Advances in Computational Mathematics (2024) 50:18
https://doi.org/10.1007/s10444-024-10109-8

Structured interpolation for multivariate transfer functions of quadratic-bilinear systems

Peter Benner[1,2] · **Serkan Gugercin**[3] · **Steffen W. R. Werner**[3]

Received: 10 May 2023 / Accepted: 7 February 2024 / Published online: 12 March 2024

Abstract

High-dimensional/high-fidelity nonlinear dynamical systems appear naturally when the goal is to accurately model real-world phenomena. Many physical properties are thereby encoded in the internal differential structure of these resulting large-scale nonlinear systems. The high dimensionality of the dynamics causes computational bottlenecks, especially when these large-scale systems need to be simulated for a variety of situations such as different forcing terms. This motivates model reduction where the goal is to replace the full-order dynamics with accurate reduced-order surrogates. Interpolation-based model reduction has been proven to be an effective tool for the construction of cheap-to-evaluate surrogate models that preserve the internal structure in the case of weak nonlinearities. In this paper, we consider the construction of multivariate interpolants in frequency domain for structured quadratic-bilinear systems. We propose definitions for structured variants of the symmetric subsystem and generalized transfer functions of quadratic-bilinear systems and provide conditions for structure-preserving interpolation by projection. The theoretical results are illustrated using two numerical examples including the simulation of molecular dynamics in crystal structures.

Communicated by: Tobias Breiten

✉ Peter Benner
benner@mpi-magdeburg.mpg.de; peter.benner@ovgu.de

Serkan Gugercin
gugercin@vt.edu

Steffen W. R. Werner
steffen.werner@vt.edu

[1] Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstr. 1, 39106 Magdeburg, Germany

[2] Faculty of Mathematics, Otto von Guericke University Magdeburg, Universitätsplatz 2, 39106 Magdeburg, Germany

[3] Department of Mathematics and Division of Computational Modeling and Data Analytics, Academy of Data Science, Virginia Tech, Blacksburg, VA 24061, USA

Keywords Model order reduction · Quadratic-bilinear systems · Structure-preserving approximation · Multivariate interpolation

Mathematics Subject Classification (2010) 30E05 · 34K17 · 65D05 · 93C10 · 93A15

1 Introduction

The accurate modeling of real-world phenomena and processes yields dynamical systems typically including nonlinearities. Additionally, these systems often inherit internal differential structures such as higher order time derivatives or time delays from the underlying physical nature of the considered problem. A particular example for such internal structures that usually arises in the modeling of mechanical structures is the description of the system states by second-order time derivatives. Such nonlinear mechanical systems take the form

$$\begin{aligned} \widetilde{M}\ddot{q}(t) &= f\big(q(t), \dot{q}(t), u(t)\big), \\ y(t) &= Cpq(t) + Cv\dot{q}(t), \end{aligned} \tag{1}$$

with internal states $q(t) \in \mathbb{R}^{\ell}$, describing the system behavior, the external controls $u(t) \in \mathbb{R}^{m}$ that allow the user to change the internal behavior, and the quantities of interest $y(t) \in \mathbb{R}^{p}$ that can be observed from the outside, e.g., by sensor measurements. Thereby, the first equation in (1) is a second-order differential equation with mass (descriptor) matrix $\widetilde{M} \in \mathbb{R}^{\ell \times \ell}$ and the nonlinear time evolution function $f : \mathbb{R}^{\ell} \times \mathbb{R}^{\ell} \times \mathbb{R}^{m} \to \mathbb{R}^{\ell}$. The second, algebraic equation describes the quantities of interest as linear combination of the states and their first-order derivatives. Throughout this paper, we assume for any system to have homogeneous initial conditions.

In many applications, in particular those involving discretizations of partial differential equations, the number of differential equations ℓ in (1), describing the internal system behavior, is large and increases further with the demand for more accuracy. However, an increasing amount of differential equations also leads to an increasing demand for computational resources such as time and memory for simulations of the models or their use in optimization. A remedy to this problem is model order reduction, which aims for the computation of cheap-to-evaluate surrogate models described by significantly fewer differential equations, $r \ll \ell$, which approximate the input-to-output behavior of the original system as

$$\|y - \hat{y}\| \leq \tau \cdot \|u\|,$$

in some suitable norms for the output of the reduced model $\hat{y}$ and all admissible inputs u.

An established approach for model reduction of general (structured) nonlinear systems such as (1) is proper orthogonal decomposition (POD) [22, 26, 41], in which time simulations are used to extract information about the system dynamics for the construction of a basis matrix to project the system states. Other approaches aim for the extension of balancing-related model reduction to nonlinear systems using Gramians

defined via time simulations as in the empirical Gramian method [21, 27, 28] or by new energy measures [24, 36] to construct suitable projection matrices. Nevertheless, a problem arising in the general system case is the approximation of the nonlinear time evolution function f in (1) that circumvents the computationally expensive lifting and truncation of the low-dimensional state in every time step. A solution to that are hyperreduction techniques such as the (discrete) empirical interpolation method ((D)EIM) [4, 12, 13, 30], which computes a selection operator to restrict the evaluation of f to its most dominant rows. This introduces another layer of approximations and needs explicit access to the implementation of the original time evolution function f.

An alternative to hyperreduction that gained significant popularity in model reduction in the last decade is quadratic-bilinearization [19]; in optimization also known as McCormick relaxation [29]: For f smooth enough, general nonlinear systems can be rewritten into quadratic-bilinear form by introducing auxiliary variables and differential-algebraic equations, which then can be reduced directly using the classical projection-based model reduction approach. In the case of (1), the corresponding quadratic-bilinear system retains the internal mechanical structure as

$$\begin{aligned} 0 &= M\ddot{q}(t) + D\dot{q}(t) + Kq(t) \\ &\quad + H_{\mathrm{vv}}\big(\dot{q}(t) \otimes \dot{q}(t)\big) + H_{\mathrm{vp}}\big(\dot{q}(t) \otimes q(t)\big) \\ &\quad + H_{\mathrm{pv}}\big(q(t) \otimes \dot{q}(t)\big) + H_{\mathrm{pp}}\big(q(t) \otimes q(t)\big) \\ &\quad - \sum_{j=1}^{m} N_{\mathrm{v,j}}\dot{q}(t)u_j(t) - \sum_{j=1}^{m} N_{\mathrm{p,j}}q(t)u_j(t) - B_{\mathrm{u}}u(t), \\ y(t) &= Cpq(t) + Cv\dot{q}(t), \end{aligned} \tag{2}$$

where $M, D, K, N_{\mathrm{v,j}}, N_{\mathrm{p,j}} \in \mathbb{R}^{n\times n}$, for $j = 1, \ldots, m$, $H_{\mathrm{vv}}, H_{\mathrm{vp}}, H_{\mathrm{pv}}, H_{\mathrm{pp}} \in \mathbb{R}^{n\times n^2}$, $B_{\mathrm{u}} \in \mathbb{R}^{n\times m}$, $Cp, Cv \in \mathbb{R}^{p\times n}$, and $\otimes$ denotes the Kronecker product. Due to the introduction of auxiliary variables, we have that $n \geq \ell$, which appears counter-intuitive to the actual task of reducing the number of internal system states in model reduction. However, the new nonlinearity structure of (2) allows to apply well-established model reduction techniques without the necessity of the hyperreduction step for the nonlinearity.

In this paper, we extend the idea of quadratic-bilinear subsystem interpolation to systems with additional internal differential structures such as in the mechanical case (2). We propose extensions for the definitions of the first three symmetric subsystem transfer functions and the first three generalized transfer functions to the structured system case and then present subspace conditions for structure-preserving interpolation of these transfer functions. The effectiveness of the resulting model reduction methods based on this interpolation theory is illustrated on two different structured examples. In this manuscript, we present a refined version of the theoretical results that were derived in the course of writing the dissertation of the corresponding author [39]. In comparison to [39], we reformulated the results in Propositions 1, 2 and 3 to allow practitioners an easy implementation of the proposed theory. Corollary 1 yields a special case of the results of the theory in [39] adapted to the most common choice of interpolation

point selection. We present new results on the implicit interpolation of multivariate transfer functions in Theorem 1. Also, we provide new numerical experiments to test the performance of the proposed approaches.

The rest of the paper is organized as follows: In Section 2, we recapitulate the ideas of Volterra series expansions and unstructured quadratic-bilinear systems in the frequency domain. We present the definitions of structured transfer functions of quadratic-bilinear systems in Section 3 and the results on structure-preserving interpolation in Section 4. We employ the interpolation results for model reduction of two structured numerical examples in Section 5. The paper is concluded in Section 6.

2 Mathematical preliminaries

In this section, we remind the reader of unstructured quadratic-bilinear systems and summarize the concept of Volterra series as well as two resulting transfer function formulations. Lastly, we recapitulate the concept of internal (differential) structures.

2.1 Volterra series expansions for unstructured quadratic-bilinear systems

Unstructured (first-order) quadratic-bilinear systems are dynamical systems of the form

$$\begin{aligned} E\dot{x}(t) &= Ax(t) + H\big(x(t) \otimes x(t)\big) + \sum_{j=1}^{m} N_j x(t) u_j(t) + Bu(t), \\ y(t) &= Cx(t), \end{aligned} \tag{3}$$

with $E, A, N_j \in \mathbb{R}^{n\times n}$, for $j = 1, \ldots, m$, $H \in \mathbb{R}^{n\times n^2}$, $B \in \mathbb{R}^{n\times m}$ and $C \in \mathbb{R}^{p\times n}$. Model reduction methods developed for (3) include the interpolation of multivariate subsystem transfer functions [1, 2, 6, 19], Volterra series interpolation [2, 9], balanced truncation [7], learning models from frequency domain data via the Loewner framework [16], or learning models from time domain data by operator inference [31, 32]. The reformulation of general nonlinear systems into quadratic-bilinear form (3) has also been proven to be an effective strategy for classical nonlinear model reduction methods such as POD [25].

The development of frequency domain model reduction methods for nonlinear systems of the form (3) is based on the Volterra series expansion. This allows to describe the solution of nonlinear dynamical systems as a series of solutions of coupled linear systems [34]. A common approach to derive the Volterra series expansion is by variational analysis [19]. Let a scaled input signal $\alpha u(t)$, with $\alpha > 0$, be given for the quadratic-bilinear system (3) and assume the system state to have an analytic representation of the form

$$x(t) = \sum_{k=1}^{\infty} \alpha^k x_k(t), \tag{4}$$

with a sequence of states $x_k(t)$. Inserting (4) into (3) and extracting the terms corresponding to the same power of the coefficient α yields the states $x_k(t)$ to be described

by cascaded subsystems, which are linear in their respective unknown state $x_k(t)$. For example, the first three resulting linear subsystems for (4) are given by

$$\begin{aligned} E\dot{x}_1(t) &= Ax_1(t) + Bu(t), \\ E\dot{x}_2(t) &= Ax_2(t) + H\big(x_1(t) \otimes x_1(t)\big) + \sum_{j=1}^{m} N_j x_1(t) u_j(t), \\ E\dot{x}_3(t) &= Ax_3(t) + H\big(x_1(t) \otimes x_2(t) + x_2(t) \otimes x_1(t)\big) + \sum_{j=1}^{m} N_j x_2(t) u_j(t). \end{aligned} \tag{5}$$

Applying the variation-of-constants formula to the subsystems in (5) allows the description of the input-to-output behavior of (3) via its Volterra series expansion:

$$y(t) = \sum_{k=1}^{\infty} \int_0^t \int_0^{t_1} \cdots \int_0^{t_{k-1}} g_k(t_1, \ldots, t_k)\big(u(t-t_1) \otimes \cdots \otimes u(t-t_k)\big)\, \mathrm{d}\, t_k \cdots \mathrm{d}\, t_1, \tag{6}$$

where $g_k(t_1, \ldots, t_k)$ are the Volterra kernels of the corresponding representation. The kernels used in (6) are of the symmetric type [19, 34]. Applying the multivariate Laplace transformation [34] to (6) results in an equivalent description of the quadratic-bilinear system (3) in the frequency domain by multivariate transfer functions.

2.2 Subsystem transfer functions of quadratic-bilinear systems

In this work, we restrict ourselves to the presentation of the transfer functions corresponding to the first three coupled linear subsystems for brevity and practical relevance. General formulas for arbitrarily high levels of multivariate transfer functions for (3) have been developed in [39, Sec. 2.3.2].

2.2.1 Symmetric subsystem transfer functions

The symmetric subsystem transfer functions are based on the symmetric Volterra kernels from [34]; cf. (6). Historically, this is the first transfer function type that has been investigated for the model reduction of (3) in [19]. Here, the term "symmetric" refers to the fact that the transfer functions are invariant with respect to the order of their frequency arguments.

The first symmetric subsystem transfer function corresponds to the linear part of (3) as it can be seen in (5) such that

$$G_{\mathrm{sym},1}(s_1) = C g_{\mathrm{sym},1}(s_1) = C(s_1 E - A)^{-1} B, \tag{7}$$

with $s_1 \in \mathbb{C}$. Thereby, the term $g_{\mathrm{sym},1}(s_1) \in \mathbb{C}^{n \times m}$ is used in the following for notational convenience and denotes the input-to-state transition of the first subsystem.

