2.44)–(2.46) and (5.16). For the time domain norms, simulations in the interval $[0, 100]$ s were computed using the input signal

$$u(t) = \eta(t_j), \quad \text{for } t_j \le t < t_{j+1}, \tag{6.35}$$

with $j = 0, \ldots, 99$, equidistant time steps $t_j = j \cdot \frac{100}{99}$ and presampled Gaussian white noise $\eta(t)$. Due to the different transfer function concepts, $\mathcal{H}_\infty^{(1)}$ will denote the relative $\mathcal{H}_\infty$-error for the linear transfer function, $\mathcal{H}_\infty^{(2,\mathrm{sym})}$ the extension of (5.16) to the second symmetric subsystem transfer function, and $\mathcal{H}_\infty^{(2,\mathrm{reg})}$ the extension to the second regular subsystem transfer function. Also, the usual pointwise error measures (4.19) and (4.20) will be used in plots.



(a) Time simulation.



(b) Pointwise relative errors.

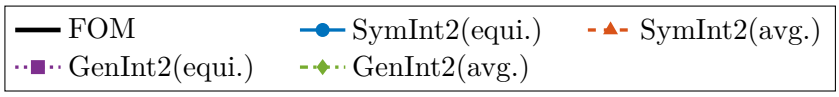
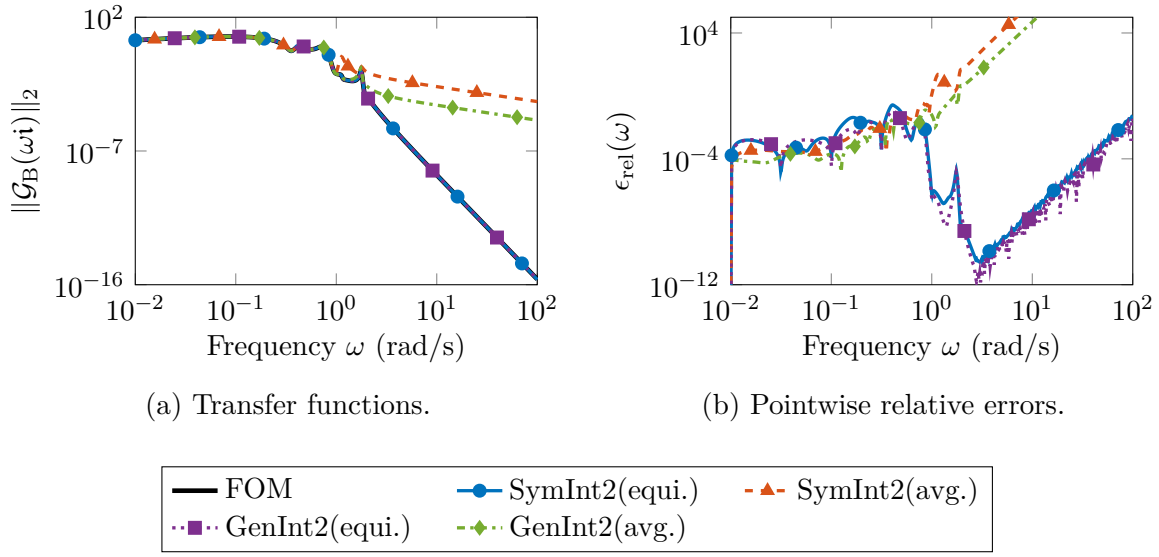


Figure 6.1: Time domain results for the QBDAE Toda lattice example.

6.5.1 Toda lattice QBDAE version

As first example, the QBDAE version of Toda lattice model (6.6) is considered. The reduced order was set to $r_2 = 36$ for the basic interpolation methods such that the split-congruence-based models are of order $r_2 = 72$. The resulting relative approximation errors are shown in Table 6.1. A time domain error being infinity implies that the reduced-order model did not produce a stable time simulation for the given input signal (6.35). In that sense, only the split-congruence-based reduced-order models were successfully used in the simulations. Comparing the different methods, SymInt2(equi.) and GenInt2(equi.) performed best in the time domain and StrInt2(avg.) and GenInt2(avg.) provided reasonably small errors in the frequency domain $\mathcal{H}_\infty$-norms.

Figure 6.1 shows the time domain results for the split-congruence-based methods. In Figure 6.1a, the amplitude of the output signal is shown and only the averaged subspace methods differ from the original system's behavior in the eyeball-norm. Looking at the



(a) Transfer functions.

(b) Pointwise relative errors.

Figure 6.2: First subsystem transfer functions and approximation errors for the QBDAE Toda lattice example.

pointwise relative errors in Figure 6.1b, the interpolation methods with equidistant points clearly perform better than the averaged subspace approaches. But SymInt2(avg.) and GenInt2(avg.) are able to provide a very constant error behavior over the simulated time range except at the beginning.

The results for the linear transfer functions are shown in Figure 6.2. It becomes clear that the small relative $\mathcal{H}_\infty$-errors in the averaged subspaces basically resulted from a good approximation of the transfer function region with large magnitudes. SymInt2(avg.) and GenInt2(avg.) basically fail to approximate the transfer function behavior for higher frequencies. The pointwise relative errors in Figure 6.2b reveal all methods to have a similar error behavior for low frequencies. But only SymInt2(equi.) and GenInt2(equi.) are capable of keeping this error level and even improving the approximation quality for higher frequencies due to enforced interpolation in this frequency region. While those results are not explicitly shown here, a very similar behavior in the frequency domain can be seen for the reduced-order models without the split congruence transformation ($r_2 = 36$). This is not surprising since the classical and split-congruence-based models are based on the same interpolation conditions for the construction of the projection spaces.

6.5.2 Toda lattice QBODE version

As second numerical example, the QBODE version of the Toda lattice model (6.8) is considered. It turned out that this model was more complicated to reduce than the QBDAE version. Therefore, the reduced order was raised to $r_2 = 60$ in the basic

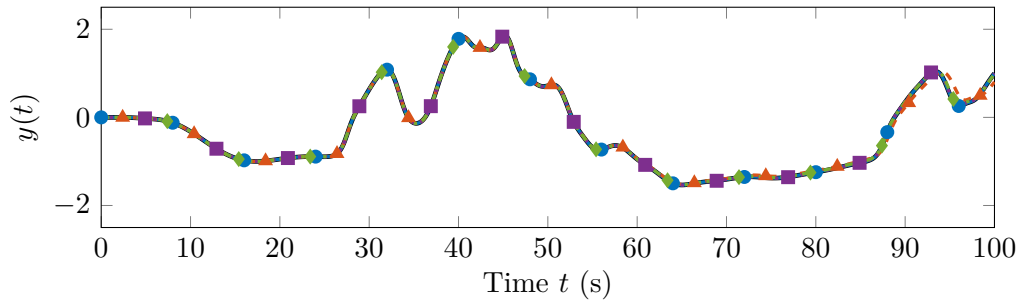
Table 6.2: Relative approximation errors for the QBODE Toda lattice example with reduced orders $r_2 = 60$ or $r_2 = 120$.

Method	L_2	L_∞	$\mathcal{H}_\infty^{(1)}$	$\mathcal{H}_\infty^{(2,\mathrm{sym})}$	$\mathcal{H}_\infty^{(2,\mathrm{reg})}$
SymInt(equi.)	∞	∞	5.792e-02	4.116e-01	2.916e-01
SymInt2(equi.)	9.478e-05	2.342e-04	2.979e-03	3.634e-02	1.596e-02
SymInt(avg.)	∞	∞	2.503e-02	2.278e-01	2.327e+01
SymInt2(avg.)	6.206e-02	2.106e-01	1.304e-03	1.888e-02	1.619e-02
GenInt(equi.)	∞	∞	1.886e-01	1.098e+00	4.346e-01
GenInt2(equi.)	5.270e-03	9.482e-03	9.601e-02	1.137e-01	6.487e-02
GenInt(avg.)	∞	∞	1.195e-01	7.094e-01	5.414e+01
GenInt2(avg.)	1.142e-02	3.866e-02	2.090e-03	9.805e-03	8.086e-02

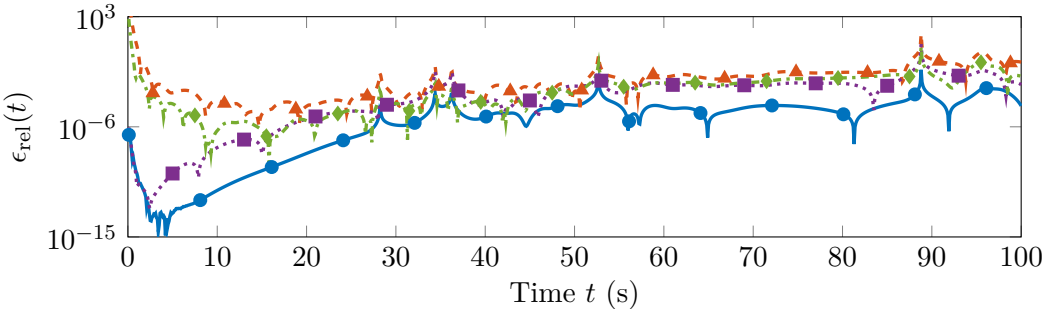
interpolation approach and, consequently, to $r_2 = 120$ for the split-congruence-based methods. This order was chosen such that all constructed reduced-order models based on the split congruence transformation provided a stable time simulation behavior. The resulting relative approximation errors are shown in Table 6.2. It is not surprising that due to the increased reduced orders most of the relative errors are several orders of magnitude smaller than those in Table 6.1. But this time, the symmetric transfer function interpolation performs exceptionally better than the generalized transfer function interpolation as the relative approximation errors are one to two orders of magnitude smaller. For the $\mathcal{H}_\infty$-errors in the frequency domain, SymInt2(equi.) is even compatible with the results of the averaged subspace approaches.

The time simulation results for the split-congruence-based methods are shown in Figure 6.3. In the eyball-norm, only SymInt2(avg.) misses a bit of the time simulation behavior of the full-order system at the end of the time range in Figure 6.3a. The relative pointwise errors (Figure 6.3b) look very similar to the results in the QBDAE case, with SymInt2(equi.) and GenInt2(equi.) best performing followed by GenInt2(avg.) and SymInt2(avg.) with a flatter error behavior. The clear winner of the comparison is SymInt2(equi.).

In frequency domain, with the linear transfer functions shown in Figure 6.4, similar results to the QBDAE case are obtained. While this time the approximation quality of SymInt2(avg.) and GenInt2(avg.) increased significantly, both approaches still fail to reflect the transfer function behavior for higher frequencies. However, this is again provided by SymInt2(equi.) and GenInt2(equi.) due to interpolation points in that frequency region. Similar results are obtained for the corresponding methods without the split congruence transformation ($r_2 = 60$), which are not plotted in Figure 6.4.



(a) Time simulation.



(b) Pointwise relative errors.

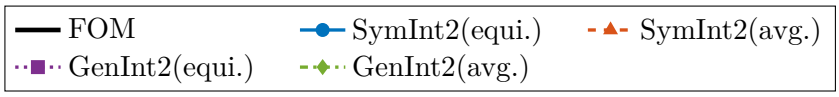
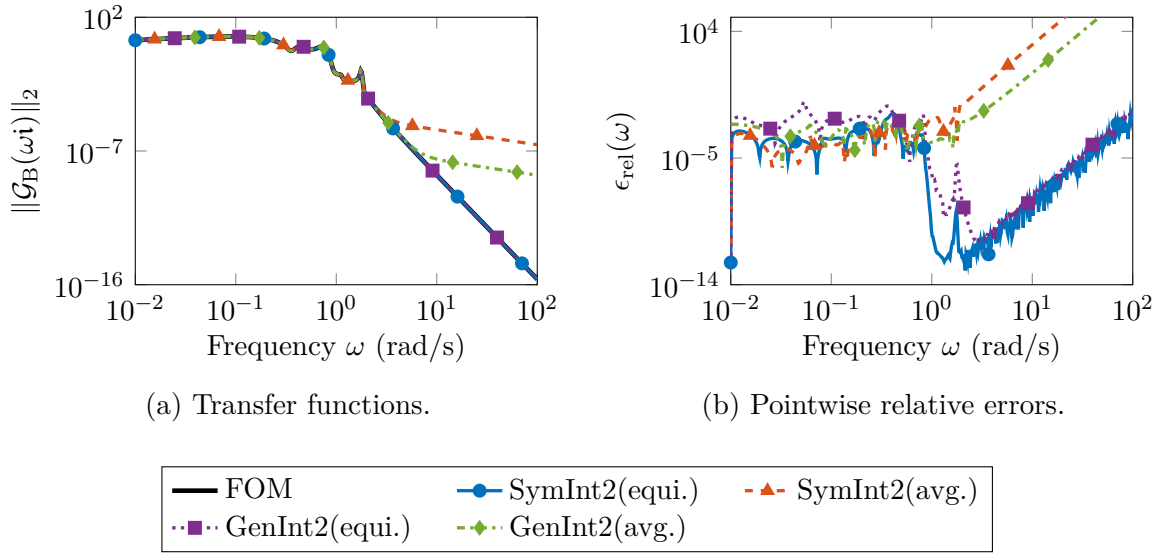


Figure 6.3: Time domain results for the QBODE Toda lattice example.

6.6 Conclusions

In this chapter, the case of structured nonlinear systems was considered, with a particular focus on nonlinear mechanical systems. To transform the general nonlinearities into an easier manageable form, the process of quadratic-bilinearization, known from the unstructured first-order system case, was outlined. It was used to derive systems of quadratic-bilinear differential-algebraic and ordinary differential equations for the motivating initial example of the nonlinear Toda lattice model. Then, the three known transfer function concepts of quadratic-bilinear systems were extended to the structured system setting using the example of quadratic-bilinear mechanical systems for motivation. As a result, new formulations of the symmetric, regular and generalized subsystem transfer functions were proposed for the case of structured quadratic-bilinear MIMO systems. In a similar fashion to Chapter 5, structured interpolation theory was developed for all three different transfer function concepts. This included results generalizing a



(a) Transfer functions.

(b) Pointwise relative errors.

Figure 6.4: First subsystem transfer functions and approximation errors for the QBODE Toda lattice example.

lot of theory known from the literature about unstructured quadratic-bilinear systems as well as formulations for the special case of SISO systems with underlying symmetric tensors for the quadratic terms. The theory was tested in numerical experiments on the two quadratic-bilinear versions (QBDAE and QBODE) of the Toda lattice model. Concerning the numerical approximation results, one can say that the additional step from DAEs to ODEs in the quadratic-bilinearization of the Toda lattice did not pay off significantly. This is seen in the more complex structure in the QBODE case leading to larger reduced-order models to provide stable simulations in comparison to the QBDAE case. However, in practice, a careful consideration of the different possible formulations resulting from the quadratic-bilinearization process is necessary. In both cases that were presented here, it was possible to construct reasonably small, structured reduced-order models that could be used as surrogates in numerical computations. Using the split congruence transformation approach, it was possible to retain even parts of the block structure of the system matrices, which allowed to compute structure-preserving, interpolating reduced-order models, which also provided stable time simulations.

CHAPTER 7

CONCLUSIONS

Contents

7.1 Summary

This thesis investigates new techniques for the problem of structure-preserving model order reduction for different kinds of mechanical systems described by differential equations involving second-order time derivatives. The contributions described in this thesis are of theoretical as well as of computational nature.

In Chapter 4, the case of linear mechanical systems is studied. In the first part, a new structure-preserving dominant pole algorithm has been developed for the special case of modally damped second-order systems. A new definition of dominant pole pairs is used to derive a numerical procedure that preserves the internal system structure in all computational steps and to develop error bounds in the $\mathcal{H}_\infty$-norm implying good approximations via dominant poles if the computed dominance measure decays fast enough. In contrast to related approaches from the literature, the new method takes the complete system structure into account leading to an efficient model reduction algorithm, for which even a structure-preserving extension has been proposed to tackle occurring problems of the approximation quality using structured interpolation. Numerical experiments have revealed the new dominant pole methods to easily outperform the classical modal truncation approach, as well as to be compatible with other types of structure-preserving model reduction methods as, e.g., second-order balanced truncation, in terms of approximation quality.

In practical applications, often only limited ranges in frequency or time domain are of interest. Under the observation that localized reduced-order models can be more

accurate or, alternatively, smaller than global approximations, in the second part of Chapter 4, new structure-preserving balanced truncation methods for the limited model reduction of linear second-order systems have been proposed. These methods are based on an extension of the definition of limited system Gramians to second-order systems, which generalizes ideas known from the literature. The application of the methods to large-scale sparse systems becomes feasible by using appropriate solvers for the occurring matrix equations with right-hand side matrix functions. The application of these solvers to mechanical systems is heavily improved by using an appropriate underlying first-order realization, namely the strictly dissipative realization. In general, reduced-order models computed by limited balanced truncation methods tend to be unstable already in the first-order system case. Therefore, alternative approaches were suggested replacing the fully limited system Gramians by their infinite counterparts or by modified versions. Also, the misconception from the literature is clarified that even these modifications cannot guarantee stable reduced-order models in the general case. The capabilities of the new structure-preserving limited model reduction methods is tested in numerical experiments, which show the new approaches to be appropriate tools for localized approximation problems.

A first step into the direction of describing nonlinear physical phenomena are bilinear control systems. In Chapter 5, general structured bilinear systems has been considered using the case of mechanical systems as motivation. Based on the Volterra series expansion of bilinear systems, a new concept of structured subsystem transfer functions has been developed. This new framework allows the representation of structured bilinear systems in the frequency domain, and enables the development of structure-preserving model reduction methods for bilinear systems. The usual question in projection-based model reduction is the choice of the projection spaces. Here, transfer function interpolation is suggested as it has been proven to be an efficient tool for the construction of reduced-order models in the linear and unstructured bilinear case. Similar to [24], a variety of interpolation results has been developed for scalar and matrix interpolation of structured subsystem transfer functions. In contrast to the existing interpolation theory for bilinear systems in the literature, the developed results hold for a large variety of internal system structures. In numerical experiments, the new structure-preserving interpolation is compared to existing model reduction methods that produce unstructured bilinear systems. The structured interpolation turned out to be the superior approach in terms of numerical costs as well as approximation quality. The considered system class is then further extended by providing interpolation results for parametric structured bilinear systems. A common problem in case of MIMO dynamical systems is the fast growth of the reduced-order models to satisfy matrix interpolation conditions. Especially in the bilinear system case with the multivariate subsystem transfer functions, the projection space dimensions grow exponentially fast with the transfer function level. Therefore, a new tangential interpolation framework has been developed that efficiently tackles this problem. In contrast to an existing approach for blockwise tangential interpolation

for unstructured bilinear systems from the literature [31, 160], the new framework is capable of restricting the projection space dimensions to only linear growth with the transfer function level, and can also handle structured bilinear systems. Also, due to the generality of the new framework, it can be used to easily generalize the blockwise tangential interpolation from the literature to the structured transfer function setting. Numerical examples have verified the new tangential interpolation framework to be an efficient tool for model reduction of structured bilinear MIMO systems.

Following the bilinear system case in Chapter 5, the case of structured nonlinear systems has been discussed in Chapter 6, with a strong focus on nonlinear mechanical systems. The process of quadratic-bilinearization has been used to simplify system nonlinearities into quadratic-bilinear form. Based on that, the different concepts for subsystem transfer functions from the literature have been generalized from first-order unstructured systems to a new structured setting, which allows for a large variety of internal system structures such as second-order time derivatives arising in the mechanical system case. A new transfer function interpolation framework has been developed for the three considered transfer function types of quadratic-bilinear systems, namely symmetric, regular and generalized transfer functions. These results are, on the one hand, a generalization of known theory from the literature to the structured system case. On the other hand, they generalize previous interpolation theory in terms of system assumptions, since the complete theory is formulated for the MIMO system case, as well as very often the tensor associated with the quadratic terms is not assumed to be symmetric.