The second symmetric subsystem transfer function depends on two complex frequency arguments $s_1, s_2 \in \mathbb{C}$ and is given by

$$G_{\mathrm{sym},2}(s_1, s_2) = C g_{\mathrm{sym},2}(s_1, s_2), \tag{8}$$

where the function describing the input-to-state transition on the right-hand side of (8) is given by

$$\begin{aligned} g_{\mathrm{sym},2}(s_1, s_2) = \frac{1}{2}\big((s_1 + s_2)E - A\big)^{-1} \\ \times \Big(H\big(g_{\mathrm{sym},1}(s_1) \otimes g_{\mathrm{sym},1}(s_2) + g_{\mathrm{sym},1}(s_2) \otimes g_{\mathrm{sym},1}(s_1)\big) \\ + N\big(I_m \otimes \big(g_{\mathrm{sym},1}(s_1) + g_{\mathrm{sym},1}(s_2)\big)\big)\Big), \end{aligned}$$

with $g_{\mathrm{sym},1}$ from (7), I_m denoting the m-dimensional identity matrix and the column concatenation of the bilinear terms as

$$N = \begin{bmatrix} N_1 & N_2 & \dots & N_m \end{bmatrix}. \tag{9}$$

Last, the third symmetric subsystem transfer function is defined similarly to (8) by

$$G_{\mathrm{sym},3}(s_1, s_2, s_3) = C g_{\mathrm{sym},3}(s_1, s_2, s_3), \tag{10}$$

with $s_1, s_2, s_3 \in \mathbb{C}$ and the input-to-state transition given via

$$\begin{aligned} g_{\mathrm{sym},3}(s_1, s_2, s_3) = \frac{1}{6}\big((s_1 + s_2 + s_3)E - A\big)^{-1} \\ \times \Big(H\big(g_{\mathrm{sym},1}(s_1) \otimes g_{\mathrm{sym},2}(s_2, s_3) + g_{\mathrm{sym},1}(s_2) \otimes g_{\mathrm{sym},2}(s_1, s_3) \\ + g_{\mathrm{sym},1}(s_3) \otimes g_{\mathrm{sym},2}(s_1, s_2) + g_{\mathrm{sym},2}(s_1, s_2) \otimes g_{\mathrm{sym},1}(s_3) \\ + g_{\mathrm{sym},2}(s_1, s_3) \otimes g_{\mathrm{sym},1}(s_2) + g_{\mathrm{sym},2}(s_2, s_3) \otimes g_{\mathrm{sym},1}(s_1)\big) \\ + N\big(I_m \otimes \big(g_{\mathrm{sym},2}(s_1, s_2) + g_{\mathrm{sym},2}(s_1, s_3) + g_{\mathrm{sym},2}(s_2, s_3)\big)\big)\Big), \end{aligned}$$

using $g_{\mathrm{sym},1}$ from (7) and $g_{\mathrm{sym},2}$ from (8).

2.2.2 Generalized transfer functions

In contrast to the symmetric case, the generalized transfer functions do not directly correspond to a Volterra kernel representation. They have been introduced in [16] for the extension of the data-driven Loewner framework to quadratic-bilinear systems and are inspired by the regular transfer functions of bilinear systems, which only consist of products of the terms of the dynamical system; see, e.g., [3]. The formulation given in [16] for the single-input/single-output (SISO) system case has been extended to multi-input/multi-output systems in [39, Sec. 2.3.2].

As in the symmetric case, the first regular transfer function corresponds to the linear system components, with

$$G^{(\mathrm{B})}_{\mathrm{gen},1}(s_1) = C(s_1E - A)^{-1}B. \tag{11}$$

Also the second regular transfer function is uniquely defined resulting from one multiplication with the bilinear terms:

$$G^{(\mathrm{N},(\mathrm{B}))}_{\mathrm{gen},2}(s_1, s_2) = C(s_2E - A)^{-1}N\big(I_m \otimes (s_1E - A)^{-1}B\big). \tag{12}$$

For the third level however, two different generalized transfer functions exist. The first one is identical to the third regular bilinear transfer function with

$$\begin{aligned} G^{(\mathrm{N},(\mathrm{N},(\mathrm{B})))}_{\mathrm{gen},3}(s_1, s_2, s_3) = C(s_3E - A)^{-1}N\Big(I_m \otimes \big((s_2E - A)^{-1}N \\ \otimes (I_m \otimes (s_1E - A)^{-1}B)\big)\Big); \end{aligned} \tag{13}$$

see also [10]. The second one involves the quadratic term and is given by

$$G^{(\mathrm{H},(\mathrm{B}),(\mathrm{B}))}_{\mathrm{gen},3}(s_1, s_2, s_3) = C(s_3E - A)^{-1}H\big((s_2E - A)^{-1}B \otimes (s_1E - A)^{-1}B\big). \tag{14}$$

Note that the index levels of the generalized and symmetric transfer functions do not coincide since the second symmetric subsystem transfer function does contain the quadratic term in contrast to the second level generalized transfer function, which has only the bilinear terms.

2.3 Structured linear systems

Before deriving structured quadratic-bilinear transfer functions, we briefly recall the ideas from [5], which considered the structured transfer functions for linear dynamical systems in which frequency-dependent equations are used for describing the dynamics. First, consider linear first-order (unstructured) systems with the time domain representation

$$E\dot{x}(t) - Ax(t) = Bu(t), \quad y(t) = Cx(t). \tag{15}$$

By taking the Laplace transform of (15), the dynamical system in (15) can be equivalently described in the frequency domain as

$$(sE - A)X(s) = BU(s), \quad Y(s) = CX(s), \tag{16}$$

with $s \in \mathbb{C}$, and where $U(s)$, $X(s)$, $Y(s)$ are the Laplace transforms of the inputs $u(t)$, states $x(t)$, and outputs $y(t)$, respectively. Now consider a linear dynamical system with a second-order structure with the time domain representation

$$M\ddot{x}(t) + D\dot{x}(t) + Kx(t) = B_{\mathrm{u}}u(t), \quad y(t) = C_{\mathrm{p}}x(t) + C_{\mathrm{v}}\dot{x}(t). \tag{17}$$

As in the unstructured case, taking the Laplace transform of (17) yields the representation in the frequency domain

$$(s^2M + sD + K)X(s) = B_{\mathrm{u}}U(s), \quad Y(s) = (C_{\mathrm{p}} + sC_{\mathrm{v}})X(s). \tag{18}$$

Observe that in both cases of (16) and (18), the system states in the frequency domain can be described as solution of frequency-dependent linear systems of equations of the form

$$\mathcal{K}(s)X(s) = \mathcal{B}(s)U(s), \quad Y(s) = \mathcal{C}(s), \tag{19}$$

where the 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 $\mathcal{C}\colon \mathbb{C} \to \mathbb{C}^{p\times m}$ describe the linear dynamics, and the input and output behavior of the system. In particular, we recover (16) by setting $\mathcal{K}(s) = sE - A$, $\mathcal{B}(s) = B$, and $\mathcal{C}(s) = B$. Similarly, we recover (18) by setting $\mathcal{K}(s) = s^2M + sD + K$, $\mathcal{B}(s) = B_{\mathrm{u}}$, and $\mathcal{C}(s) = C_{\mathrm{p}} + sC_{\mathrm{v}}$. We refer the reader to [5] for further examples of structured dynamics that fit into the general framework of (19). Then, for every $s \in \mathbb{C}$ for which $\mathcal{K}(s)$ in (19) is invertible, the transfer function of the underlying linear dynamical system is given by

$$G(s) = \mathcal{C}(s)\mathcal{K}(s)^{-1}\mathcal{B}(s).$$

3 Structured transfer functions of quadratic-bilinear systems

In this section, we extend the transfer function formulations from Section 2 to the setting of structured quadratic-bilinear systems. We start with the illustrative example of quadratic-bilinear mechanical systems (2). Based on this motivation, we introduce the formulas for structured symmetric and generalized transfer functions before we consider the case of quadratic-bilinear time-delay systems as another motivating example for internal differential structures at the end of this section.

3.1 Transfer functions for mechanical systems

In general, any system in second-order form (2) can be rewritten in first-order form (3) using the concatenated first-order state $x(t) = \left[q(t)^{\mathsf{T}}\ \dot{q}(t)^{\mathsf{T}}\right]^{\mathsf{T}}$. The first-order system matrices are then, for example, given by

$$\begin{aligned} E &= \begin{bmatrix} I_n & 0 \\ 0 & M \end{bmatrix}, \quad A = \begin{bmatrix} 0 & I_n \\ -K & -D \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ B_{\mathrm{u}} \end{bmatrix}, \\ C &= \begin{bmatrix} C_{\mathrm{p}}\ C_{\mathrm{v}} \end{bmatrix}, \quad N_j = \begin{bmatrix} 0 & 0 \\ N_{\mathrm{p,j}} & N_{\mathrm{v,j}} \end{bmatrix}, \end{aligned} \tag{20}$$

for $j = 1, \ldots, m$, with the quadratic term

$$H = -\begin{bmatrix} 0 & 0 & \ldots & 0 & 0 & 0 & 0 & \ldots & 0 & 0 \\ H_{\mathrm{pp,1}} & H_{\mathrm{pv,1}} & \ldots & H_{\mathrm{pp,n}} & H_{\mathrm{pv,n}} & H_{\mathrm{vp,1}} & H_{\mathrm{vv,1}} & \ldots & H_{\mathrm{vp,n}} & H_{\mathrm{vv,n}} \end{bmatrix}. \tag{21}$$

Thereby, the matrix blocks in (21) are $n \times n$ matrix slices of the second-order quadratic terms in (2), e.g., for H_{pp} modeling the multiplication of the state with itself we have

$$H_{\mathrm{pp}} = \begin{bmatrix} H_{\mathrm{pp,1}} & H_{\mathrm{pp,2}} & \dots & H_{\mathrm{pp,n}} \end{bmatrix},$$

with $H_{\mathrm{pp,j}} \in \mathbb{R}^{n \times n}$ for all $j = 1, \dots, n$.

Now, we exploit the block structures of the matrices in (20) and (21) to derive the symmetric and generalized transfer functions of (2). In both transfer function cases, the first level transfer functions correspond to the linear system case and it can be observed that for (2) it holds that

$$\begin{aligned} G_{\mathrm{sym,1}}(s_1) = G^{(\mathrm{B})}_{\mathrm{gen,1}}(s_1) &= (C_{\mathrm{p}} + s_1 C_{\mathrm{v}}) g_{\mathrm{sym,1}}(s_1) \\ &= (C_{\mathrm{p}} + s_1 C_{\mathrm{v}})(s_1^2 M + s_1 D + K)^{-1} B_{\mathrm{u}}, \end{aligned} \tag{22}$$

where $g_{\mathrm{sym,1}}(s_1)$ denotes here the input-to-state transition of the second-order state q.

For the next two levels, we concentrate first on the symmetric transfer function case. Inserting (20) and (21) into (8) yields the second symmetric subsystem transfer function of (2) to be

$$\begin{aligned} G_{\mathrm{sym,2}}(s_1, s_2) &= \big(C_{\mathrm{p}} + (s_1 + s_2) C_{\mathrm{v}}\big) g_{\mathrm{sym,2}}(s_1, s_2) \\ &= \frac{1}{2}\big(C_{\mathrm{p}} + (s_1 + s_2) C_{\mathrm{v}}\big)\big((s_1 + s_2)^2 M + (s_1 + s_2) D + K\big)^{-1} \\ &\quad \times \Big(- (H_{\mathrm{pp}} + s_2 H_{\mathrm{pv}} + s_1 H_{\mathrm{vp}} + s_1 s_2 H_{\mathrm{vv}})\big(g_{\mathrm{sym,1}}(s_1) \otimes g_{\mathrm{sym,1}}(s_2)\big) \\ &\quad - (H_{\mathrm{pp}} + s_1 H_{\mathrm{pv}} + s_2 H_{\mathrm{vp}} + s_1 s_2 H_{\mathrm{vv}})\big(g_{\mathrm{sym,1}}(s_2) \otimes g_{\mathrm{sym,1}}(s_1)\big) \\ &\quad + (N_{\mathrm{p}} + s_1 N_{\mathrm{v}})\big(I_m \otimes g_{\mathrm{sym,1}}(s_1)\big) \\ &\quad + (N_{\mathrm{p}} + s_2 N_{\mathrm{v}})\big(I_m \otimes g_{\mathrm{sym,1}}(s_2)\big)\Big), \end{aligned} \tag{23}$$

where $g_{\mathrm{sym,2}}$ denotes the input-to-state transition of the second subsystem, $g_{\mathrm{sym,1}}$ is the input-to-state transition from the first subsystem in (22), and the bilinear terms are concatenated as

$$N_{\mathrm{p}} = \begin{bmatrix} N_{\mathrm{p,1}} & \dots & N_{\mathrm{p,m}} \end{bmatrix} \quad \text{and} \quad N_{\mathrm{v}} = \begin{bmatrix} N_{\mathrm{v,1}} & \dots & N_{\mathrm{v,m}} \end{bmatrix}. \tag{24}$$

Similarly, inserting the block matrices (20) and (21) into the unstructured third symmetric subsystem transfer function (10) leads to the structured third symmetric subsystem transfer function representation for (2). Due to the complexity of the resulting formula, we will not write it out here explicitly but outline some of its features. The third subsystem transfer function of (3) has a similar structure to (10) and (23) with linear combinations of the quadratic and bilinear terms multiplied with the previous input-to-state transition terms $g_{\mathrm{sym,1}}$ and $g_{\mathrm{sym,2}}$ from (22) and (23). Due to the occurrence of the sum of the frequency arguments in the first term of (23), this translates into the frequency dependence of the second-order quadratic and bilinear terms such

that terms of the forms

$$\begin{aligned} &-\big(H_{\mathrm{pp}} + s_3 H_{\mathrm{pv}} + (s_1+s_2) H_{\mathrm{vp}} + (s_1+s_2) s_3 H_{\mathrm{vv}}\big)\big(g_{\mathrm{sym},2}(s_1,s_2) \otimes g_{\mathrm{sym},1}(s_3)\big) \\ &\text{and} \quad \big(N_{\mathrm{p}} + (s_1+s_2) N_{\mathrm{v}}\big)\big(I_m \otimes g_{\mathrm{sym},2}(s_1,s_2)\big) \end{aligned} \tag{25}$$

appear. For more details, we refer the reader to the next section, which contains the definitions of the transfer function formulas for general structures.