7.2 Future research perspectives

A common problem that was repeatedly mentioned in Chapters 4 to 6 is the preservation of stability in the constructed reduced-order models. Already for linear systems, this problem is only solved in very particular cases or by certain model reduction methods, e.g., using a balanced truncation-based approach for first-order systems or using only a one-sided projection in case of systems with dissipative generalized or quadratic matrix pencils. This amounts even further to the question of appropriate stability concepts in the structured bilinear and nonlinear system cases and what could be preserved in the model reduction process. However, further research is needed with respect to structured transfer functions for all the different system classes and the stability of the corresponding dynamical systems.

While for the dominant pole-based method in the first part of Chapter 4 an error bound in the $\mathcal{H}_\infty$-norm has been developed, the general problem of bounding the approximation error remains open for the limited second-order balanced truncation approaches in the second part of Chapter 4. So far no error bounds are known even for the global (classical) second-order balanced truncation methods. Also, in case of the first-order limited balanced truncation methods, no error bounds for the local approximations in frequency

or time domain have been developed so far. Only for the modified frequency-limited balanced truncation, an a-priori error bound is provided in the $\mathcal{H}_\infty$-norm for the global approximation behavior [102]. In principle, a-posteriori error computations would be possible. For frequency-limited methods, $\mathcal{H}_\infty$-norm computation methods [6–8] can be used to determine the local approximation quality. There are also generalizations of the $\mathcal{H}_2$-norm to the frequency- and time-limited cases that could be computed; see, e.g., [94, 153]. However, the computation of a-posteriori errors is usually hard and corresponding numerical algorithms sometimes behave unreliably. In the frequency limited case, an alternative is given by error estimators like in [82] that can be used to efficiently bound the approximation errors in a localized version of the $\mathcal{H}_\infty$-norm.

Drawing the attention to the new structured interpolation frameworks for bilinear and quadratic-bilinear systems in Chapters 5 and 6 a question arises, which also exists for the linear system case: What is a good or even optimal choice of interpolation points for a given structured system? This question becomes even more involved when dealing with the multivariate transfer functions of bilinear and quadratic-bilinear systems that need sequences of interpolation points to be selected. In this thesis, heuristics inspired by the linear system case have been used like the greedy $\mathcal{H}_\infty$-selection or the points resulting from TF-IRKA for the linear system components. These have been shown to work well in the numerical examples. However, the question of optimal or better selections of interpolation points remains open. The optimal interpolation point selection problem has been solved only in the $\mathcal{H}_2$-norm for selected structures in the linear system case; see, e.g., [22, 103, 144, 155, 157]. Also in case of unstructured bilinear and quadratic-bilinear systems, optimal interpolation points for $\mathcal{H}_2$-norm minimization are known but only in the framework of Volterra series interpolation; see, e.g., [29, 40, 85, 190]. A remedy could be seen in the extension of the work on error estimators [82, 121], but in general, further research is needed for the different structured nonlinear system types. In the same sense, what is practically needed are bounds or estimates on the resulting approximation errors in suitable norms for model reduction of bilinear and quadratic-bilinear system. Thinking of the new tangential interpolation framework for bilinear systems in Section 5.6, similarly to the problem of interpolation point selection, also appropriate tangential directions as well as scaling vectors need to be found. In the experiments of this thesis, random tangential directions and the $\mathcal{H}_2$-optimal directions from the linear TF-IRKA have been used. The scaling vectors were chosen according to different motivations in frequency and time domain. However, the question of heuristic or even optimal directions and scaling vectors for the tangential subsystem interpolation of structured bilinear systems remains open.

BIBLIOGRAPHY

[1] M. I. Ahmad, P. Benner, and L. Feng, *A new two-sided projection technique for model reduction of quadratic-bilinear descriptor systems*, International Journal of Computer Mathematics, 96 (2019), pp. 1899–1909, https://doi.org/10.1080/00207160.2018.1542134. 183

[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), pp. 303–318, https://doi.org/10.1016/j.cam.2016.11.009. 110

[3] M. I. Ahmad, P. Benner, P. Goyal, and J. Heiland, *Moment-matching based model reduction for Navier-Stokes type quadratic-bilinear descriptor systems*, Z. Angew. Math. Mech., 97 (2017), pp. 1252–1267, https://doi.org/10.1002/zamm.201500262. 183

[4] M. I. Ahmad, P. Benner, and I. Jaimoukha, *Krylov subspace projection methods for model reduction of quadratic-bilinear systems*, IET Control Theory & Applications, 10 (2016), pp. 2010–2018, https://doi.org/10.1049/iet-cta.2016.0415. 14, 28, 183, 213, 219

[5] S. Al-Baiyat, A. S. Farag, and M. Bettayeb, *Transient approximation of a bilinear two-area interconnected power system*, Electric Power Systems Research, 26 (1993), pp. 11–19, https://doi.org/10.1016/0378-7796(93)90064-L. 110

[6] N. Aliyev, P. Benner, E. Mengi, P. Schwerdtner, and M. Voigt, *A greedy subspace method for computing the $\mathcal{L}_\infty$-norm*, Proc. Appl. Math. Mech., 17 (2017), pp. 751–752, https://doi.org/10.1002/pamm.201710343. 42, 47, 69, 246

[7] N. Aliyev, P. Benner, E. Mengi, P. Schwerdtner, and M. Voigt, *Large-scale computation of $\mathcal{L}_\infty$-norms by a greedy subspace method*, SIAM J. Matrix Anal. Appl., 38 (2017), pp. 1496–1516, https://doi.org/10.1137/16M1086200. 42, 47, 69, 246

[8] N. Aliyev, P. Benner, E. Mengi, and M. Voigt, *A subspace framework for $\mathcal{H}_\infty$-norm minimization*, SIAM J. Matrix Anal. Appl., 41 (2020), pp. 928–956, https://doi.org/10.1137/19M125892X. 42, 47, 69, 246

[9] A. C. Antoulas, *Approximation of Large-Scale Dynamical Systems*, vol. 6 of Adv. Des. Control, SIAM Publications, Philadelphia, PA, 2005, https://doi.org/10.1137/1.9780898718713. 15, 16, 17, 18, 36, 41

[10] A. C. Antoulas, C. A. Beattie, and S. Gugercin, *Interpolatory Methods for Model Reduction*, Computational Science & Engineering, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2020, https://doi.org/10.1137/1.9781611976083. 15, 40, 110, 116, 120, 136, 141

[11] A. C. Antoulas, P. Benner, and L. Feng, *Model reduction by iterative error system approximation*, Math. Comput. Model. Dyn. Syst., 24 (2018), pp. 103–118, https://doi.org/10.1080/13873954.2018.1427116. 42

[12] A. C. Antoulas, I. V. Gosea, and A. C. Ionita, *Model reduction of bilinear systems in the Loewner framework*, SIAM J. Sci. Comput., 38 (2016), pp. B889–B916, https://doi.org/10.1137/15M1041432. 110, 133

[13] M. M. A. Asif, M. I. Ahmad, P. Benner, L. Feng, and T. Stykel, *Implicit higher-order moment matching technique for model reduction of quadratic-bilinear systems*, Journal of the Franklin Institute, 358 (2021), pp. 2015–2038, https://doi.org/10.1016/j.jfranklin.2020.11.012. 183

[14] Z. Bai, *Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems*, Appl. Numer. Math, 43 (2002), pp. 9–44, https://doi.org/10.1016/S0168-9274(02)00116-2. 41, 43

[15] Z. Bai and D. Skoogh, *A projection method for model reduction of bilinear dynamical systems*, Linear Algebra Appl., 415 (2006), pp. 406–425, https://doi.org/10.1016/j.laa.2005.04.032. 110, 141

[16] Z. Bai and Y. Su, *SOAR: A second-order Arnoldi method for the solution of the quadratic eigenvalue problem*, SIAM J. Matrix Anal. Appl., 26 (2005), pp. 640–659, https://doi.org/10.1137/S0895479803438523. 43

[17] J. Baker, M. Embree, and J. Sabino, *Fast singular value decay for Lyapunov solutions with nonnormal coefficients*, SIAM J. Matrix Anal. Appl., 36 (2015), pp. 656–668, https://doi.org/10.1137/140993867. 50

[18] G. A. Baker Jr., *Essentials of Padé Approximants*, Academic Press, New York, 1975. 41

[19] J. A. Ball, I. Gohberg, and L. Rodman, *Interpolation of Rational Matrix Functions*, vol. 45 of Operator Theory: Advances and Applications, Birkhäuser, Basel, 1990, https://doi.org/10.1007/978-3-0348-7709-1. 43

[20] M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera, *An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations*, C. R. Math. Acad. Sci. Paris, 339 (2004), pp. 667–672, https://doi.org/10.1016/j.crma.2004.08.006. 4, 182

[21] U. Baur, C. A. Beattie, P. Benner, and S. Gugercin, *Interpolatory projection methods for parameterized model reduction*, SIAM J. Sci. Comput., 33 (2011), pp. 2489–2518, https://doi.org/10.1137/090776925. 42, 141, 146

[22] C. A. Beattie and P. Benner, *$\mathcal{H}_2$-optimality conditions for structured dynamical systems*, Preprint MPIMD/14-18, Max Planck Institute for Dynamics of Complex Technical System, Magdeburg, Germany, 2014, https://csc.mpi-magdeburg.mpg.de/preprints/2014/18/. 44, 47, 59, 246, 269

[23] C. A. Beattie and S. Gugercin, *Krylov-based model reduction of second-order systems with proportional damping*, in Proceedings of the 44th IEEE Conference on Decision and Control, 2005, pp. 2278–2283, https://doi.org/10.1109/CDC.2005.1582501. 43

[24] C. A. Beattie and S. Gugercin, *Interpolatory projection methods for structure-preserving model reduction*, Systems Control Lett., 58 (2009), pp. 225–232, https://doi.org/10.1016/j.sysconle.2008.10.016. 16, 43, 44, 46, 47, 111, 156, 178, 244

[25] C. A. Beattie and S. Gugercin, *Realization-independent $\mathcal{H}_2$-approximation*, in 51st IEEE Conference on Decision and Control (CDC), 2012, pp. 4953–4958, https://doi.org/10.1109/CDC.2012.6426344. 47

[26] R. S. Beddig, P. Benner, I. Dorschky, T. Reis, P. Schwerdtner, M. Voigt, and S. W. R. Werner, *Model reduction for second-order dynamical systems revisited*, Proc. Appl. Math. Mech., 19 (2019), p. e201900224, https://doi.org/10.1002/pamm.201900224. iii, 47, 69, 80, 93, 269

[27] R. S. Beddig, P. Benner, I. Dorschky, T. Reis, P. Schwerdtner, M. Voigt, and S. W. R. Werner, *Structure-preserving model reduction for dissipative mechanical systems*, e-print 2010.06331, arXiv, 2020, https://arxiv.org/abs/2010.06331. math.OC. iii, 47, 58, 69, 80, 93, 269

[28] P. Benner and T. Breiten, *On H_2-model reduction of linear parameter-varying systems*, Proc. Appl. Math. Mech., 11 (2011), pp. 805–806, https://doi.org/10.1002/pamm.201110391. 110

[29] P. Benner and T. Breiten, *Interpolation-based $\mathcal{H}_2$-model reduction of bilinear control systems*, SIAM J. Matrix Anal. Appl., 33 (2012), pp. 859–885, https://doi.org/10.1137/110836742. 110, 246

[30] P. Benner and T. Breiten, *Two-sided projection methods for nonlinear model order reduction*, SIAM J. Sci. Comput., 37 (2015), pp. B239–B260, https://doi.org/10.1137/14097255X. 14, 26, 183, 197, 202, 203, 204, 208

[31] P. Benner, T. Breiten, and T. Damm, *Generalized tangential interpolation for model reduction of discrete-time MIMO bilinear systems*, Internat. J. Control, 84 (2011), pp. 1398–1407, https://doi.org/10.1080/00207179.2011.601761. 137, 154, 158, 166, 179, 245

[32] P. Benner, X. Cao, and W. Schilders, *A bilinear $\mathcal{H}_2$ model order reduction approach to linear parameter-varying systems*, Adv. Comput. Math., 45 (2019), pp. 2241–2271, https://doi.org/10.1007/s10444-019-09695-9. 110

[33] P. Benner, J. M. Claver, and E. S. Quintana-Ortí, *Efficient solution of coupled Lyapunov equations via matrix sign function iteration*, in Proc. 3rd Portuguese Conf. on Automatic Control CONTROLO'98, Coimbra, 1998, pp. 205–210. 92

[34] P. Benner, A. Cohen, M. Ohlberger, and K. Willcox, *Model Reduction and Approximation: Theory and Algorithms*, Computational Science & Engineering, SIAM Publications, Philadelphia, PA, 2017, https://doi.org/10.1137/1.9781611974829. 15, 46

[35] P. Benner and T. Damm, *Lyapunov equations, energy functionals, and model order reduction of bilinear and stochastic systems*, SIAM J. Control Optim., 49 (2011), pp. 686–711, https://doi.org/10.1137/09075041X. 110

[36] P. Benner and P. Goyal, *Algebraic Gramians for quadratic-bilinear systems and their application in model order reduction*, in Proc. of the 22nd International Symposium on Mathematical Theory of Networks and Systems - MTNS 2016 , Minnesota, USA, 2016, pp. 81–83, http://hdl.handle.net/11299/181518. 183

[37] P. Benner and P. Goyal, *Multipoint interpolation of Volterra series and $\mathcal{H}_2$-model reduction for a family of bilinear descriptor systems*, Systems Control Lett., 97 (2016), pp. 1–11, https://doi.org/10.1016/j.sysconle.2016.08.008. 110

[38] P. Benner and P. Goyal, *Balanced truncation model order reduction for quadratic-bilinear systems*, e-prints 1705.00160, arXiv, 2017, https://arxiv.org/abs/1705.00160. math.OC. 183

[39] P. Benner and P. Goyal, *Interpolation-based model order reduction for polynomial systems*, SIAM J. Sci. Comput., 43 (2021), pp. A84–A108, https://doi.org/10.1137/19M1259171. 29

[40] P. Benner, P. Goyal, and S. Gugercin, *$\mathcal{H}_2$-quasi-optimal model order reduction for quadratic-bilinear control systems*, SIAM J. Matrix Anal. Appl., 39 (2018), pp. 983–1032, https://doi.org/10.1137/16M1098280. 183, 246

[41] P. Benner, P. Goyal, and I. Pontes Duff, *Identification of dominant subspaces for linear structured parametric systems and model reduction*, e-prints 1910.13945, arXiv, 2019, https://arxiv.org/abs/1910.13945. math.NA. 48

[42] P. Benner, S. Gugercin, and S. W. R. Werner, *Structure-preserving interpolation for model reduction of parametric bilinear systems*, Automatica J. IFAC, 132 (2021), p. 109799, https://doi.org/10.1016/j.automatica.2021.109799. iv, 111, 141, 270

[43] P. Benner, S. Gugercin, and S. W. R. Werner, *Structure-preserving interpolation of bilinear control systems*, Adv. Comput. Math., 47 (2021), p. 43, https://doi.org/10.1007/s10444-021-09863-w. iv, 111, 128, 129, 270

[44] P. Benner, M. Köhler, and J. Saak, *Matrix Equations, Sparse Solvers: M-M.E.S.S.-2.0.1 – Philosophy, features and application for (parametric) model order reduction*, e-print 2003.02088, arXiv, 2020, https://arxiv.org/abs/2003.02088. cs.MS. 32

[45] P. Benner, P. Kürschner, and J. Saak, *A reformulated low-rank ADI iteration with explicit residual factors*, Proc. Appl. Math. Mech., 13 (2013), pp. 585–586, https://doi.org/10.1002/pamm.201310273. 87

[46] P. Benner, P. Kürschner, and J. Saak, *Self-generating and efficient shift parameters in ADI methods for large Lyapunov and Sylvester equations*, Electron. Trans. Numer. Anal., 43 (2014), pp. 142–162, http://etna.mcs.kent.edu/volumes/2011-2020/vol43/abstract.php?vol=43&pages=142-162. 87

[47] P. Benner, P. Kürschner, and J. Saak, *Frequency-limited balanced truncation with low-rank approximations*, SIAM J. Sci. Comput., 38 (2016), pp. A471–A499, https://doi.org/10.1137/15M1030911. 4, 51, 86, 87, 88, 269

[48] P. Benner, P. Kürschner, Z. Tomljanović, and N. Truhar, *Semi-active damping optimization of vibrational systems using the parametric dominant pole algorithm*, Z. Angew. Math. Mech., 96 (2016), pp. 604–619, https://doi.org/10.1002/zamm.201400158. 3, 40, 61, 62, 63

[49] P. Benner, J.-R. Li, and T. Penzl, *Numerical solution of large-scale Lyapunov equations, Riccati equations, and linear-quadratic optimal control problems*, Numer. Lin. Alg. Appl., 15 (2008), pp. 755–777, https://doi.org/10.1002/nla.622. 87

[50] P. Benner and E. S. Quintana-Ortí, *Model reduction based on spectral projection methods*, in Dimension Reduction of Large-Scale Systems, P. Benner, V. Mehrmann, and D. C. Sorensen, eds., vol. 45 of Lect. Notes Comput. Sci. Eng., Berlin/Heidelberg, Germany, 2005, Springer-Verlag, pp. 5–45, https://doi.org/10.1007/3-540-27909-1_1. 38, 65

[51] P. Benner and J. Saak, *Numerical solution of large and sparse continuous time algebraic matrix Riccati and Lyapunov equations: a state of the art survey*, GAMM Mitteilungen, 36 (2013), pp. 32–52, https://doi.org/10.1002/gamm.201310003. 50

[52] P. Benner, Z. Tomljanović, and N. Truhar, *Damping optimization for linear vibrating systems using dimension reduction*, in Vibration Problems ICOVP 2011, J. Náprstek, J. Horáček, M. Okrouhlík, B. Marvalová, F. Verhulst, and J. T. Sawicki, eds., vol. 139, Part 5 of Springer Proceedings in Physics, Prag, Czech Republic, 2011, Springer-Verlag, pp. 297–305, https://doi.org/10.1007/978-94-007-2069-5_41. 59

[53] P. Benner, Z. Tomljanović, and N. Truhar, *Optimal damping of selected eigenfrequencies using dimension reduction*, Numer. Lin. Alg. Appl., 20 (2013), pp. 1–17, https://doi.org/10.1002/nla.833. 59

[54] P. Benner and S. W. R. Werner, *Frequenz- und zeitbeschränktes balanciertes Abschneiden für Systeme zweiter Ordnung*, in Tagungsband GMA-FA 1.30 'Modellbildung, Identifikation und Simulation in der Automatisierungstechnik' und GMA-FA 1.40 'Systemtheorie und Regelungstechnik', Workshops in Anif, Salzburg, 23.-27.09.2019, T. Meurer and F. Woittennek, eds., 2019, pp. 460–474. iii, 80

[55] P. Benner and S. W. R. Werner, *MORLAB – Model Order Reduction LABoratory (version 5.0)*, 2019, https://doi.org/10.5281/zenodo.3332716. see also: https://www.mpi-magdeburg.mpg.de/projects/morlab. iv, 32, 92, 103, 269

[56] P. Benner and S. W. R. Werner, *MORLAB – The Model Order Reduction LABoratory*, e-print 2002.12682, arXiv, 2020, https://arxiv.org/abs/2002.12682. cs.MS. 32

[57] P. Benner and S. W. R. Werner, *Frequency- and time-limited balanced truncation for large-scale second-order systems*, Linear Algebra Appl., 623 (2021),

pp. 68–103, https://doi.org/10.1016/j.laa.2020.06.024. Special issue in honor of P. Van Dooren, Edited by F. Dopico, D. Kressner, N. Mastronardi, V. Mehrmann, and R. Vandebril. iii, xiii, 48, 53, 80, 88, 91, 93, 103, 269