For the generalized transfer functions, one can observe that the second level transfer function resembles the corresponding bilinear regular transfer function, for which the structured system case has been developed in [10]. The resulting transfer function for (2) is thereby given as

$$\begin{aligned} G_{\mathrm{gen},2}^{(\mathrm{N},(\mathrm{B}))}(s_1,s_2) = (C_{\mathrm{p}} + s_2 C_{\mathrm{v}})(s_2^2 M + s_2 D + K)^{-1}(N_{\mathrm{p}} + s_1 N_{\mathrm{v}}) \\ \times \big(I_m \otimes (s_1^2 M + s_1 D + K)^{-1} B_{\mathrm{u}}\big), \end{aligned} \tag{26}$$

where the bilinear terms are concatenated as in (24). Similarly, the purely bilinear third level generalized transfer function of (2) is given by

$$\begin{aligned} G_{\mathrm{gen},3}^{(\mathrm{N},(\mathrm{N},(\mathrm{B})))} = (C_{\mathrm{p}} + s_3 C_{\mathrm{v}})(s_3^2 M + s_3 D + K)^{-1}(N_{\mathrm{p}} + s_2 N_{\mathrm{v}}) \\ \times \Big(I_m \otimes (s_2^2 M + s_2 D + K)^{-1}(N_{\mathrm{p}} + s_1 N_{\mathrm{v}}) \\ \times \big(I_m \otimes (s_1^2 M + s_1 D + K)^{-1} B_{\mathrm{u}}\big)\Big). \end{aligned} \tag{27}$$

On the other hand, the third level generalized transfer function of (2) involving the quadratic term can be derived by inserting (20) into (14), which yields

$$\begin{aligned} G_{\mathrm{gen},3}^{(\mathrm{H},(\mathrm{B}),(\mathrm{B}))}(s_1,s_2,s_3) = -(C_{\mathrm{p}} + s_3 C_{\mathrm{v}})(s_3^2 M + s_3 D + K)^{-1} \\ \times (H_{\mathrm{pp}} + s_1 H_{\mathrm{pv}} + s_2 H_{\mathrm{vp}} + s_1 s_2 H_{\mathrm{vv}}) \\ \times \big((s_2^2 M + s_2 D + K)^{-1} B_{\mathrm{u}} \otimes (s_1^2 M + s_1 D + K)^{-1} B_{\mathrm{u}}\big). \end{aligned} \tag{28}$$

Overall, and similar to the linear and bilinear cases, we can observe the occurrence of the same terms describing the linear, bilinear or quadratic dynamics in the different transfer functions by means of the given system matrices and the complex variables s_1, s_2, s_3. This motivates the definitions for the general structured framework in the upcoming section.

3.2 Structured transfer function formulas for quadratic-bilinear systems

Recently, an extension of structured transfer functions described in Section 2.3 to bilinear systems has been proposed in [10]. For this extension, a new frequency-dependent function $\mathcal{N}\colon \mathbb{C} \to \mathbb{C}^{n\times nm}$ was introduced, modeling the effect of the bilinear terms, where

$$\mathcal{N}(s) = \big[\mathcal{N}_1(s) \ \ldots \ \mathcal{N}_m(s)\big],$$

with $\mathcal{N}_j\colon \mathbb{C} \to \mathbb{C}^{n\times n}$ for all $j = 1, \ldots, m$. In this manuscript, we further extend the structured transfer function framework to the quadratic-bilinear case, which appears ubiquitously in prominent applications as we briefly discussed in Section 1.

First, we consider the symmetric transfer function case. Inspired by (7), (8), and (10), from the unstructured first-order case, and (22), (23), and (25), we introduce the following definition for the structured symmetric subsystem transfer functions.

Definition 1 Given matrix-valued functions of the form $\mathcal{C}\colon \mathbb{C} \to \mathbb{C}^{p\times m}$, $\mathcal{K}\colon \mathbb{C} \to \mathbb{C}^{n\times n}$, $\mathcal{B}\colon \mathbb{C} \to \mathbb{C}^{n\times m}$, $\mathcal{N}\colon \mathbb{C} \to \mathbb{C}^{n\times nm}$, $\mathcal{H}\colon \mathbb{C}\times\mathbb{C} \to \mathbb{C}^{n\times n^2}$, for which there exists an $s \in \mathbb{C}$ at which they can be evaluated and $\mathcal{K}(s)$ is invertible. The first three *structured symmetric subsystem transfer functions* are defined as

$$\begin{aligned} G_{\mathrm{sym},1}(s_1) &= \mathcal{C}(s_1)g_{\mathrm{sym},1}(s_1),\\ G_{\mathrm{sym},2}(s_1,s_2) &= \mathcal{C}(s_1+s_2)g_{\mathrm{sym},2}(s_1,s_2),\\ G_{\mathrm{sym},3}(s_1,s_2,s_3) &= \mathcal{C}(s_1+s_2+s_3)g_{\mathrm{sym},3}(s_1,s_2,s_3), \end{aligned}$$

where the input-to-state transitions are recursively given by

$$\begin{aligned} g_{\mathrm{sym},1}(s_1) &= \mathcal{K}(s_1)^{-1}\mathcal{B}(s_1),\\ g_{\mathrm{sym},2}(s_1,s_2) &= \frac{1}{2}\mathcal{K}(s_1+s_2)^{-1}\Big(\mathcal{H}(s_1,s_2)\big(g_{\mathrm{sym},1}(s_1)\otimes g_{\mathrm{sym},1}(s_2)\big)\\ &\quad + \mathcal{H}(s_2,s_1)\big(g_{\mathrm{sym},1}(s_2)\otimes g_{\mathrm{sym},1}(s_1)\big) + \mathcal{N}(s_1)\big(I_m\otimes g_{\mathrm{sym},1}(s_1)\big)\\ &\quad + \mathcal{N}(s_2)\big(I_m\otimes g_{\mathrm{sym},1}(s_2)\big)\Big),\\ g_{\mathrm{sym},3}(s_1,s_2,s_3) &= \frac{1}{6}\mathcal{K}(s_1+s_2+s_3)^{-1}\Big(\mathcal{H}(s_1+s_2,s_3)\big(g_{\mathrm{sym},2}(s_1,s_2)\otimes g_{\mathrm{sym},1}(s_3)\big)\\ &\quad + \mathcal{H}(s_1+s_3,s_2)\big(g_{\mathrm{sym},2}(s_1,s_3)\otimes g_{\mathrm{sym},1}(s_2)\big)\\ &\quad + \mathcal{H}(s_2+s_3,s_1)\big(g_{\mathrm{sym},2}(s_2,s_3)\otimes g_{\mathrm{sym},1}(s_1)\big)\\ &\quad + \mathcal{H}(s_1,s_2+s_3)\big(g_{\mathrm{sym},1}(s_1)\otimes g_{\mathrm{sym},2}(s_2,s_3)\big)\\ &\quad + \mathcal{H}(s_2,s_1+s_3)\big(g_{\mathrm{sym},1}(s_2)\otimes g_{\mathrm{sym},2}(s_1,s_3)\big)\\ &\quad + \mathcal{H}(s_3,s_1+s_2)\big(g_{\mathrm{sym},1}(s_3)\otimes g_{\mathrm{sym},2}(s_1,s_2)\big)\\ &\quad + \mathcal{N}(s_1+s_2)\big(I_m\otimes g_{\mathrm{sym},2}(s_1,s_2)\big)\\ &\quad + \mathcal{N}(s_1+s_3)\big(I_m\otimes g_{\mathrm{sym},2}(s_1,s_3)\big)\\ &\quad + \mathcal{N}(s_2+s_3)\big(I_m\otimes g_{\mathrm{sym},2}(s_2,s_3)\big)\Big). \end{aligned}$$

Similarly, we give a definition for the structured variant of the generalized transfer functions inspired by the first-order case (11)–(14), and second-order case (22) and (26)–(28) in the following.

Definition 2 Given matrix-valued functions of the form $\mathcal{C}\colon \mathbb{C} \to \mathbb{C}^{p\times m}$, $\mathcal{K}\colon \mathbb{C} \to \mathbb{C}^{n\times n}$, $\mathcal{B}\colon \mathbb{C} \to \mathbb{C}^{n\times m}$, $\mathcal{N}\colon \mathbb{C} \to \mathbb{C}^{n\times nm}$, $\mathcal{H}\colon \mathbb{C}\times\mathbb{C} \to \mathbb{C}^{n\times n^2}$, for which there exists an $s \in \mathbb{C}$ at which they can be evaluated and $\mathcal{K}(s)$ is invertible. The first three levels

of *structured generalized transfer functions* are defined as

$$\begin{aligned}
G^{(\mathrm{B})}_{\mathrm{gen},1}(s_1) &= \mathcal{C}(s_1)\mathcal{K}(s_1)^{-1}\mathcal{B}(s_1),\\
G^{(\mathrm{N},(\mathrm{B}))}_{\mathrm{gen},2}(s_1,s_2) &= \mathcal{C}(s_2)\mathcal{K}(s_2)^{-1}\mathcal{N}(s_1)\big(I_m \otimes \mathcal{K}(s_1)^{-1}\mathcal{B}(s_1)\big),\\
G^{(\mathrm{N},(\mathrm{N},(\mathrm{B})))}_{\mathrm{gen},3}(s_1,s_2,s_3) &= \mathcal{C}(s_3)\mathcal{K}(s_3)^{-1}\mathcal{N}(s_2)\Big(I_m \otimes \mathcal{K}(s_2)^{-1}\mathcal{N}(s_1)\\
&\quad \times \big(I_m \otimes \mathcal{K}(s_1)^{-1}\mathcal{B}(s_1)\big)\Big),\\
G^{(\mathrm{H},(\mathrm{B}),(\mathrm{B}))}_{\mathrm{gen},3}(s_1,s_2,s_3) &= \mathcal{C}(s_3)\mathcal{K}(s_3)^{-1}\mathcal{H}(s_2,s_1)\big(\mathcal{K}(s_2)^{-1}\mathcal{B}(s_2) \otimes \mathcal{K}(s_1)^{-1}\mathcal{B}(s_1)\big).
\end{aligned}$$

Note that in both Definitions 1 and 2, the new matrix-valued function $\mathcal{H}\colon \mathbb{C}\times\mathbb{C} \to \mathbb{C}^{n\times n^2}$ results from the quadratic terms in the time domain. In [8], a simplified variant of the generalized transfer functions (11)–(14) has been extended to systems with polynomial nonlinearities. Based on the structured definitions above, these simplified generalized transfer functions have then been extended to the structured case in [18].

Both examples of internal system structures considered so far can be represented in the new structured transfer function framework. Unstructured first-order systems of the form (3) are given by

$$\mathcal{C}(s) = C, \quad \mathcal{K}(s) = sE - A, \quad \mathcal{B}(s) = B, \quad \mathcal{N}(s) = N, \quad \mathcal{H}(s_1,s_2) = H,$$

where the bilinear terms are concatenated as in (9). For the second-order system of the form (2), the symmetric and generalized transfer functions given in Section 3.1 can be recovered from Definitions 1 and 2 using

$$\begin{aligned}
\mathcal{C}(s) &= C_{\mathrm{p}} + sC_{\mathrm{v}},\\
\mathcal{K}(s) &= s^2M + sD + K,\\
\mathcal{B}(s) &= B_{\mathrm{u}},\\
\mathcal{N}(s) &= N_{\mathrm{p}} + sN_{\mathrm{v}},\\
\mathcal{H}(s_1,s_2) &= -(H_{\mathrm{pp}} + s_2H_{\mathrm{pv}} + s_1H_{\mathrm{vp}} + s_1s_2H_{\mathrm{vv}}),
\end{aligned}$$

with the bilinear terms concatenated as in (24). For the definition of higher level structured transfer functions for quadratic-bilinear systems see [39, Sec. 6.3].

3.3 Quadratic-bilinear time-delay systems

A different, commonly occurring differential structure in the model process of real-world phenomena are time delays. These are typically used to model the postponed reactions of dynamics [14]. Quadratic-bilinear systems with constant time delays in

the linear dynamic components can be written as

$$\begin{aligned} E\dot{x}(t) &= \sum_{k=1}^{\ell} A_k x(t-\tau_k) + H\big(x(t)\otimes x(t)\big) + \sum_{j=1}^{m} N_j x(t) u_j(t) + Bu(t), \\ y(t) &= Cx(t), \end{aligned} \tag{29}$$

with the matrices $A_k \in \mathbb{R}^{n\times n}$ describing the effect of state delayed by $\tau_k \in \mathbb{R}_{\geq 0}$, for all $k = 1, \ldots, \ell$, and the remaining system matrices as defined in (3). Following the variational analyses from (5), we observe that the time-delay structure only affects the terms with the linear dynamics. Therefore, the structured transfer functions for (29) are given by using the matrix-valued functions

$$\mathcal{C}(s) = C, \quad \mathcal{K}(s) = sE - \sum_{k=1}^{\ell} A_k e^{-\tau_k s}, \quad \mathcal{B}(s) = B, \quad \mathcal{N}(s) = N, \quad \mathcal{H}(s_1, s_2) = H$$

in Definitions 1 and 2. This is in accordance to the results for bilinear time-delay systems obtained in [10, 17].

The structure as given in (29) is used in different application and research areas. It appears in the design process of optoelectronic delayed feedback for the control of lasers [35] or in the analysis of nonlinear delayed vibrational systems [23]. Time-delay systems of the form (29) are also used in reaction-diffusion equations from the engineering and chemical sciences [20]. We consider such a reaction-diffusion example in our numerical experiments in Section 5.2.