[58] P. Benner and S. W. R. Werner, *SOLBT – Limited balanced truncation for large-scale sparse second-order systems (version 3.0)*, 2021, https://doi.org/10.5281/zenodo.4600763. iv, 32, 88, 95, 269

[59] P. Benner and S. W. R. Werner, *SOMDDPA – Second-Order Modally-Damped Dominant Pole Algorithm (version 2.0)*, 2021, https://doi.org/10.5281/zenodo.3997649. iv, 32, 63, 269

[60] B. Besselink, U. Tabak, A. Lutowska, N. van de Wouw, H. Nijmeijer, D. J. Rixen, M. E. Hochstenbach, and W. H. A. Schilders, *A comparison of model reduction techniques from structural dynamics, numerical mathematics and systems and control*, Journal of Sound and Vibration, 332 (2013), pp. 4403–4422, https://doi.org/10.1016/j.jsv.2013.03.025. 3, 38

[61] D. Billger, *The butterfly gyro*, in Dimension Reduction of Large-Scale Systems, P. Benner, V. Mehrmann, and D. C. Sorensen, eds., vol. 45 of Lecture Notes in Computational Science and Engineering, Berlin/Heidelberg, Germany, 2005, Springer-Verlag, pp. 349–352, https://doi.org/10.1007/3-540-27909-1_18. xi, 5

[62] T. Bonin, H. Fassbender, A. Soppa, and M. Zaeh, *A fully adaptive rational global Arnoldi method for the model-order reduction of second-order MIMO systems with proportional damping*, Math. Comput. Simulat., 122 (2016), pp. 1–19, https://doi.org/10.1016/j.matcom.2015.08.017. 44

[63] T. Breiten, *Interpolatory Methods for Model Reduction of Large-Scale Dynamical Systems*, Dissertation, Department of Mathematics, Otto-von-Guericke University, Magdeburg, Germany, 2013, https://doi.org/10.25673/3917. 4, 12, 183, 184

[64] T. Breiten, *Structure-preserving model reduction for integro-differential equations*, SIAM J. Control Optim., 54 (2016), pp. 2992–3015, https://doi.org/10.1137/15M1032296. 55

[65] T. Breiten and T. Damm, *Krylov subspace methods for model order reduction of bilinear control systems*, Systems Control Lett., 59 (2010), pp. 443–450, https://doi.org/10.1016/j.sysconle.2010.06.003. 110, 141

[66] A. Bruns and P. Benner, *Parametric model order reduction of thermal models using the bilinear interpolatory rational Krylov algorithm*, Math. Comput. Model.

Dyn. Syst., 21 (2015), pp. 103–129, https://doi.org/10.1080/13873954.2014.924534. 110

[67] A. Bultheel and M. Van Barel, *Padé techniques for model reduction in linear system theory: a survey*, J. Comput. Appl. Math., 14 (1986), pp. 401–438, https://doi.org/10.1016/0377-0427(86)90076-2. 41

[68] T. Carleman, *Application de la théorie des équations intégrales linéaires aux systèmes d'équations différentielles non linéaires*, Acta Math., 59 (1932), pp. 63–87, https://doi.org/10.1007/BF02546499. 110

[69] Y. Chahlaoui, D. Lemonnier, A. Vandendorpe, and P. Van Dooren, *Second-order balanced truncation*, Linear Algebra Appl., 415 (2006), pp. 373–384, https://doi.org/10.1016/j.laa.2004.03.032. 4, 52, 53, 54

[70] S. Chaturantabut, C. Beattie, and S. Gugercin, *Structure-preserving model reduction for nonlinear port-Hamiltonian systems*, SIAM J. Sci. Comput., 38 (2016), pp. B837–B865, https://doi.org/10.1137/15M1055085. 7

[71] S. Chaturantabut and D. C. Sorensen, *Nonlinear model reduction via discrete empirical interpolation*, SIAM J. Sci. Comput., 32 (2010), pp. 2737–2764, https://doi.org/10.1137/090766498. 4, 182

[72] M. Condon and R. Ivanov, *Krylov subspaces from bilinear representations of nonlinear systems*, Compel-Int. J. Comp. Math. Electr. Electron. Eng., 26 (2007), pp. 399–406, https://doi.org/10.1108/03321640710727755. 110, 141

[73] R. R. Craig and M. C. C. Bampton, *Coupling of substructures for dynamic analysis*, AIAA J., 6 (1968), pp. 1313–1319, https://doi.org/10.2514/3.4741. 3, 38

[74] F. E. Curtis, T. Mitchell, and M. L. Overton, *A BFGS-SQP method for nonsmooth, nonconvex, constrained optimization and its evaluation using relative minimization profiles*, Optim. Methods Softw., 32 (2017), pp. 148–181, https://doi.org/10.1080/10556788.2016.1208749. 33

[75] E. J. Davison, *A method for simplifying linear dynamic systems*, IEEE Trans. Autom. Control, 11 (1966), pp. 93–101, https://doi.org/10.1109/TAC.1966.1098264. 3, 37

[76] I. Dorschky, T. Reis, and M. Voigt, *Balanced truncation model reduction for symmetric second order systems – A passivity-based approach*, e-print 2006.09170, arXiv, 2020, https://arxiv.org/abs/2006.09170. math.NA. 55

[77] Z. Drmač and S. Gugercin, *A new selection operator for the discrete empirical interpolation method—improved a priori error bound and extensions*, SIAM J. Sci. Comput., 38 (2016), pp. A631–A648, https://doi.org/10.1137/15M1019271. 4, 182

[78] V. Druskin and V. Simoncini, *Adaptive rational Krylov subspaces for large-scale dynamical systems*, Systems Control Lett., 60 (2011), pp. 546–560, https://doi.org/10.1016/j.sysconle.2011.04.013. 88

[79] J. Fehr and P. Eberhard, *Error-controlled model reduction in flexible multibody dynamics*, J. Comput. Nonlinear Dynam., 5 (2010), p. 031005, https://doi.org/10.1115/1.4001372. 91

[80] 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 (2017), pp. 2127–2158, https://doi.org/10.1051/m2an/2017014. 42

[81] L. Feng and P. Benner, *A note on projection techniques for model order reduction of bilinear systems*, in AIP Conference Proceedings, vol. 936, 2007, pp. 208–211, https://doi.org/10.1063/1.2790110. 110, 141

[82] L. Feng and P. Benner, *A new error estimator for reduced-order modeling of linear parametric systems*, IEEE Transactions on Microwave Theory and Techniques, 67 (2019), pp. 4848–4859, https://doi.org/10.1109/TMTT.2019.2948858. 42, 47, 69, 246

[83] F. Fish, *Advantages of natural propulsive systems*, Marine Technology Society Journal, 47 (2013), pp. 37–44, https://doi.org/10.4031/MTSJ.47.5.2. 6

[84] G. M. Flagg, C. A. Beattie, and S. Gugercin, *Interpolatory $\mathcal{H}_\infty$ model reduction*, Systems Control Lett., 62 (2013), pp. 567–574, https://doi.org/10.1016/j.sysconle.2013.03.006. 16

[85] G. M. Flagg and S. Gugercin, *Multipoint Volterra series interpolation and $\mathcal{H}_2$ optimal model reduction of bilinear systems*, SIAM J. Matrix Anal. Appl., 36 (2015), pp. 549–579, https://doi.org/10.1137/130947830. 110, 246

[86] F. Freitas, J. Rommes, and N. Martins, *Gramian-based reduction method applied to large sparse power system descriptor models*, IEEE Trans. Power Syst., 23 (2008), pp. 1258–1270, https://doi.org/10.1109/TPWRS.2008.926693. 89, 107

[87] R. W. Freund, *Model reduction methods based on Krylov subspaces*, Acta Numer., 12 (2003), pp. 267–319, https://doi.org/10.1017/S0962492902000120. 41, 43

[88] R. W. Freund, *Padé-type model reduction of second-order and higher-order linear dynamical systems*, in Dimension Reduction of Large-Scale Systems, P. Benner, V. Mehrmann, and D. C. Sorensen, eds., vol. 45 of Lect. Notes Comput. Sci. Eng., Berlin/Heidelberg, Germany, 2005, Springer-Verlag, pp. 173–189, https://doi.org/10.1007/3-540-27909-1_8. 43, 235

[89] K. Gallivan, A. Vandendorpe, and P. Van Dooren, *Model reduction of MIMO systems via tangential interpolation*, SIAM J. Matrix Anal. Appl., 26 (2004), pp. 328–349, https://doi.org/10.1137/S0895479803423925. 43, 136

[90] W. Gawronski and J.-N. Juang, *Model reduction in limited time and frequency intervals*, Int. J. Syst. Sci., 21 (1990), pp. 349–376, https://doi.org/10.1080/00207729008910366. 4, 50, 51, 81, 84

[91] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, Baltimore, fourth ed., 2013. 12

[92] I. V. Gosea and A. C. Antoulas, *Data-driven model order reduction of quadratic-bilinear systems*, Numer. Lin. Alg. Appl., 25 (2018), p. e2200, https://doi.org/10.1002/nla.2200. 14, 29, 183, 227, 234

[93] I. V. Gosea, I. Pontes Duff, P. Benner, and A. C. Antoulas, *Model order reduction of bilinear time-delay systems*, in 18th European Control Conference (ECC), 2019, pp. 2289–2294, https://doi.org/10.23919/ECC.2019.8796085. 110, 115, 132, 133

[94] P. Goyal and M. Redmann, *Time-limited $\mathcal{H}_2$-optimal model order reduction*, Appl. Math. Comput., 355 (2019), pp. 184–197, https://doi.org/10.1016/j.amc.2019.02.065. 246

[95] P. K. Goyal, *System-theoretic model order reduction for bilinear and quadratic-bilinear systems*, Dissertation, Department of Mathematics, Otto von Guericke University, Magdeburg, Germany, 2018, https://doi.org/10.25673/5319. 4, 12, 183, 184

[96] W. B. Gragg and A. Lindquist, *On the partial realization problem*, Linear Algebra Appl., 50 (1983), pp. 277–319, https://doi.org/10.1016/0024-3795(83)90059-9. 41