4 Structured transfer function interpolation

In this section, we present results on the construction of structured interpolants for the symmetric or generalized transfer functions from the previous section.

4.1 Structure-preserving model reduction via projection

For the construction of interpolating reduced-order models, we will use the projection approach in this work. Thereby, two constant basis matrices $V, W \in \mathbb{C}^{n\times r}$ ware constructed, which allow the computation of the reduced-order quantities via multiplication with the original system matrices. Given the full-order matrix-valued functions $\mathcal{C}\colon \mathbb{C} \to \mathbb{C}^{p\times n}$, $\mathcal{K}\colon \mathbb{C} \to \mathbb{C}^{n\times n}$, $\mathcal{B}\colon \mathbb{C} \to \mathbb{C}^{n\times m}$, $\mathcal{N}\colon \mathbb{C} \to \mathbb{C}^{n\times nm}$ and $\mathcal{H}\colon \mathbb{C}\times\mathbb{C} \to \mathbb{C}^{n\times n^2}$ that describe a structured quadratic-bilinear system in the frequency domain, reduced-order model quantities are computed by

$$\begin{aligned} \widehat{\mathcal{C}}(s) &= \mathcal{C}V, & \widehat{\mathcal{K}}(s) &= W^{\mathsf{H}}\mathcal{K}(s)V, \\ \widehat{\mathcal{B}}(s) &= W^{\mathsf{H}}\mathcal{B}(s), & \widehat{\mathcal{N}}(s) &= W^{\mathsf{H}}\mathcal{N}(s)(I_m \otimes V) \\ \text{and} & & \widehat{\mathcal{H}}(s_1, s_2) &= W^{\mathsf{H}}\mathcal{H}(s_1, s_2)(V \otimes V), \end{aligned} \tag{30}$$

where $W^{\mathsf{H}} := \overline{W}^{\mathsf{T}}$ denotes the conjugate transpose of the matrix W. The Kronecker product in the multiplication with the concatenation of the bilinear terms in (30) boils down to the multiplication of each single bilinear term with the two basis matrices as

$$\widehat{\mathcal{N}}(s) = \begin{bmatrix} \widehat{\mathcal{N}}_1(s) & \dots & \widehat{\mathcal{N}}_m(s) \end{bmatrix} = \begin{bmatrix} W^{\mathsf{H}}\mathcal{N}_1(s)V & \dots & W^{\mathsf{H}}\mathcal{N}_m(s)V \end{bmatrix}.$$

Moreover, the Kronecker product of the basis matrix V for the reduction of the quadratic term in (30) can be implemented efficiently without explicitly forming $V \otimes V$, using techniques from tensor algebra; see, e.g., [6, 7, 39].

Model reduction by projection preserves internal structures by construction. Any matrix-valued function can be decomposed into frequency-affine form, e.g., in the case of the term describing the linear dynamics, it can be written as

$$\mathcal{K}(s) = \sum_{j=1}^{n_{\mathcal{K}}} h_j(s)\mathcal{K}_j, \tag{31}$$

with $n_{\mathcal{K}} \in \mathbb{N}$, some frequency-dependent scalar functions $h_j : \mathbb{C} \to \mathbb{C}$ and constant matrices $\mathcal{K}_j \in \mathbb{C}^{n\times n}$, for all $j = 1, \dots, n_{\mathcal{K}}$. The reduced-order matrix-valued function is then given by

$$\widehat{\mathcal{K}}(s) = W^{\mathsf{H}}\mathcal{K}(s)V = \sum_{j=1}^{n_{\mathcal{K}}} h_j(s)W^{\mathsf{H}}\mathcal{K}_j V = \sum_{j=1}^{n_{\mathcal{K}}} h_j(s)\widehat{\mathcal{K}}_j, \tag{32}$$

with the reduced-order constant matrices $\widehat{\mathcal{K}}_j \in \mathbb{C}^{r\times r}$, for all $j = 1, \dots, n_{\mathcal{K}}$. The frequency-dependent scalar functions in (32) are the same as in (31), i.e., the internal structure is preserved and the reduced-order matrices replace their high-dimensional counterparts from the original system to describe the reduced-order model.

To illustrate the computation of reduced-order quadratic-bilinear systems via projection, we consider the two motivational differential structures from the previous section. In the case of second-order quadratic-bilinear systems (2), reduced-order systems are computed as

$$\begin{aligned} \widehat{\mathcal{C}}(s) &= C_{\mathrm{p}}V + sC_{\mathrm{v}}V, & \widehat{\mathcal{K}}(s) &= s^2W^{\mathsf{H}}MV + W^{\mathsf{H}}DV + W^{\mathsf{H}}KV, \\ \widehat{\mathcal{B}}(s) &= W^{\mathsf{H}}B_{\mathrm{u}}, & \widehat{\mathcal{N}}(s) &= W^{\mathsf{H}}N_{\mathrm{p}}(I_m \otimes V) + sW^{\mathsf{H}}N_{\mathrm{v}}(I_m \otimes V), \end{aligned}$$

with the reduced quadratic terms given by

$$\begin{aligned} \widehat{\mathcal{H}}(s_1, s_2) = -(&W^{\mathsf{H}}H_{\mathrm{pp}}(V \otimes V) + s_2W^{\mathsf{H}}H_{\mathrm{pv}}(V \otimes V) \\ &+ s_1W^{\mathsf{H}}H_{\mathrm{vp}}(V \otimes V) + s_1s_2W^{\mathsf{H}}H_{\mathrm{vv}}(V \otimes V)). \end{aligned}$$

Evaluating the matrix products yields the reduced-order matrices that represent the reduced-order system in the same structure as the original system (2). Similarly, for

the quadratic-bilinear time-delay system (29), reduced-order systems are computed via

$$\widehat{\mathcal{C}}(s) = CV, \qquad \widehat{\mathcal{K}}(s) = sW^{\mathsf{H}}EV - \sum_{k=1}^{\ell} W^{\mathsf{H}}A_kVe^{-\tau_k s},$$
$$\widehat{\mathcal{B}}(s) = W^{\mathsf{H}}B, \qquad \widehat{\mathcal{N}}(s) = W^{\mathsf{H}}N(I_m \otimes V)$$
$$\text{and} \qquad \widehat{\mathcal{H}}(s_1, s_2) = W^{\mathsf{H}}H(V \otimes V).$$

As in the second-order system case, evaluating the matrix products allows to replace the original, high-dimensional system matrices in (29) by the reduced ones to describe the reduced-order system using the same structure.

The essential question of projection-based model order reduction is the construction of the basis matrices V and W. In the following, conditions are derived to enforce interpolation of the original symmetric or generalized transfer functions by the corresponding transfer functions given via the reduced matrix-valued functions (30).

4.2 Interpolating symmetric transfer functions

In this section, we consider the interpolation of the structured symmetric subsystem transfer functions from Definition 1. The following proposition states some first results for the general interpolation of the first two symmetric subsystem transfer functions at different frequency points.

Proposition 1 ([39], Cor. 6.3) *Let G be a quadratic-bilinear system, described by its symmetric subsystem transfer functions $G_{\mathrm{sym},k}$ from* Definition 1*, and $\widehat{G}$ the reduced-order quadratic-bilinear system constructed by* (30)*, with its reduced-order symmetric subsystem transfer functions $\widehat{G}_{\mathrm{sym},k}$. Also, let $\sigma_1, \sigma_2 \in \mathbb{C}$ be interpolation points such that the matrix-valued functions $\mathcal{C}, \mathcal{K}, \mathcal{B}, \mathcal{N}, \mathcal{H}$ and $\mathcal{K}(.)^{-1}$ are defined in these points and their sum. Construct the basis matrix V by*

$$\begin{aligned}
V_{1,1} &= \mathcal{K}(\sigma_1)^{-1}\mathcal{B}(\sigma_1),\\
V_{1,2} &= \mathcal{K}(\sigma_2)^{-1}\mathcal{B}(\sigma_2),\\
V_2 &= \mathcal{K}(\sigma_1+\sigma_2)^{-1}\big(\mathcal{H}(\sigma_1,\sigma_2)(V_{1,1}\otimes V_{1,2}) + \mathcal{H}(\sigma_2,\sigma_1)(V_{1,2}\otimes V_{1,1})\\
&\quad + \mathcal{N}(\sigma_1)(I_m \otimes V_{1,1}) + \mathcal{N}(\sigma_2)(I_m \otimes V_{1,2})\big),\\
\operatorname{span}(V) &\supseteq \operatorname{span}\left(\begin{bmatrix} V_{1,1} & V_{1,2} & V_2 \end{bmatrix}\right),
\end{aligned}$$

and let W be an arbitrary full-rank matrix of appropriate dimensions. Then, the symmetric subsystem transfer functions of $\widehat{G}$ interpolate those of G in the following way:

$$\begin{aligned}
G_{\mathrm{sym},1}(\sigma_1) &= \widehat{G}_{\mathrm{sym},1}(\sigma_1),\\
G_{\mathrm{sym},1}(\sigma_2) &= \widehat{G}_{\mathrm{sym},1}(\sigma_2),\\
G_{\mathrm{sym},2}(\sigma_1,\sigma_2) &= \widehat{G}_{\mathrm{sym},1}(\sigma_1,\sigma_2).
\end{aligned}$$

Note that using [39, Thm. 6.2], the basis matrix V in Proposition 1 can be extended such that the third symmetric subsystem transfer function is interpolated as well. In practice however, this leads to an exponential increase in the number of terms to evaluate for the construction of the projection space, which is typically undesired due to its computational complexity. Therefore, this result is omitted here.

The next proposition considers a similar interpolation result as in Proposition 1 by setting conditions on the second basis matrix as well.

Proposition 2 ([39], Lem. 6.4) *Given the same assumptions as in* Proposition 1*, let the matrices $V_{1,1}$ and $V_{1,2}$ be as in* Proposition 1*. Construct the two basis matrices such that*

$$\operatorname{span}(V) \supseteq \operatorname{span}\left(\begin{bmatrix} V_{1,1} & V_{1,2} \end{bmatrix}\right),$$
$$\operatorname{span}(W) \supseteq \operatorname{span}\left(\mathcal{K}(\sigma_1+\sigma_2)^{-\mathsf{H}}\mathcal{C}(\sigma_1+\sigma_2)^{\mathsf{H}}\right),$$

and let V and W be of the same dimension. Then, the symmetric subsystem transfer functions of $\widehat{G}$ interpolate those of G in the following way:

$$\begin{aligned} G_{\mathrm{sym},1}(\sigma_1) &= \widehat{G}_{\mathrm{sym},1}(\sigma_1), \\ G_{\mathrm{sym},1}(\sigma_2) &= \widehat{G}_{\mathrm{sym},1}(\sigma_2), \\ G_{\mathrm{sym},1}(\sigma_1+\sigma_2) &= \widehat{G}_{\mathrm{sym},1}(\sigma_1+\sigma_2), \\ G_{\mathrm{sym},2}(\sigma_1,\sigma_2) &= \widehat{G}_{\mathrm{sym},2}(\sigma_1,\sigma_2). \end{aligned}$$

The result of Proposition 2 allows to enforce the same and more interpolation conditions than in Proposition 1 in an implicit way using the second basis matrix. This reduces the computational complexity of the construction of the basis matrices, since no nonlinear terms are involved, and allows to match more interpolation conditions with smaller reduced-order models.

The choice of interpolation points for the different subsystem levels is crucial for the quality of the computed reduced-order model. Good or even optimal choices of interpolation points are currently unknown. However, an advantageous choice to minimize the amount of basis contributions necessary for the interpolation of higher level subsystem transfer functions in the symmetric case is $\sigma_1 = \sigma_2 = \sigma_3 = \sigma$. The following theorem states the interpolation conditions for this particular selection of interpolation points and also gives conditions for the interpolation of the third subsystem transfer function.

Theorem 1 *Let G be a quadratic-bilinear system, described by its symmetric subsystem transfer functions $G_{\mathrm{sym},k}$ as in* Definition 1*, and $\widehat{G}$ the reduced-order quadratic-bilinear system constructed by* (30)*, with its reduced-order symmetric subsystem transfer functions $\widehat{G}_{\mathrm{sym},k}$. Also, let $\sigma \in \mathbb{C}$ be an interpolation point such that the matrix-valued functions $\mathcal{C}, \mathcal{K}, \mathcal{B}, \mathcal{N}, \mathcal{H}$ and $\mathcal{K}(.)^{-1}$ are defined at σ as well as at 2σ and 3σ. Construct the matrices*

$$\begin{aligned} V_1 &= \mathcal{K}(\sigma)^{-1}\mathcal{B}(\sigma), \\ V_2 &= \mathcal{K}(2\sigma)^{-1}\big(\mathcal{H}(\sigma,\sigma)(V_1\otimes V_1) + \mathcal{N}(\sigma)(I_m\otimes V_1)\big), \\ V_3 &= \mathcal{K}(3\sigma)^{-1}\big(\mathcal{H}(2\sigma,\sigma)(V_2\otimes V_1) + \mathcal{H}(\sigma,2\sigma)(V_1\otimes V_2) + \mathcal{N}(2\sigma)(I_m\otimes V_2)\big), \end{aligned}$$

and

$$W_1 = \mathcal{K}(2\sigma)^{-\mathsf{H}}\mathcal{C}(2\sigma)^{\mathsf{H}},$$
$$W_2 = \mathcal{K}(3\sigma)^{-\mathsf{H}}\mathcal{C}(3\sigma)^{\mathsf{H}}.$$

Then, the following statements hold true:

(a) If the basis matrix V is such that

$$\operatorname{span}(V) \supseteq \operatorname{span}\left(\begin{bmatrix} V_1 & V_2 & V_3 \end{bmatrix}\right),$$

and W is full-rank and of the same dimension as V, then the symmetric transfer functions of $\widehat{G}$ interpolate those of G in the following way:

$$G_{\mathrm{sym},1}(\sigma) = \widehat{G}_{\mathrm{sym},1}(\sigma),$$
$$G_{\mathrm{sym},2}(\sigma,\sigma) = \widehat{G}_{\mathrm{sym},2}(\sigma,\sigma),$$
$$G_{\mathrm{sym},3}(\sigma,\sigma,\sigma) = \widehat{G}_{\mathrm{sym},3}(\sigma,\sigma,\sigma).$$

(b) If the basis matrices V and W are such that

$$\operatorname{span}(V) \supseteq \operatorname{span}(V_1) \quad \textit{and} \quad \operatorname{span}(W) \supseteq \operatorname{span}(W_1),$$

and have the same dimension, then the symmetric transfer functions of $\widehat{G}$ interpolate those of G in the following way:

$$G_{\mathrm{sym},1}(\sigma) = \widehat{G}_{\mathrm{sym},1}(\sigma),$$
$$G_{\mathrm{sym},1}(2\sigma) = \widehat{G}_{\mathrm{sym},1}(2\sigma),$$
$$G_{\mathrm{sym},2}(\sigma,\sigma) = \widehat{G}_{\mathrm{sym},2}(\sigma,\sigma).$$

(c) If the basis matrices V and W are such that

$$\operatorname{span}(V) \supseteq \operatorname{span}\left(\begin{bmatrix} V_1 & V_2 \end{bmatrix}\right) \quad \textit{and} \quad \operatorname{span}(W) \supseteq \operatorname{span}(W_2),$$

and both of appropriate dimensions, then the symmetric transfer functions of $\widehat{G}$ interpolate those of G in the following way:

$$G_{\mathrm{sym},1}(\sigma) = \widehat{G}_{\mathrm{sym},1}(\sigma),$$
$$G_{\mathrm{sym},1}(3\sigma) = \widehat{G}_{\mathrm{sym},1}(3\sigma),$$
$$G_{\mathrm{sym},2}(\sigma,\sigma) = \widehat{G}_{\mathrm{sym},2}(\sigma,\sigma),$$
$$G_{\mathrm{sym},3}(\sigma,\sigma,\sigma) = \widehat{G}_{\mathrm{sym},3}(\sigma,\sigma,\sigma).$$

Proof Part (a) is an extension of Proposition 1 to the third symmetric subsystem transfer function with the special selection of interpolation points and follows immediately from [39, Thm. 6.2]. Part (b) follows directly from Proposition 2 such that only Part (c) is left to be proven. The first three interpolation conditions follow from previous results,

therefore we concentrate on the third symmetric subsystem transfer function. Inserting the selection of interpolation points into Definition 1 yields

$$\begin{aligned}\widehat{G}_{\mathrm{sym},3}(\sigma,\sigma,\sigma) = \frac{1}{2}\widehat{\mathcal{C}}(3\sigma)\widehat{\mathcal{K}}(3\sigma)^{-1}\big(&\widehat{\mathcal{H}}(2\sigma,\sigma)(\widehat{V}_2\otimes\widehat{V}_1)\\ &+\widehat{\mathcal{H}}(\sigma,2\sigma)(\widehat{V}_1\otimes\widehat{V}_2)+\widehat{\mathcal{N}}(2\sigma)(I_m\otimes\widehat{V}_2)\big),\end{aligned}$$

where

$$\begin{aligned}\widehat{V}_1 &= \widehat{\mathcal{K}}(\sigma)^{-1}\widehat{\mathcal{B}}(\sigma) \quad \text{and}\\ \widehat{V}_2 &= \widehat{\mathcal{K}}(2\sigma)^{-1}\big(\widehat{\mathcal{H}}(\sigma,\sigma)(\widehat{V}_1\otimes\widehat{V}_1)+\widehat{\mathcal{N}}(\sigma)(I_m\otimes\widehat{V}_1)\big).\end{aligned}$$

Using the projector $P_{\mathrm{V}} = V(W^{\mathsf{H}}\mathcal{K}(\sigma)V)^{-1}W^{\mathsf{H}}\mathcal{K}(\sigma)$ onto the space spanned by the columns of V, it follows that

$$V\widehat{V}_1 = V_1 \quad \text{and} \quad V\widehat{V}_2 = V_2$$

hold due to the construction of the space spanned by the columns of V via the columns of V_1 and V_2. Using the projector $P_{\mathrm{W}} = W(W^{\mathsf{H}}\mathcal{K}(\sigma)V)^{-\mathsf{H}}V^{\mathsf{H}}\mathcal{K}(\sigma)^{\mathsf{H}}$ onto the space spanned by the columns of W, it also holds that

$$\widehat{\mathcal{C}}(\sigma)\widehat{\mathcal{K}}(\sigma)^{-1}W^{\mathsf{H}} = \mathcal{C}(\sigma)\mathcal{K}(\sigma)^{-1},$$

such that the final result of the theorem holds via

$$\begin{aligned}\widehat{G}_{\mathrm{sym},3}(\sigma,\sigma,\sigma) &= \frac{1}{2}\widehat{\mathcal{C}}(3\sigma)\widehat{\mathcal{K}}(3\sigma)^{-1}W^{\mathsf{H}}\big(\mathcal{H}(2\sigma,\sigma)(V\widehat{V}_2\otimes V\widehat{V}_1)\\ &\quad+\mathcal{H}(\sigma,2\sigma)(V\widehat{V}_1\otimes V\widehat{V}_2)+\mathcal{N}(2\sigma)(I_m\otimes V\widehat{V}_2)\big)\\ &= \frac{1}{2}\mathcal{C}(3\sigma)\mathcal{K}(3\sigma)^{-1}\big(\mathcal{H}(2\sigma,\sigma)(V_2\otimes V_1)\\ &\quad+\mathcal{H}(\sigma,2\sigma)(V_1\otimes V_2)+\mathcal{N}(2\sigma)(I_m\otimes V_2)\big)\\ &= G_{\mathrm{sym},3}(\sigma,\sigma,\sigma).\end{aligned}$$

4.3 Interpolating generalized transfer functions

In this section, we investigate conditions on the projection spaces for the interpolation of the structured generalized transfer functions from Definition 2. The following proposition can be seen as an analog to Proposition 1 for the generalized case.

Proposition 3 ([39], Thm. 6.13) *Let G be a quadratic-bilinear system, described by its generalized transfer functions $G^{(.)}_{\mathrm{gen},k}$ from* Definition 2*, and $\widehat{G}$ the reduced-order quadratic-bilinear system constructed by* (30)*, with its reduced-order generalized transfer functions $\widehat{G}^{(.)}_{\mathrm{gen},k}$. Also, let $\sigma_1,\sigma_2,\sigma_3 \in \mathbb{C}$ be interpolation points such that the matrix-valued functions $\mathcal{C},\mathcal{K},\mathcal{B},\mathcal{N},\mathcal{H}$ and $\mathcal{K}(.)^{-1}$ are defined at these points.*

Compute

$$\begin{aligned} V_{1,1} &= \mathcal{K}(\sigma_1)^{-1}\mathcal{B}(\sigma_1), \\ V_{1,2} &= \mathcal{K}(\sigma_2)^{-1}\mathcal{B}(\sigma_2), \\ V_2 &= \mathcal{K}(\sigma_2)^{-1}\mathcal{N}(\sigma_1)(I_m \otimes V_{1,1}), \\ V_{3,1} &= \mathcal{K}(\sigma_3)^{-1}\mathcal{N}(\sigma_2)(I_m \otimes V_2), \\ V_{3,2} &= \mathcal{K}(\sigma_3)^{-1}\mathcal{H}(\sigma_2, \sigma_1)(V_{1,2} \otimes V_{1,1}), \end{aligned}$$

and construct the basis matrix V such that

$$\operatorname{span}(V) \supseteq \operatorname{span}\left(\begin{bmatrix} V_{1,1} & V_{1,2} & V_2 & V_{3,1} & V_{3,2} \end{bmatrix}\right).$$

Let W be an arbitrary full-rank matrix of appropriate dimensions. Then, the generalized transfer functions of $\widehat{G}$ interpolate those of G in the following way:

$$\begin{aligned} G_{\text{gen},1}^{(\text{B})}(\sigma_1) &= \widehat{G}_{\text{gen},1}^{(\text{B})}(\sigma_1), \\ G_{\text{gen},1}^{(\text{B})}(\sigma_2) &= \widehat{G}_{\text{gen},1}^{(\text{B})}(\sigma_2), \\ G_{\text{gen},2}^{(\text{N},(\text{B}))}(\sigma_1, \sigma_2) &= \widehat{G}_{\text{gen},2}^{(\text{N},(\text{B}))}(\sigma_1, \sigma_2), \\ G_{\text{gen},3}^{(\text{N},(\text{N},(\text{B})))}(\sigma_1, \sigma_2, \sigma_3) &= \widehat{G}_{\text{gen},3}^{(\text{N},(\text{N},(\text{B})))}(\sigma_1, \sigma_2, \sigma_3), \\ G_{\text{gen},3}^{(\text{H},(\text{B}),(\text{B}))}(\sigma_1, \sigma_2, \sigma_3) &= \widehat{G}_{\text{gen},3}^{(\text{H},(\text{B}),(\text{B}))}(\sigma_1, \sigma_2, \sigma_3). \end{aligned}$$

Remark 1 As for the symmetric subsystem transfer functions, it is possible to reduce the dimensions of the constructed projection space by choosing suitable interpolation points and, also as in the symmetric subsystem transfer function case, this can be achieved by choosing $\sigma_1 = \sigma_2 = \sigma_3$. However, only marginal savings in terms of basis contributions and computational costs can be achieved by this since in Proposition 3, only the matrix $V_{1,2}$ can be omitted.

Note that in Proposition 3, the basis contribution from $V_{3,1}$ results in the interpolation of the third generalized transfer function with two bilinear terms $G_{\text{gen},3}^{(\text{N},(\text{N},(\text{B})))}$. It may happen that no interpolation conditions are imposed for this transfer function such that the subspace dimensions can be reduced by omitting $V_{3,1}$. As in the case of symmetric subsystem transfer functions, the second basis matrix W can be used to reduce the minimal dimension of the constructed subspaces and to simplify the construction of the subspaces. These results are given in the following corollary.

Corollary 1 *Let G be a quadratic-bilinear system, described by its generalized transfer functions $G_{\text{gen},k}^{(.)}$ from Definition 2, and $\widehat{G}$ the reduced-order quadratic-bilinear system constructed by (30), with its reduced-order generalized transfer functions $\widehat{G}_{\text{gen},k}^{(.)}$. Also, let $\sigma_1, \sigma_2 \in \mathbb{C}$ be interpolation points such that the matrix-valued functions $\mathcal{C}, \mathcal{K}, \mathcal{B}, \mathcal{N}, \mathcal{H}$ and $\mathcal{K}(.)^{-1}$ are defined at these points. Let the basis matrices V and*

W be constructed by

$$\operatorname{span}(V) \supseteq \operatorname{span}\left(\mathcal{K}(\sigma_1)^{-1}\mathcal{B}(\sigma_1)\right),$$
$$\operatorname{span}(W) \supseteq \operatorname{span}\left(\mathcal{K}(\sigma_2)^{-\mathsf{H}}\mathcal{C}(\sigma_2)^{\mathsf{H}}\right),$$

and are of the same dimension. Then, the generalized transfer functions of $\widehat{G}$ *interpolate those of* G *in the following way:*

$$\begin{aligned} G^{(\mathrm{B})}_{\mathrm{gen},1}(\sigma_1) &= \widehat{G}^{(\mathrm{B})}_{\mathrm{gen},1}(\sigma_1), \\ G^{(\mathrm{B})}_{\mathrm{gen},1}(\sigma_2) &= \widehat{G}^{(\mathrm{B})}_{\mathrm{gen},1}(\sigma_2), \\ G^{(\mathrm{N},(\mathrm{B}))}_{\mathrm{gen},2}(\sigma_1,\sigma_2) &= \widehat{G}^{(\mathrm{N},(\mathrm{B}))}_{\mathrm{gen},2}(\sigma_1,\sigma_2), \\ G^{(\mathrm{H},(\mathrm{B}),(\mathrm{B}))}_{\mathrm{gen},3}(\sigma_1,\sigma_1,\sigma_2) &= \widehat{G}^{(\mathrm{H},(\mathrm{B}),(\mathrm{B}))}_{\mathrm{gen},3}(\sigma_1,\sigma_1,\sigma_2). \end{aligned}$$

Proof The result follows directly from [39, Lem. 6.15] by restriction to two interpolation points.

Similar to Proposition 2, the result in Corollary 1 states that the interpolation of higher level transfer functions is possible in an implicit way without evaluating any of the nonlinear terms. Corollary 1 shows the version of the implicit interpolation result with the smallest achievable minimal subspace dimensions for the interpolation of the third level generalized transfer function with quadratic term. The choice of identical interpolation points will not further reduce the dimensions of the projection spaces, but allows to replace the interpolation of the first level generalized transfer function in two points by matching the transfer function value and its derivative in one point; see [5] and [39, Lem. 6.15].

5 Numerical experiments

Now, we employ the interpolation results from above for constructing structured reduced-order quadratic-bilinear systems in two numerical examples. The experiments were run on compute nodes of the `TinkerCliffs` high-performance computing cluster of the Advanced Research Computing (ARC) facility at Virginia Tech using 16 processing cores of the AMD EPYC 7702 200W CPU at 2.0 GHz and 32 GB main memory. We used MATLAB 9.13.0.2126072 (R2022b) running on Red Hat Enterprise Linux release 7.9 (Maipo). The source code, data and results of the numerical experiments are open source/open access and available at [40].