[97] L. Grasedyck, D. Kressner, and C. Tobler, *A literature survey of low-rank tensor approximation techniques*, GAMM-Mitt., 36 (2013), pp. 53–78, https://doi.org/10.1002/gamm.201310004. 12

[98] L. Grasedyk, *Hierarchical singular value decomposition of tensors*, SIAM J. Matrix Anal. Appl., 31 (2010), pp. 2029–2054, https://doi.org/10.1137/090764189. 12

[99] M. Green and D. J. N. Limebeer, *Linear Robust Control*, Prentice-Hall, Englewood Cliffs, NJ, 1995. 65

[100] E. J. Grimme, *Krylov projection methods for model reduction*, Ph.D. Thesis, University of Illinois, Urbana-Champaign, USA, 1997, https://perso.uclouvain.be/paul.vandooren/ThesisGrimme.pdf. 41, 42

[101] C. Gu, *QLMOR: A projection-based nonlinear model order reduction approach using quadratic-linear representation of nonlinear systems*, IEEE Trans. Comput. Aided Des. Integr. Circuits. Syst., 30 (2011), pp. 1307–1320, https://doi.org/10.1109/TCAD.2011.2142184. 25, 26, 183, 184, 270

[102] S. Gugercin and A. C. Antoulas, *A survey of model reduction by balanced truncation and some new results*, Internat. J. Control, 77 (2004), pp. 748–766, https://doi.org/10.1080/00207170410001713448. 85, 246

[103] S. Gugercin, A. C. Antoulas, and C. Beattie, *$\mathcal{H}_2$ model reduction for large-scale linear dynamical systems*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 609–638, https://doi.org/10.1137/060666123. 42, 246

[104] S. Gugercin, T. Stykel, and S. Wyatt, *Model reduction of descriptor systems by interpolatory projection methods*, SIAM J. Sci. Comput., 35 (2013), pp. B1010–B1033, https://doi.org/10.1137/130906635. 42

[105] R. J. Guyan, *Reduction of stiffness and mass matrices*, AIAA J., 3 (1965), p. 380, https://doi.org/10.2514/3.2874. 3, 38

[106] K. Haider, A. Ghafoor, M. Imran, and F. M. Malik, *Model reduction of large scale descriptor systems using time limited Gramians*, Asian J. Control, 19 (2017), pp. 1217–1227, https://doi.org/10.1002/asjc.1444. 52, 84

[107] K. Haider, A. Ghafoor, M. Imran, and F. M. Malik, *Frequency interval Gramians based structure preserving model reduction for second-order systems*, Asian J. Control, 20 (2018), pp. 790–801, https://doi.org/10.1002/asjc.1598. 4, 80, 81, 82, 84, 86

[108] K. Haider, A. Ghafoor, M. Imran, and F. M. Malik, *Time-limited Gramian-based model order reduction for second-order form systems*, Trans. Inst. Meas. Control, 41 (2019), pp. 2310–2318, https://doi.org/10.1177/0142331218798893. 4, 80, 82, 84

[109] N. J. Higham, *Functions of Matrices: Theory and Computation*, Applied Mathematics, SIAM Publications, Philadelphia, 2008, https://doi.org/10.1137/1.9780898717778. 87, 88

[110] C. Himpe, *emgr—The Empirical Gramian Framework*, Algorithms, 11 (2018), p. 91, https://doi.org/10.3390/a11070091. 87, 182

[111] C. Himpe, *Comparing (empirical-Gramian-based) model order reduction algorithms*, e-print 2002.12226, arXiv, 2020, https://arxiv.org/abs/2002.12226. math.OC. 33, 34

[112] C. Himpe and M. Ohlberger, *A unified software framework for empirical Gramians*, J. Math., 2013 (2013), pp. 1–6, https://doi.org/10.1155/2013/365909. 4, 182

[113] D. Hinrichsen and A. J. Pritchard, *Mathematical Systems Theory I: Modelling, State Space Analysis, Stability and Robustness*, vol. 48 of Texts in Applied Mathematics, Springer-Verlag, Berlin/Heidelberg, Germany, 2005, https://doi.org/10.1007/b137541. 15

[114] M. Hinze and S. Volkwein, *Proper orthogonal decomposition surrogate models for nonlinear dynamical systems: Error estimates and suboptimal control*, in Dimension Reduction of Large-Scale Systems, P. Benner, V. Mehrmann, and D. C. Sorensen, eds., vol. 45 of Lect. Notes Comput. Sci. Eng., Berlin/Heidelberg, Germany, 2005, Springer-Verlag, pp. 261–306, https://doi.org/10.1007/3-540-27909-1_10. 4, 182

[115] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, New York, NY, USA, second ed., 2012. 12

[116] C. S. Hsu, U. B. Desai, and C. A. Crawley, *Realization algorithms and approximation methods of bilinear systems*, in The 22nd IEEE Conference on Decision and Control, San Antonio, TX, USA, 1983, pp. 783–788, https://doi.org/10.1109/CDC.1983.269628. 110

[117] M. Imran and A. Ghafoor, *Model reduction of descriptor systems using frequency limited Gramians*, J. Franklin Inst., 352 (2015), pp. 33–51, https://doi.org/10.1016/j.jfranklin.2014.10.013. 51, 84, 86

[118] I. M. Jaimoukha and E. M. Kasenally, *Krylov subspace methods for solving large Lyapunov equations*, SIAM J. Numer. Anal., 31 (1994), pp. 227–251, https://doi.org/10.1137/0731012. 88

[119] E. Kelasidi, P. Liljebäck, K. Y. Pettersen, and J. Gravdahl, *Innovation in underwater robots: Biologically inspired swimming snake robots*, IEEE Robotics & Automation Magazine, 23 (2016), pp. 44–62, https://doi.org/10.1109/MRA.2015.2506121. 6

[120] A. Y. Khapalov, *Controllability of the semilinear parabolic equation governed by a multiplicative control in the reaction term: a qualitative approach*, in 42nd IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475), vol. 2, 2003, pp. 1487–1491, https://doi.org/10.1109/CDC.2003.1272822. 110

[121] M. A. Khattak, M. I. Ahmad, L. Feng, and B. Benner, *Multivariate moment matching for model order reduction of quadratic-bilinear systems using error bounds*, e-prints 2105.12966, arXiv, 2021, https://arxiv.org/abs/2105.12966. math.OC. 246

[122] T. G. Kolda, *Multilinear operators for higher-order decompositions*, Technical report SAND2006-2081, Sandia National Laboratories, New Mexico/California, USA, 2006, https://doi.org/10.2172/923081. 12

[123] T. G. Kolda and B. W. Bader, *Tensor decompositions and applications*, SIAM Rev., 51 (2009), pp. 455–500, https://doi.org/10.1137/07070111X. 12, 13, 14

[124] S. G. Korpeoglu and I. Kucuk, *Optimal control of a bilinear system with a quadratic cost functional*, in 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), 2018, pp. 1–6, https://doi.org/10.1109/ICCUBEA.2018.8697554. 110

[125] P. Koutsovasilis and M. Beitelschmidt, *Comparison of model reduction techniques for large mechanical systems*, Multibody System Dynamics, 20 (2008), pp. 111–128, https://doi.org/10.1007/s11044-008-9116-4. 3, 38

[126] K. Kowalski and W.-H. Steeb, *Nonlinear Dynamical Systems and Carleman Linearization*, World Scientific, Singapore, 1991, https://doi.org/10.1142/1347. 110

[127] D. Kressner and C. Tobler, *Krylov subspace methods for linear systems with tensor product structure*, SIAM J. Matrix Anal. Appl., 31 (2010), pp. 1688–1714, https://doi.org/10.1137/090756843. 12

[128] K. Kunisch and S. Volkwein, *Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics*, SIAM J. Numer. Anal., 40 (2002), pp. 492–515, https://doi.org/10.1137/S0036142900382612. 4, 182

[129] P. Kürschner, *Efficient Low-Rank Solution of Large-Scale Matrix Equations*, Dissertation, Department of Mathematics, Otto von Guericke University, Magdeburg, Germany, 2016, http://hdl.handle.net/11858/00-001M-0000-0029-CE18-2. 87

[130] P. Kürschner, *Balanced truncation model order reduction in limited time intervals for large systems*, Advances in Computational Mathematics, 44 (2018), pp. 1821–1844, https://doi.org/10.1007/s10444-018-9608-6. 4, 52, 84, 269

[131] S. Lall, J. E. Marsden, and S. Glavaški, *Empirical model reduction of controlled nonlinear systems*, IFAC Proceedings Volumes (14th IFAC World Congress), 32 (1999), pp. 2598–2603, https://doi.org/10.1016/S1474-6670(17)56442-3. 87, 182

[132] N. Lang, H. Mena, and J. Saak, *On the benefits of the LDL^T factorization for large-scale differential matrix equation solvers*, Linear Algebra Appl., 480 (2015), pp. 44–71, https://doi.org/10.1016/j.laa.2015.04.006. 87

[133] D. Lawrence, J. H. Myatt, and R. C. Camphouse, *On model reduction via empirical balanced truncation*, in Proc. Am. Control Conf., vol. 5, 2005, pp. 3139–3144, https://doi.org/10.1109/ACC.2005.1470454. 4, 182

[134] M. Lehner, *Modellreduktion in elastischen Mehrkörpersystemen*, Dissertation, Institut für Technische und Numerische Mechanik, Universität Stuttgart, Germany, 2007, https://www.shaker.de/de/content/catalogue/index.asp?lang=de&ID=8&ISBN=978-3-8322-6783-4. 39

[135] M. Lehner and P. Eberhard, *A two-step approach for model reduction in flexible multibody dynamics*, Multibody Syst. Dyn., 17 (2007), pp. 157–176, https://doi.org/10.1007/s11044-007-9039-5. 91

[136] J.-R. Li and J. White, *Low rank solution of Lyapunov equations*, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 260–280, https://doi.org/10.1137/S0895479801384937. 87

[137] A. D. Marchese, C. D. Onal, and D. Rus, *Autonomous soft robotic fish capable of escape maneuvers using fluidic elastomer actuators*, Soft Robotics, 1 (2014), pp. 75–87, https://doi.org/10.1089/soro.2013.0009. 6

[138] 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 (1996), pp. 162–170, https://doi.org/10.1109/59.486093. 3, 39, 40, 60

[139] The MathWorks, Inc., *MATLAB*, http://www.matlab.com. 32

[140] A. J. Mayo and A. C. Antoulas, *A framework for the solution of the generalized realization problem*, Linear Algebra Appl., 425 (2007), pp. 634–662, https://doi.org/10.1016/j.laa.2007.03.008. Special Issue in honor of P. A. Fuhrmann, Edited by A. C. Antoulas, U. Helmke, J. Rosenthal, V. Vinnikov, and E. Zerz. 47

[141] G. P. McCormick, *Computability of global solutions to factorable nonconvex programs: Part I — Convex underestimating problems*, Mathematical Programming, 10 (1976), pp. 147–175, https://doi.org/10.1007/BF01580665. 184

[142] V. Mehrmann and T. Stykel, *Balanced truncation model reduction for large-scale systems in descriptor form*, in Dimension Reduction of Large-Scale Systems, P. Benner, V. Mehrmann, and D. C. Sorensen, eds., vol. 45 of Lect. Notes Comput. Sci. Eng., Berlin/Heidelberg, Germany, 2005, Springer-Verlag, pp. 83–115, https://doi.org/10.1007/3-540-27909-1_3. 95, 129

[143] D. G. Meyer and S. Srinivasan, *Balancing and model reduction for second-order form linear systems*, IEEE Trans. Autom. Control, 41 (1996), pp. 1632–1644, https://doi.org/10.1109/9.544000. 4, 52, 53

[144] P. Mlinarić, *Structure-preserving model order reduction for network systems*, Dissertation, Department of Mathematics, Otto von Guericke University, Magdeburg, Germany, 2020, https://doi.org/10.25673/33570. 44, 47, 246

[145] R. R. Mohler, *Natural bilinear control processes*, IEEE Transactions on Systems Science and Cybernetics, 6 (1970), pp. 192–197, https://doi.org/10.1109/TSSC.1970.300341. 4, 23, 110

[146] R. R. Mohler, *Bilinear Control Processes: With Applications to Engineering, Ecology and Medicine*, vol. 106 of Mathematics in Science and Engineering, Academic Press, New York, London, 1973. 4, 110

[147] B. C. Moore, *Principal component analysis in linear systems: controllability, observability, and model reduction*, IEEE Trans. Autom. Control, AC–26 (1981), pp. 17–32, https://doi.org/10.1109/TAC.1981.1102568. 48, 49

[148] P. C. Müller and W. Schiehlen, *Linear vibrations*, vol. 7 of Mechanics: Dynamical Systems, Springer-Verlag, 1985, https://doi.org/10.1007/978-94-009-5047-4. 58

[149] Oberwolfach Benchmark Collection, *Butterfly gyroscope*. hosted at MORwiki – Model Order Reduction Wiki, 2004, http://modelreduction.org/index.php/Butterfly_Gyroscope. xi, 5

[150] Y. Ou, *Optimal Control of a Class of Nonlinear Parabolic PDE Systems Arising in Fusion Plasma Current Profile Dynamics*, PhD thesis, Lehigh University, Bethlehem, Pennsylvania, USA, 2010. 110

[151] H. Panzer, T. Wolf, and B. Lohmann, *A strictly dissipative state space representation of second order systems*, at-Automatisierungstechnik, 60 (2012), pp. 392–397, https://doi.org/10.1524/auto.2012.1015. 19, 20, 21, 88

[152] H. K. F. Panzer, *Model Order Reduction by Krylov Subspace Methods with Global Error Bounds and Automatic Choice of Parameters*, Dissertation, Technische Universität München, Munich, Germany, 2014, https://mediatum.ub.tum.de/doc/1207822/1207822.pdf. 43

[153] D. Petersson and J. Löfberg, *Model reduction using a frequency-limited $\mathcal{H}_2$-cost*, Systems Control Lett., 67 (2014), pp. 32–39, https://doi.org/10.1016/j.sysconle.2014.02.004. 246

[154] J. W. Polderman and J. C. Willems, *Introduction to Mathematical Systems Theory: A Behavioral Approach*, vol. 26 of Texts in Applied Mathematics, Springer-Verlag, New York, NY, USA, 1998, https://doi.org/10.1007/978-1-4757-2953-5. 15

[155] I. Pontes Duff, S. Gugercin, C. Beattie, C. Poussot-Vassal, and C. Seren, *$\mathcal{H}_2$-optimality conditions for reduced time-delay systems of dimensions one*, IFAC-PapersOnLine, 49 (2016), pp. 7–12, https://doi.org/10.1016/j.ifacol.2016.07.464. 13th IFAC on Time Delay Systems TDS 2019. 47, 246

[156] I. Pontes Duff, C. Poussot-Vassal, and C. Seren, *Realization independent single time-delay dynamical model interpolation and $\mathcal{H}_2$-optimal approximation*, in 54th IEEE Conference on Decision and Control (CDC), 2015, pp. 4662–4667, https://doi.org/10.1109/CDC.2015.7402946. 47

[157] I. Pontes Duff, C. Poussot-Vassal, and C. Seren, *$\mathcal{H}_2$-optimal model approximation by input/output-delay structured reduced-order models*, Systems Control Lett., 117 (2018), pp. 60–67, https://doi.org/10.1016/j.sysconle.2018.05.003. 47, 246

[158] K. Qian and Y. Zhang, *Bilinear model predictive control of plasma keyhole pipe welding process*, J. Manuf. Sci. Eng., 136 (2014), p. 031002, https://doi.org/10.1115/1.4025337. 110

[159] T. Reis and T. Stykel, *Balanced truncation model reduction of second-order systems*, Math. Comput. Model. Dyn. Syst., 14 (2008), pp. 391–406, https://doi.org/10.1080/13873950701844170. 4, 19, 21, 52, 53, 54, 80, 81, 84

[160] A. C. Rodriguez, S. Gugercin, and J. Boggaard, *Interpolatory model reduction of parameterized bilinear dynamical systems*, Adv. Comput. Math., 44 (2018), pp. 1887–1916, https://doi.org/10.1007/s10444-018-9611-y. 137, 141, 146, 154, 158, 166, 179, 245

[161] J. Rommes, *Methods for eigenvalue problems with applications in model order reduction*, Dissertation, Utrecht University, Netherlands, 2007, https://dspace.library.uu.nl/handle/1874/21787. 3, 39, 40

[162] J. Rommes and N. Martins, *Efficient computation of multivariable transfer function dominant poles using subspace acceleration*, IEEE Trans. Power Syst., 21 (2006), pp. 1471–1483, https://doi.org/10.1109/TPWRS.2006.881154. 3, 40, 62

[163] J. Rommes and N. Martins, *Computing transfer function dominant poles of large-scale second-order dynamical systems*, SIAM J. Sci. Comput., 30 (2008), pp. 2137–2157, https://doi.org/10.1137/070684562. 3, 40, 62, 63

[164] J. Rommes and G. L. G. Sleijpen, *Convergence of the dominant pole algorithm and Rayleigh quotient iteration*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 346–363, https://doi.org/10.1137/060671401. 61

[165] C. W. Rowley, *Model reduction for fluids, using balanced proper orthogonal decomposition*, Int. J. Bifurcat. Chaos, 15 (2005), pp. 997–1013, https://doi.org/10.1142/S0218127405012429. 4, 182

[166] W. J. Rugh, *Nonlinear System Theory: The Volterra/Wiener Approach*, The Johns Hopkins University Press, Baltimore, 1981. 23, 24, 25, 26

[167] J. Saak, M. Köhler, and P. Benner, *M-M.E.S.S.-2.0.1 – The Matrix Equations Sparse Solvers library*, 2020, https://doi.org/10.5281/zenodo.3606345. see also: https://www.mpi-magdeburg.mpg.de/projects/mess. 32, 88, 95

[168] J. Saak, D. Siebelts, and S. W. R. Werner, *A comparison of second-order model order reduction methods for an artificial fishtail*, at-Automatisierungstechnik, 67 (2019), pp. 648–667, https://doi.org/10.1515/auto-2019-0027. iii, xi, 6, 58, 60, 61, 63, 75, 80, 93, 269

[169] B. Salimbahrami, *Structure Preserving Order Reduction of Large Scale Second Order Models*, Dissertation, Technische Universität München, Munich, Germany, 2005, https://mediatum.ub.tum.de/doc/601950/00000941.pdf. 43

[170] B. Salimbahrami and B. Lohmann, *Order reduction of large scale second-order systems using Krylov subspace methods*, Linear Algebra Appl., 415 (2006), pp. 385–405, https://doi.org/10.1016/j.laa.2004.12.013. 43

[171] J. Saputra, R. Saragih, and D. Handayani, *Robust H_∞ controller for bilinear system to minimize HIV concentration in blood plasma*, J. Phys.: Conf. Ser., 1245 (2019), p. 012055, https://doi.org/10.1088/1742-6596/1245/1/012055. 110

[172] P. Schwerdtner and M. Voigt, *Computation of the $\mathcal{L}_\infty$-norm using rational interpolation*, IFAC-PapersOnLine, 51 (2018), pp. 84–89, https://doi.org/10.1016/j.ifacol.2018.11.086. 9th IFAC Symposium on Robust Control Design ROCOND 2018, Florianópolis, Brazil. 47, 69

[173] Y. Shamash, *Linear system reduction using Pade approximation to allow retention of dominant modes*, Internat. J. Control, 21 (1975), pp. 257–272, https://doi.org/10.1080/00207177508921985. 41

[174] D. Siebelts, A. Kater, and T. Meurer, *Modeling and motion planning for an artificial fishtail*, IFAC-PapersOnLine, 51 (2018), pp. 319–324, https://doi.org/10.1016/j.ifacol.2018.03.055. 6, 39, 75