5.1 Experimental setup

In both numerical examples, we compute reduced-order models via structure-preserving interpolation of the symmetric subsystem transfer functions and the generalized transfer functions, denoted by `SymInt` and `GenInt`, respectively. We

compute models using either (i) only the construction of the basis matrix V and a one-sided projection by setting $W = V$, which we abbreviate further on by `V`, or (ii) by also constructing the left basis matrix W for a two-sided projection following the results in Theorem 1 and Corollary 1, which we abbreviate by `VW`. For simplicity, the interpolation points are chosen logarithmically equidistant on the imaginary axis in all cases. If we compute only the basis matrices for interpolation without additional information, this is denoted by `equi`. On the other hand, if we oversampled the frequency range of interest and compressed the resulting basis to a prescribed dimension, e.g., using pivoted QR, this is denoted by `avg`; cf. [39, Rem. 3.3]. In all cases, we focus on the interpolation of either (i) the first two symmetric subsystem transfer functions or (ii) the first two levels of the generalized transfer functions and the third level generalized transfer function containing the quadratic term. The following overview summarizes the considered interpolation methods:

`SymInt(V, equi)`	is the interpolation of symmetric subsystem transfer functions via one-sided projection by constructing the basis matrix V.
`SymInt(VW, equi)`	is the interpolation of symmetric subsystem transfer functions via two-sided projection. Additional interpolation points are selected for the construction of W to match the dimension of V.
`SymInt(V, avg)`	is the approximation of an interpolation basis for symmetric subsystem transfer functions using only samples for the construction of V and one-sided projection.
`SymInt(VW, avg)`	is the approximation of left and right interpolation bases for symmetric subsystem transfer functions using samples for the construction of V and W and two-sided projection. Additional interpolation points are selected for the construction of W to match the computational work to the construction of V.
`GenInt(V, equi)`	is the interpolation of generalized transfer functions via one-sided projection by constructing the basis matrix V. Samples from the second and third level transfer functions are taken alternating.
`GenInt(VW, equi)`	is the interpolation of generalized transfer functions via two-sided projection. For the construction of V, samples from the second and third level transfer functions are taken alternating. Additional interpolation points are selected for the construction of W to match the dimension of V.
`GenInt(V, avg)`	is the approximation of an interpolation basis for generalized transfer functions using only samples for the construction of V and one-sided projection. At all interpolation points, second and third level samples are taken.
`SymInt(VW, avg)`	is the approximation of left and right interpolation bases for generalized transfer functions using samples for the construction of V and W and two-sided projection. For the

construction of V, second and third level samples are taken at all interpolation points. Additional interpolation points are selected for the construction of W to equalize the computational work to the construction of V.

As an additional comparison, we have computed reduced-order models via proper orthogonal decomposition (POD). All POD models have been trained via simulations of the unit step response to remove the correlation of the training and test input signals. For a fair comparison, the trajectory lengths used for POD are chosen with respect to comparable amounts of computational work to the interpolation methods. We consider therefore:

`POD`, which has a computational workload similar to `SymInt(V, equi)` and `GenInt(V, equi)`, and

`POD(avg)`, which uses similar to `SymInt(V, avg)` and `GenInt(V, avg)` an oversampling and computes the orthogonal basis via truncated singular value decomposition.

For the comparison of the reduced-order models in time domain, we simulate the models over finite time intervals using the implicit-explicit (IMEX) Euler scheme and input signals taken from a Gaussian process $\mathrm{GP}(\mu, \mathsf{K})$, with constant mean $\mu \in \mathbb{R}$ and the squared exponential kernel

$$\mathsf{K}(x, y) = \exp\left(-\frac{|x-y|^2}{2\varsigma^2}\right),$$

where $\varsigma \geq 0$ is a smoothing parameter. The parameters μ and ς are chosen independently for the two examples and are given below. For visualization, we compute and plot the maximum pointwise relative errors

$$\mathrm{relerr}(t) := \max_j \left|\frac{y_j(t) - \hat{y}_j(t)}{y_j(t)}\right|.$$

Also, we compute discretized approximations of the relative L_2 and L_∞ errors via

$$\mathrm{relerr}_{L_2} := \frac{\|\mathrm{vec}(y_\mathrm{h} - \hat{y}_\mathrm{h})\|_2}{\|\mathrm{vec}(y_\mathrm{h})\|_2} \quad \text{and} \quad \mathrm{relerr}_{L_\infty} := \frac{\|\mathrm{vec}(y_\mathrm{h} - \hat{y}_\mathrm{h})\|_\infty}{\|\mathrm{vec}(y_\mathrm{h})\|_\infty},$$

where $y_\mathrm{h}, \hat{y}_\mathrm{h} \in \mathbb{R}^{p \times n_h}$ are the discretized output signals of the original and reduced-order model, respectively, in the time interval $[0, t_\mathrm{f}]$, and vec(.) is the vectorization operator.

In frequency domain, we consider the pointwise relative spectral norm errors defined as

$$\mathrm{relerr}(\omega) := \frac{\|G_{\mathrm{sym},1}(\imath\omega) - \widehat{G}_{\mathrm{sym},1}(\imath\omega)\|_2}{\|G_{\mathrm{sym},1}(\imath\omega)\|_2} \quad \text{and}$$

$$\mathrm{relerr}(\omega_1, \omega_2) := \frac{\|G_{\mathrm{sym},2}(\imath\omega_1, \imath\omega_2) - \widehat{G}_{\mathrm{sym},2}(\imath\omega_1, \imath\omega_2)\|_2}{\|G_{\mathrm{sym},2}(\imath\omega_1, \imath\omega_2)\|_2},$$

over the limited frequency intervals $\omega, \omega_1, \omega_2 \in [\omega_{\min}, \omega_{\max}]$. Additionally, we compute approximations to the relative $\mathcal{L}_\infty$-norm errors for the first and second symmetric subsystem transfer functions via

$$\text{relerr}^{(1)}_{\mathcal{L}_\infty} := \frac{\max\limits_{\omega} \|G_{\text{sym},1}(\imath\omega) - \widehat{G}_{\text{sym},1}(\imath\omega)\|_2}{\max\limits_{\omega} \|G_{\text{sym},1}(\imath\omega)\|_2} \qquad \text{and}$$

$$\text{relerr}^{(2)}_{\mathcal{L}_\infty} := \frac{\max\limits_{\omega_1,\,\omega_2} \|G_{\text{sym},2}(\imath\omega_1, \imath\omega_2) - \widehat{G}_{\text{sym},2}(\imath\omega_1, \imath\omega_2)\|_2}{\max\limits_{\omega_1,\,\omega_2} \|G_{\text{sym},2}(\imath\omega_1, \imath\omega_2)\|_2},$$

using 500 logarithmically equidistant sampling points in the frequency interval of interest $[\omega_{\min}, \omega_{\max}]$. For further details on the experimental setup, we refer the reader to the accompanying code package [40].

5.2 Quadratic-bilinear time-delayed reaction-diffusion model

As first example, we consider the time-delayed heated rod with bilinear feedback from [11, 17], to which we append a quadratic reaction term to obtain

$$\begin{aligned}\partial_t v(\zeta, t) = \Delta v(\zeta, t) - 2\sin(\zeta)v(\zeta, t) + 2\sin(\zeta)v(\zeta, t-\tau) \\ - 2\sin(\zeta)v(\zeta, t)^2 + \sum_{j=1}^{m} b_j(\zeta)(v(\zeta, t) + 1)u_j(t),\end{aligned}$$

with $(t, \zeta) \in (0, t_f) \times (0, \pi)$, boundary conditions $v(t, 0) = v(t, \pi) = 0$ for all $t \in [0, t_f]$, and the constant time delay $\tau = 1$. After spatial discretization with central finite differences, we obtain a quadratic-bilinear time-delay system of the form

$$\begin{aligned}E\dot{x}(t) &= Ax(t) + A_d x(t-\tau) + H(x(t) \otimes x(t)) + \sum_{j=1}^{m} N_k x(t)u_j(t) + Bu(t), \\ y(t) &= Cx(t),\end{aligned}$$

with $E, A, A_d, N_j \in \mathbb{R}^{n\times n}$, for $j = 1, \ldots, m$, $H \in \mathbb{R}^{n\times n^2}$, $B \in \mathbb{R}^{n\times m}$ and $C \in \mathbb{R}^{p\times n}$. For our experiments, we have chosen $n = 2\,000$, $m = 2$ and $p = 2$, where u_1 controls the temperature of the first third of the rod and u_2 the rest, and y_1 observes the temperature of the first half of the rod and y_2 of the second half. The system has zero initial conditions $x(t) = 0$ for all $t \le 0$.

The reduced-order models are computed as explained in Section 5.1. We have computed reduced-order models of order $r = 12$ as well as $r = 24$ such that we can investigate the error behavior as the reduced order increases. The computation times for the construction of the reduced-order models and for the time simulations using the full-order model or the reduced-order models are reported in Table 1. We can see that the construction of the reduced-order models without oversampling takes only a

Table 1 Computation times for the time-delay example in seconds: for both sizes of reduced-order models, we can observe that the models generated without oversampling can be constructed in a fraction of the computation time needed for even a single full-order time simulation. Those models constructed via oversampling are around 1 to 1.25 times as expensive as a single time simulation such that the construction of these models immediately pays off when more than one or longer time simulations are needed

FOM simulation		28.4927	
		$r = 12$	$r = 24$
ROM construction	`SymInt(V, equi)`	2.9955	10.2805
	`SymInt(V, avg)`	26.9295	33.3453
	`SymInt(VW, equi)`	2.8717	10.1273
	`SymInt(VW, avg)`	28.3259	33.7421
	`GenInt(V, equi)`	2.6775	9.7244
	`GenInt(V, avg)`	28.2524	35.5631
	`GenInt(VW, equi)`	2.7534	9.7989
	`GenInt(VW, avg)`	31.6192	35.9506
	`POD`	2.6153	9.8389
	`POD(avg)`	22.7521	31.5421
ROM simulations		0.0502	0.1054

fraction of the computation time of a single simulation of the full-order model such that the construction of the reduced-order models immediately pays off. For the models constructed from oversampled bases, the construction of the reduced-order models pays off when more than a single time simulation is needed. In Tables 2 and 3, we can see that all reduced-order models perform well. However, the interpolation-based models provide smaller errors than those generated by POD for both chosen reduced

Table 2 Errors computed as shown in Section 5.1 for the $r = 12$ time-delay example: `POD` computes the worst performing reduced-order model, while `POD(avg)` is at least one order of magnitude worse in error than all the interpolation-based models. Here, all interpolation-based methods perform comparably well

	relerr_{L_2}	relerr_{L_∞}	$\text{relerr}^{(1)}_{\mathcal{L}_\infty}$	$\text{relerr}^{(2)}_{\mathcal{L}_\infty}$
`SymInt(V, equi)`	2.7360e−06	1.1822e−05	2.2183e−05	2.7574e−05
`SymInt(V, avg)`	5.9357e−05	1.4497e−04	1.1200e−05	2.0584e−05
`SymInt(VW, equi)`	2.2676e−06	6.2992e−06	2.1285e−05	2.9037e−05
`SymInt(VW, avg)`	2.9262e−05	8.4803e−05	3.3123e−06	7.4927e−06
`GenInt(V, equi)`	2.0735e−06	6.3304e−06	1.0049e−05	2.0322e−06
`GenInt(V, avg)`	1.3227e−05	2.9398e−05	1.0850e−05	1.1291e−05
`GenInt(VW, equi)`	1.7310e−05	7.0273e−05	2.6421e−04	2.8641e−05
`GenInt(VW, avg)`	3.8794e−05	1.3094e−04	3.5969e−06	7.2201e−06
`POD`	1.3310e−01	2.1106e−01	1.6514e−01	8.5100e−02
`POD(avg)`	4.9905e−04	1.0262e−03	3.8593e−04	8.9527e−04

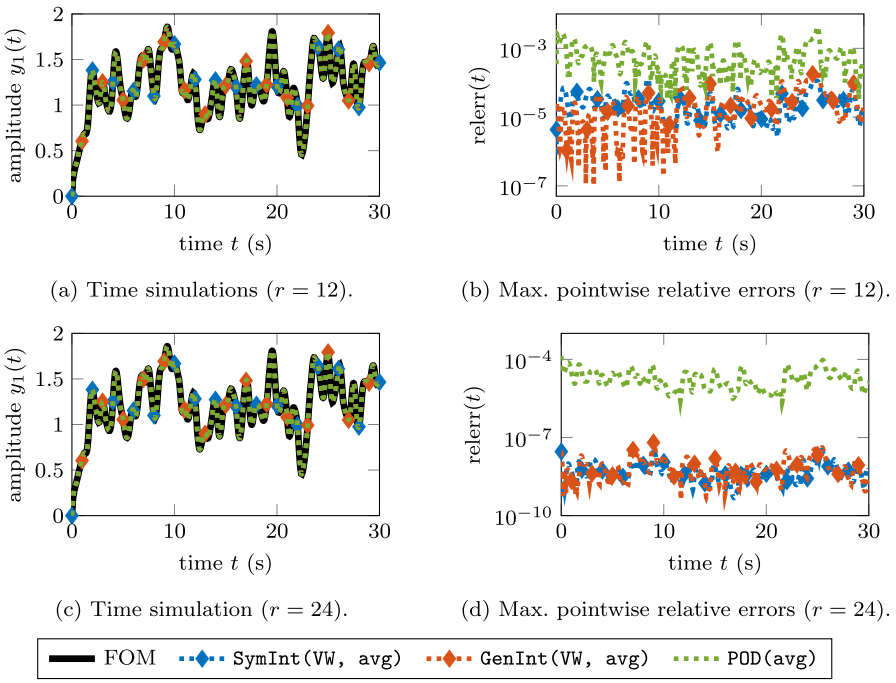
Table 3 Errors computed as shown in Section 5.1 for the $r = 24$ time-delay example: `POD` computes the worst performing reduced-order model, while `POD(avg)` is only as good as the worst interpolation-based models. Here, `SymInt(VW, avg)` performs best w.r.t. three out of the four error measures, while according to $\text{relerr}^{(1)}_{\mathcal{L}_\infty}$, the best reduced-order model is computed by `GenInt(VW, avg)`

	relerr_{L_2}	relerr_{L_∞}	$\text{relerr}^{(1)}_{\mathcal{L}_\infty}$	$\text{relerr}^{(2)}_{\mathcal{L}_\infty}$
`SymInt(V, equi)`	8.1604e−07	1.8475e−06	2.5851e−06	3.4393e−06
`SymInt(V, avg)`	3.1414e−08	4.9411e−08	3.4598e−08	1.0957e−06
`SymInt(VW, equi)`	1.3522e−05	6.6301e−05	6.2129e−08	6.1247e−06
`SymInt(VW, avg)`	6.1324e−09	2.4509e−08	4.9776e−10	2.5239e−09
`GenInt(V, equi)`	3.2366e−06	4.3858e−06	1.1710e−05	5.3091e−06
`GenInt(V, avg)`	3.9579e−08	6.7805e−08	3.1402e−08	1.0562e−06
`GenInt(VW, equi)`	5.2035e−06	1.7040e−05	4.0332e−07	4.2761e−07
`GenInt(VW, avg)`	1.0280e−08	4.3505e−08	1.1418e−10	4.1712e−09
`POD`	5.1889e−04	1.0145e−03	5.8308e−04	2.6052e−04
`POD(avg)`	1.9453e−05	4.7931e−05	5.4489e−06	4.4316e−05

orders $r = 12$ and $r = 24$, where the better POD model computed by `POD(avg)` performs mildly worse than the worst interpolation-based reduced-order models. Comparing the different interpolation approaches we cannot determine a superior choice of transfer function as in most cases symmetric as well as generalized transfer functions provide errors in the same order of magnitude. For the reduced-order models constructed via exact interpolation (`equi`), the sampling of higher order terms in the generalized transfer function needed to be restricted to match the reduced basis dimension. This restriction is removed in `avg` such that more information about the bilinear and quadratic terms can be obtained in sampling the generalized transfer functions compared to the symmetric transfer function setting. However, this does not seem to provide a significantly better approximation than using the symmetric transfer functions.

Figure 1 shows the time simulation of the full-order model and the best performing reduced-order models from each method for the two reduced orders $r = 12$ and $r = 24$ over the time interval $[0, 30]$ s. We restricted Fig. 1a and c to only the first output signal for clarity, but the pointwise relative errors in Fig. 1b and d are computed over both output entries. For the input signals, we have chosen the mean $\mu = 2$ and the parameter $\varsigma = 0.25$. The interpolation-based methods clearly outperform `POD(avg)` for both reduced orders. For the smaller order $r = 12$, the interpolation-based methods are around one order of magnitude better than `POD(avg)`, while for $r = 24$, this improves to around four orders of magnitude better approximation errors by the interpolation-based methods. There is no significant difference between the errors of `SymInt(VW, avg)` and `GenInt(VW, avg)`.

Similar results can be observed in frequency domain. The first symmetric subsystem transfer functions are shown in Fig. 2, while Fig. 3 illustrates the pointwise relative errors of the second symmetric subsystem transfer functions. In both cases, we only show the best performing methods in the frequency interval $\omega, \omega_1, \omega_2 \in [10^{-3}, 10^3]$ rad/s. As in the time domain, the interpolation-based methods outperform



(a) Time simulations ($r = 12$). (b) Max. pointwise relative errors ($r = 12$).

(c) Time simulation ($r = 24$). (d) Max. pointwise relative errors ($r = 24$).

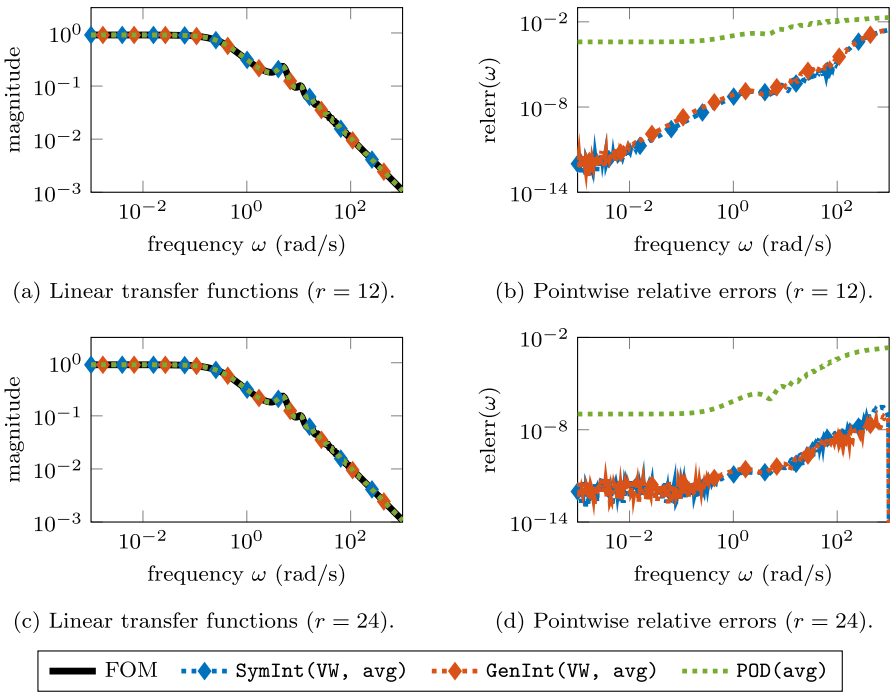
Fig. 1 Time simulations of the time-delay example: the best reduced-order models from each generating approach are shown. All reduced-order models can recover the system behavior for the given input signal, but the interpolation-based reduced-order models perform around four orders of magnitude better in terms of accuracy than the model generated by `POD(avg)`

`POD(avg)` by several orders of magnitude in terms of accuracy. Further plots of the other methods in time and frequency domain can be found in the accompanying code package [40].

5.3 Particle motion in one-dimensional crystal structures

As second example, we consider the motion of particles in a one-dimensional crystal structure described by the Toda lattice model [37]; see Fig. 4. This model belongs to the class of nonlinear mechanical systems of the form (1). The nonlinear time evolution function contains exponential terms that describe the forces between the different particles:

$$f\big(q(t), \dot{q}(t), u(t)\big) = -\widetilde{D}\dot{q}(t) - \begin{bmatrix} e^{k_1(q_1(t)-q_2(t))} - 1 \\ e^{k_2(q_2(t)-q_3(t))} - e^{k_1(q_1(t)-q_2(t))} \\ \vdots \\ e^{k_\ell q_\ell(t)} - e^{k_{\ell-1}(q_{\ell-1}(t)-q_\ell(t))} \end{bmatrix}, \tag{33}$$



(a) Linear transfer functions ($r = 12$).

(b) Pointwise relative errors ($r = 12$).

(c) Linear transfer functions ($r = 24$).

(d) Pointwise relative errors ($r = 24$).

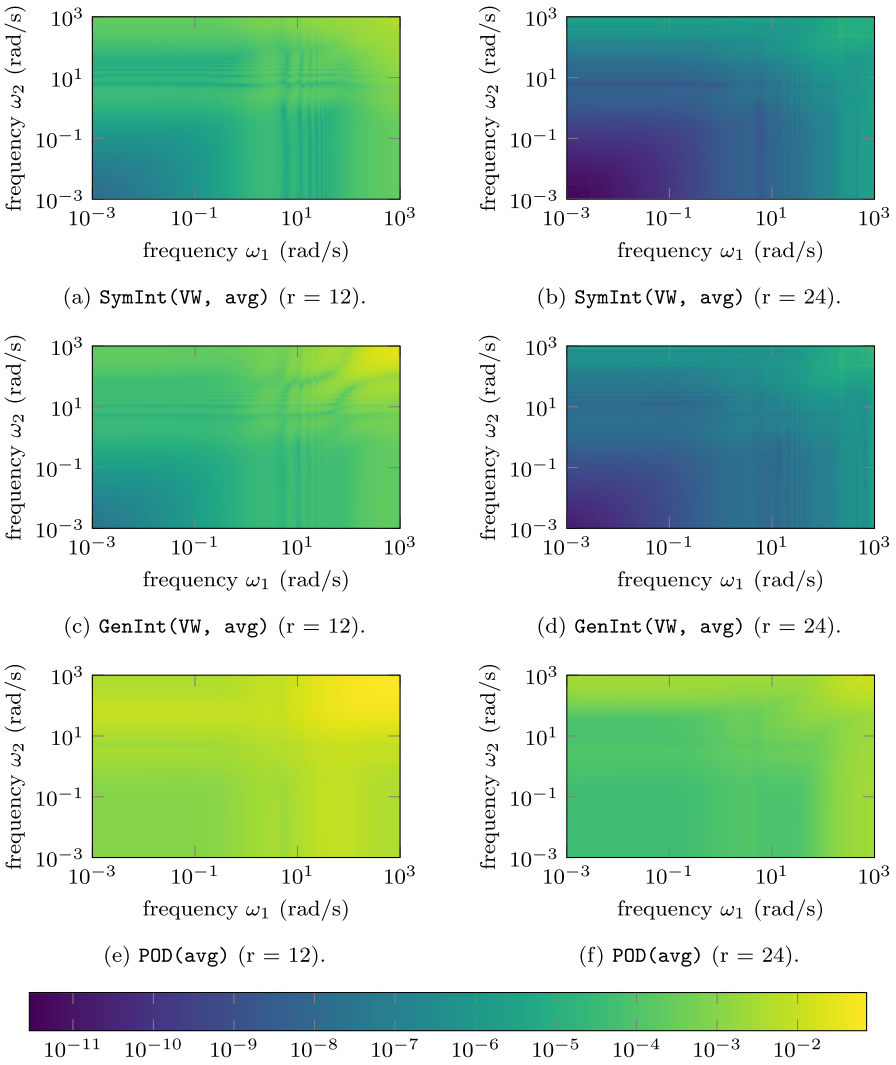
Fig. 2 Sigma plots showing $\|G(\imath\omega))\|_2$ of the first symmetric subsystem transfer function of the time-delay example: the best reduced-order models from each generating approach are shown. All reduced-order models can recover the system behavior for the given input signal, but the interpolation-based reduced-order models perform around six orders of magnitude better in terms of accuracy than the model generated by `POD(avg)`

with the positive semidefinite diagonal damping matrix $\widetilde{D} \in \mathbb{R}^{\ell\times\ell}$ and the positive stiffness coefficients $k_1, \ldots, k_\ell$. See [39, Sec. 1.3.3] for the derivation of the differential model from the underlying Hamiltonian. The original system with exponential nonlinearities (33) can be rewritten into quadratic-bilinear form by introducing the auxiliary variables

$$z_j(t) = \begin{cases} e^{k_j(q_j(t)-q_{j+1}(t))} - 1 & \text{for } j < \ell, \\ e^{k_\ell q_\ell(t)} - 1 & \text{for } j = \ell. \end{cases} \tag{34}$$

By differentiating (34) twice, the Toda lattice model can be written as a system of quadratic-bilinear ordinary differential equations of the form

$$\begin{aligned} 0 &= M\ddot{q}(t) + D\dot{q}(t) + Kq(t) + H_{\mathrm{vv}}\big(\dot{q}(t)\otimes\dot{q}(t)\big) + H_{\mathrm{pv}}\big(q(t)\otimes\dot{q}(t)\big) \\ &\quad + H_{\mathrm{pp}}\big(q(t)\otimes q(t)\big) - N_{\mathrm{p}}q(t)u(t) - B_{\mathrm{u}}u(t), \\ y(t) &= C_{\mathrm{v}}\dot{q}(t), \end{aligned} \tag{35}$$



(a) `SymInt(VW, avg)` (r = 12). (b) `SymInt(VW, avg)` (r = 24).

(c) `GenInt(VW, avg)` (r = 12). (d) `GenInt(VW, avg)` (r = 24).

(e) `POD(avg)` (r = 12). (f) `POD(avg)` (r = 24).

Fig. 3 Second symmetric subsystem transfer function relative approximation errors relerr(ω_1, ω_2) of the time-delay example: the best reduced-order models from each generating approach are shown. The errors of both interpolation-based reduced-order models are at least four orders of magnitude better than those of the `POD(avg)` model

Fig. 4 Schematic illustration of the Toda lattice with ℓ particles [39]. Atoms in a one-dimensional crystal structure are represented as point masses and connected by exponential springs modeling the forces between the particles

with the dimensions as in (2) and $m = 1$ input and $p = 1$ output. The exact parameterization of the matrices in (35) can be found in [39, Sec. 6.5]. For our experiments, we use the same setup as in [39, Sec. 6.5] with $\ell = 2\,000$ particles such that (35) has the order $n = 4\,000$.

It has been observed in [39] that the internal block structures of the matrices in (35) resulting from the original and auxiliary variables should be preserved for stability of the reduced-order models. Therefore, we follow the suggestion in [39, Sec. 6.5] and use the *split congruence transformation* approach [15, 33, 38]. That is, given a basis matrix $V = \begin{bmatrix} V_1^{\mathsf{H}} & V_2^{\mathsf{H}} \end{bmatrix}^{\mathsf{H}} \in \mathbb{C}^{2\ell \times r}$, we construct the extended basis matrix

$$\widetilde{V} = \begin{bmatrix} V_1 & 0 \\ 0 & V_2 \end{bmatrix} \in \mathbb{C}^{2\ell \times 2r},$$

and similarly for a left basis matrix W. The extended basis matrices are then used for model reduction by projection. We apply this approach in our experiments to modify all projection basis matrices computed as described in Section 5.1. By construction, it holds that

$$\operatorname{span}(V) \subseteq \operatorname{span}(\widetilde{V}).$$

Therefore, if V was constructed to satisfy any subspace conditions in Section 4 for interpolation, the basis matrix $\widetilde{V}$ also satisfies these conditions such that interpolation properties are preserved.