[175] D. Siebelts, A. Kater, T. Meurer, and J. Andrej, *Matrices for an artificial fishtail*. hosted at MORwiki – Model Order Reduction Wiki, 2019, https://doi.org/10.5281/zenodo.2558728. 6

[176] V. Simoncini, *A new iterative method for solving large-scale Lyapunov matrix equations*, SIAM J. Sci. Comput., 29 (2007), pp. 1268–1288, https://doi.org/10.1137/06066120X. 88

[177] E. D. Sontag, *Mathematical Control Theory: Deterministic Finite Dimensional Systems*, vol. 6 of Texts in Applied Mathematics, Springer-Verlag, New York, NY, USA, second ed., 1998, https://doi.org/10.1007/978-1-4612-0577-7. 15

[178] D. Spescha, *Framework for Efficient and Accurate Simulation of the Dynamics of Machine Tools*, Dissertation, Faculty of Mathematics/Computer Science and Mechanical Engineering, Clausthal University of Technology, Germany, 2018, https://doi.org/10.21268/20181119-132644. 68

[179] T. Stykel, *Analysis and Numerical Solution of Generalized Lyapunov Equations*, Dissertation, TU Berlin, 2002, http://webdoc.sub.gwdg.de/ebook/e/2003/tu-berlin/stykel_tatjana.pdf. 48

[180] M. Toda, *Vibration of a chain with nonlinear interaction*, Journal of the Physical Society of Japan, 22 (1967), pp. 431–436, https://doi.org/10.1143/JPSJ.22.431. 7

[181] N. Truhar and K. Veselić, *An efficient method for estimating the optimal dampers' viscosity for linear vibrating systems using Lyapunov equation*, SIAM J.

Matrix Anal. Appl., 31 (2009), pp. 18–39, https://doi.org/10.1137/070683052. 59

[182] A. Varga, *Enhanced modal approach for model reduction*, Math. Model. Syst., 1 (1995), pp. 91–105, https://doi.org/10.1080/13873959508837010. 40

[183] K. Veselić, *Damped oscillations of linear systems*, vol. 2023 of Lecture Notes in Math., Springer-Verlag, 2011, https://doi.org/10.1007/978-3-642-21335-9. 58

[184] D. C. Villemagne and R. E. Skelton, *Model reduction using a projection formulation*, Internat. J. Control, 46 (1987), pp. 2141–2169, https://doi.org/10.1080/00207178708934040. 36, 41

[185] P. Vuillemin, A. Maillard, and C. Poussot-Vassal, *Optimal modal truncation*, e-print 2009.11540, arXiv, 2020, http://arxiv.org/abs/2009.11540. math.OC. 40

[186] K. Willcox and J. Peraire, *Balanced model reduction via the proper orthogonal decomposition*, AIAA J., 40 (2002), pp. 2323–2330, https://doi.org/10.2514/2.1570. 4, 182

[187] T. Wolf, H. K. F. Panzer, and B. Lohmann, *Model order reduction by approximate balanced truncation: A unifying framework*, at-Automatisierungstechnik, 61 (2013), pp. 545–556, https://doi.org/10.1524/auto.2013.1007. 50, 91

[188] S. Wyatt, *Issues in Interpolatory Model Reduction: Inexact Solves, Second-order Systems and DAEs*, Ph.D. Thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, USA, 2012, http://hdl.handle.net/10919/27668. 44, 47

[189] Y. Xu and T. Zeng, *Optimal $\mathcal{H}_2$ model reduction for large scale MIMO systems via tangential interpolation*, Int. J. Numer. Anal. Model., 8 (2011), pp. 174–188, http://www.math.ualberta.ca/ijnam/Volume-8-2011/No-1-11/2011-01-10.pdf. 42

[190] L. Zhang and J. Lam, *On H_2 model reduction of bilinear systems*, Automatica J. IFAC, 38 (2002), pp. 205–216, https://doi.org/10.1016/S0005-1098(01)00204-7. 110, 246

THESES

1. This thesis is concerned with the development of new structure-preserving model order reduction methods for mechanical systems. Modal and balanced truncation-based techniques are considered for the linear system case as well as interpolation-based approaches for nonlinear system classes covering an even more general structure in differential equations than only second-order time derivatives as in the mechanical system case.

2. For the special case of modally damped linear mechanical systems, a structured pole-residue form is derived in which the system poles appear pairwise corresponding to the same residue. Based on that, dominant pole pairs for modally damped systems are defined and a new dominant pole algorithm is developed that preserves, in each computation step, the modally damped system structure.

3. Two $\mathcal{H}_\infty$-error bounds are derived for the new structured dominant pole algorithm. One of them implies good approximation results in cases where the dominance measure of the dominant pole pairs decays fast enough. This bound is reformulated to be used in practical computations.

4. A structure-preserving basis enrichment method is suggested that allows for further approximation when the error of the dominant pole algorithm stagnates. This approach preserves computed dominant poles in the reduced-order model.

5. The ideas of second-order balanced truncation methods are combined with the frequency- and time-limited system Gramians to develop structure-preserving limited balanced truncation methods for linear second-order systems. These methods have been shown to provide similar or better local approximation errors in time or frequency ranges of interest compared to the classical (unlimited) second-order balanced truncation methods while preserving the second-order system structure.

6. The preservation of stability in the constructed reduced-order models is a known problem of limited balanced truncation. Two different modifications of the structured limited balanced truncation methods are outlined to potentially preserve stability in the reduced-order models in cases where the fully limited methods fail to do so. The approach that replaces limited by infinite Gramians provides good

results in terms of stability preservation and compatible approximation quality in frequency and time regions of interest compared to the fully limited methods.

7. The formulation of multivariate subsystem transfer functions of bilinear systems is extended to the structured system case. This includes, in particular, the case of mechanical bilinear systems. Another system class covered by this framework are bilinear time-delay systems.

8. A new interpolation framework is developed for the structured subsystem transfer functions of bilinear systems. Thereby, conditions on projection spaces used for model order reduction are imposed to satisfy simple and Hermite interpolation conditions. By two-sided projection, it is possible to match interpolation conditions with higher transfer function levels as well as higher-order partial derivatives without explicitly evaluating the corresponding transfer functions.

9. The interpolation theory is further extended to the case of structured parametric bilinear systems. Conditions on the construction of the projection spaces similar to the non-parametric case are imposed. The sensitivities of the subsystem transfer functions with respect to the parameters can be matched implicitly via two-sided projection.

10. In case of MIMO bilinear systems, the subsystem transfer functions are matrices, for which the input dimension is growing exponentially with the transfer function level. The scalar transfer function interpolation changes to matrix interpolation in this case. Therefore, the dimensions of the projection spaces also grow exponentially. A new tangential interpolation framework is proposed to restrict the subsystem transfer function interpolation to vectors instead of matrices independent of the transfer function level.

11. The three known concepts for the representation of quadratic-bilinear systems in the frequency domain via multivariate transfer functions are extended to the case of structured quadratic-bilinear MIMO systems. As motivation a nonlinear mechanical system is used and rewritten into quadratic-bilinear form.

12. For all structured transfer function concepts, conditions on projection spaces for model order reduction are imposed to satisfy transfer function interpolation conditions. This includes results on the interpolation of arbitrarily high transfer function levels and the implicit interpolation of higher-order partial derivatives. The special case of SISO systems with underlying symmetric tensors representing the quadratic terms from the literature is also treated.

STATEMENT OF SCIENTIFIC COOPERATIONS

This work is based on articles and reports (published and unpublished) that have been obtained in cooperation with various coauthors. To guarantee a fair assessment of this thesis, this statement clarifies the contributions that each individual coauthor has made. The following people contributed to the content of this work:

- Peter Benner (PB), Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg, Germany;
- Serkan Gugercin (SG), Virginia Polytechnic Institute and State University, Blacksburg, USA;
- Jens Saak (JS), Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg, Germany.

Chapter 4

The work in Section 4.1 was self-directed. Initially, I got pointed to [22] by PB for ideas about structured $\mathcal{H}_2$-optimality conditions, which led instead to the idea of re-using the structured pole residue form in [22] for modal truncation. Some parts of the section were proofread first by JS for the publication in [168] and later by PB for [27]. Section 4.2 was initiated by PB with the idea to extend the previous work from [47, 130] to the second-order system case. All theoretical and computational results were obtained by myself and proofread by PB and in parts by JS for the publication in [26,27,57,168]. Also, the results published in [168] were proofread by Dirk Wolfram, formerly Dirk Siebelts, from Kiel University, and the results in [26, 27] were proofread by Matthias Voigt and Paul Schwerdtner from the Technical University Berlin, as well as by Ines Dorschky and Timo Reis from the Hamburg University, and Rebekka S. Beddig from the Hamburg University of Technology. The three MATLAB toolboxes [55, 58, 59], which implement large parts of the methods described in this chapter, and the codes for the numerical experiments were all authored by myself.

Chapter 5

SG initiated the idea to investigate structure-preserving model reduction for mechanical bilinear systems via interpolation. I extended this idea to general structured transfer functions of bilinear systems in Sections 5.2 to 5.4. For Section 5.5, PB suggested considering an extension to parametric structured bilinear systems. All these presented theoretical and computational results were obtained by myself but proofread and improved by PB and SG for the publication in [42, 43]. Section 5.6 was self-initiated. After my initial motivation of extending tangential interpolation to bilinear systems in the sense of Section 5.6.1 and discussions with JS leading to the time domain motivation in Section 5.6.2, I developed on my own the unifying interpolation framework for tangential interpolation of structured bilinear systems presented in Section 5.6.3. Discussions with SG and PB improved the presentation of the theoretical results. All codes for the numerical experiments in this chapter were authored by myself and I obtained the computational results on my own.

Chapter 6

The idea to consider quadratic-bilinear mechanical systems in a similar interpolation framework as developed for the bilinear case was suggested by SG. He pointed to [101] for the concept of symmetric transfer functions. I extended the symmetric, and later the regular and generalized transfer functions, to the general structured quadratic-bilinear system setting and developed the presented interpolation theory by myself. The codes used for the generation of the different numerical examples as well as for the numerical experiments were authored by myself, and I obtained the presented computational results on my own.