For the comparison in our experiments, we have chosen the reduced order of all computed models to be $2r = 120$. The computation times for the construction of the reduced-order models, the time simulation of the full-order model and the time simulations of the reduced-order models are shown in Table 4. The simulation of the reduced-order models takes only a fraction of the time of the full-order simulation; however, the construction of the reduced-order models takes approximately 10 times as long as one time simulation of the full-order model. Therefore, the construction of the reduced-order models pays off if more than 10 time simulations (or a simulation around 10 times as long as shown here) are needed. The approximation errors are shown in Table 5. The best performing model in terms of time simulation error is `SymInt(V, equi)` followed by its counterpart for the generalized transfer functions `GenInt(V, equi)`. None of the models resulting from a two-sided projection has a stable time-domain simulation, which appears to be a consequence of loosing additional mechanical properties by $V \neq W$. The numerical integration of the POD generated models appears to be unstable in all situations. Here, the large frequency domain errors indicate that the approximation quality is not sufficient to approximate the system behavior well enough. In frequency domain, we observe similarly to the previous numerical example that `SymInt(VW, avg)` and `GenInt(VW, avg)` perform best in terms of approximating the first symmetric subsystem transfer function.

The time-domain simulations of the full- and the best performing reduced-order models from each method are shown in Fig. 5 in the time interval $[0, 100]$ s. For the input signal, we have chosen the mean $\mu = 0$ and the smoothing parameter $\varsigma = 2$. `POD(avg)` performs visibly stable only until around 60 s, while the other two models

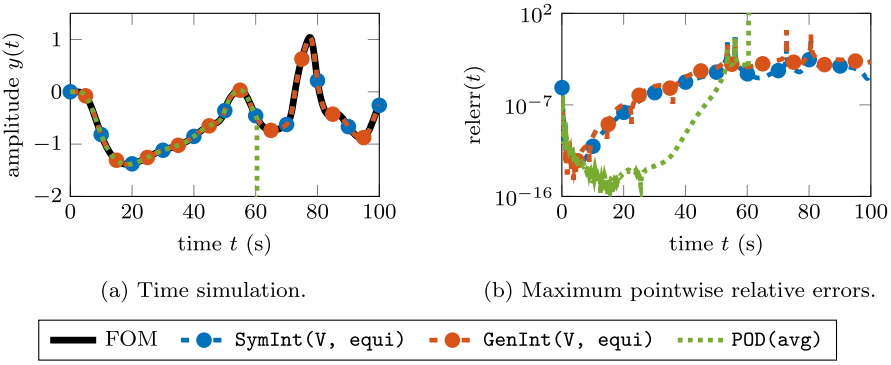
Table 4 Computation times for the Toda lattice example in seconds: the construction of any reduced-order model is around 10 times as expensive as the time simulation of the Toda lattice model. In other words, the reduced-order models pay off in terms of computation time if more than 10 time simulations or a simulation with an at least 10 times longer time horizon are needed. Note that the computation times of the reduced-order model construction is strongly dominated by the projection of the three quadratic tensors of the model such that improvements on computing the tensor projections quickly result in faster computation times

FOM simulation		204.4126
ROM construction	`SymInt(V, equi)`	2046.0838
	`SymInt(V, avg)`	2079.8549
	`SymInt(VW, equi)`	2012.4560
	`SymInt(VW, avg)`	2037.2552
	`GenInt(V, equi)`	2072.3530
	`GenInt(V, avg)`	2205.1623
	`GenInt(VW, equi)`	1973.3049
	`GenInt(VW, avg)`	2041.2817
	`POD`	2006.0416
	`POD(avg)`	2043.2449
ROM simulations		12.2813

follow the system behavior over the complete time interval. The pointwise relative errors in Fig. 5b reveal `POD(avg)` to be more accurate than `SymInt(V, equi)` and `GenInt(V, equi)` for the first half of the time interval before it assumes the same error level as the other two methods and finally becomes unstable. `SymInt(V, equi)` and `GenInt(V, equi)` have overall a similar error behavior, with the errors of `GenInt(V, equi)` being mildly larger.

Table 5 Error table of the Toda lattice example: The errors are computed as shown in Section 5.1. Only the interpolation methods with one-sided projections provide reduced-order models that are stable in the time simulation. The models with ∞ error are unstable. `SymInt(V, equi)` performs overall best with one order of magnitude smaller time domain errors in comparison to `GenInt(V, equi)`

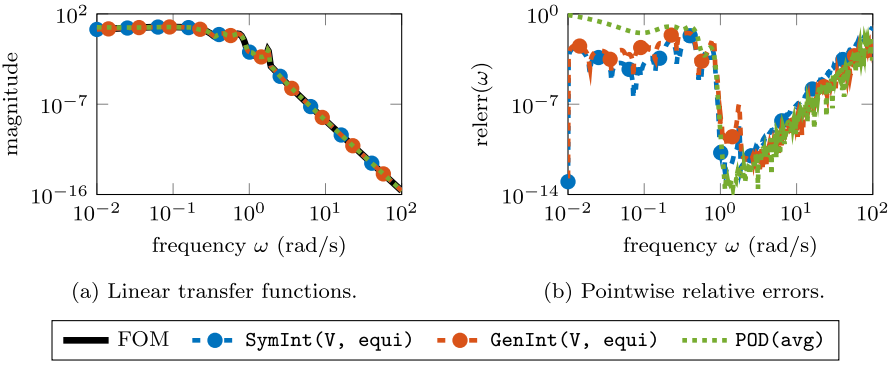
	relerr_{L_2}	relerr_{L_∞}	$\text{relerr}^{(1)}_{\mathcal{L}_\infty}$	$\text{relerr}^{(2)}_{\mathcal{L}_\infty}$
`SymInt(V, equi)`	2.2445e−04	4.1691e−04	5.3358e−03	4.3541e−02
`SymInt(V, avg)`	1.4565e−02	1.7264e−02	1.4862e−02	9.4341e−02
`SymInt(VW, equi)`	∞	∞	4.4450e−03	5.2675e−02
`SymInt(VW, avg)`	∞	∞	2.3354e−05	2.3035e−01
`GenInt(V, equi)`	1.1059e−03	2.1774e−03	1.3737e−02	7.6595e−02
`GenInt(V, avg)`	4.2052e−03	8.5704e−03	1.8517e−03	2.6190e−02
`GenInt(VW, equi)`	∞	∞	1.6930e−03	7.9712e−02
`GenInt(VW, avg)`	∞	∞	8.2451e−06	4.5499e−02
`POD`	∞	∞	8.1769e−01	7.7696e−01
`POD(avg)`	∞	∞	4.6483e−01	6.2382e−01



(a) Time simulation. (b) Maximum pointwise relative errors.

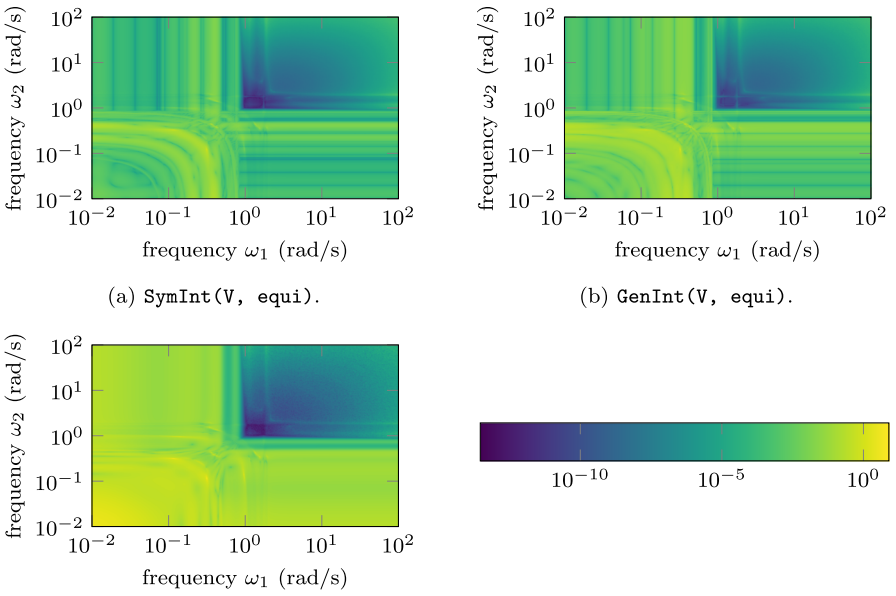
Fig. 5 Time simulation of the Toda lattice example: the best reduced-order models from each generating approach are shown. Only the interpolation-based reduced-order models recover the system behavior over the full time interval, while `POD(avg)` becomes unstable after about 60 s

On the other hand, in frequency domain, we can observe a similar behavior compared to the previous numerical example. The results for the same reduced-order models that performed best in time domain can be seen in Figs. 6 and 7, with the frequency interval of interest $\omega, \omega_1, \omega_2 \in [10^{-3}, 10^3]$ rad/s. The POD generated models perform worst, with the exception of `GenInt(VW, equi)`. The models based on oversampling generalized transfer functions perform better than those based on oversampling the symmetric transfer functions due to the additional information obtained from the nonlinear terms. However, in this example we can observe that the oversampling procedure may produce larger errors than exact interpolation. In particular, models computed via `avg` provide worse approximation errors for larger frequencies, where the transfer functions are converging to zero, while the models with `equi` pre-



(a) Linear transfer functions. (b) Pointwise relative errors.

Fig. 6 First symmetric transfer function of the Toda lattice example: only the best performing methods from the time simulation are shown. All reduced-order models recover the transfer function behavior of the original system. For low frequencies, `POD(avg)` performs two order of magnitude worse than the interpolating methods



(a) `SymInt(V, equi)`. (b) `GenInt(V, equi)`.

(c) `POD(avg)`.

Fig. 7 Second symmetric transfer function relative approximation errors relerr(ω_1, ω_2) of the Toda lattice example: only the best performing methods from the time simulation are shown. For large frequencies, `SymInt(V, equi)` and `POD(avg)` perform equally well and better than `GenInt(V, equi)`, while for low frequencies, `POD(avg)` is an order of magnitude worse than the interpolating methods

serve the system behavior due to the exact interpolation. Further plots of the other methods in time and frequency domain can be found in the accompanying code package [40].

6 Conclusions

We have extended the structure-preserving interpolation framework to quadratic-bilinear systems. Based on two motivating structured examples, we have introduced the structured variants of the symmetric subsystem and generalized transfer functions of quadratic-bilinear systems. For both transfer function types, we provided subspace conditions enabling the computation of interpolating structured reduced-order models by projection. The theoretical findings are then used to compute structured reduced-order models in two numerical examples. The theory presented here can be applied to a much broader class of structures than those used here for illustrations.

The numerical results suggest that the interpolation of symmetric transfer functions provides more accurate reduced-order models than the generalized transfer functions for the same reduced order. This is most certainly a consequence of the restriction to only products of system terms in the generalized transfer functions. However, we have seen that when the basis matrices are first constructed by oversampling and then compressed, the generalized transfer function interpolation framework provides more

accurate reduced-order models for the same computational costs as for the symmetric transfer functions due to additional information obtained from the nonlinear terms. The authors of [18] extend on this oversampling idea and the definition of structured generalized transfer functions for quadratic-bilinear systems from this paper to propose structure-preserving model reduction for systems with polynomial nonlinearities based on the interpolation of generalized transfer functions with at most one nonlinear component. While this gives a first efficient approach to simulation-free model reduction for polynomial systems, there are many open questions left. One related to our observations in this work is the question whether exact interpolation of transfer functions based on Volterra kernels may perform better than an oversampling procedure. This needs a thorough investigation of interpolation conditions for polynomial systems, which we will address in some future work.

Another transfer function type for quadratic-bilinear systems are regular subsystem transfer functions. These have been omitted in this paper since for the choice of identical interpolation points, the projection spaces of regular and symmetric subsystem transfer functions coincide. The formulas for structured variants of regular transfer functions and results on interpolation conditions will be presented in a separate work.

For simplicity of exposition, we have restricted the numerical experiments to only logarithmically equidistant interpolation points on the imaginary axis. While such a procedure is often sufficient in practice, the question of good or even optimal interpolation points remains open and crucial for the success of such model reduction methods. This is still an unresolved issue even in the case of structured linear systems and needs further investigation in the future.

Funding Open Access funding enabled and organized by Projekt DEAL. Parts of this work were carried out while Werner was with the Max Planck Institute for Dynamics of Complex Technical Systems in Magdeburg, Germany, and with the Courant Institute of Mathematical Sciences, New York University, USA. Benner and Werner have been supported by the German Research Foundation (DFG) Research Training Group 2297 "Mathematical Complexity Reduction (MathCoRe)". The work of Gugercin is based upon work supported by the National Science Foundation under Grant No. DMS–1819110.

Data Availability The data used in the numerical experiments and the generated results are open access and available at [40].

Code Availability The source code of the numerical experiments is open source and available at [40].

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Ahmad, M.I., Benner, P., Jaimoukha, I.: Krylov subspace methods for model reduction of quadratic-bilinear systems. IET Control Theory Appl. **10**(16), 2010–2018 (2016). https://doi.org/10.1049/iet-cta.2016.0415
2. Antoulas, A.C., Beattie, C.A., Gugercin, S.: Interpolatory Methods for Model Reduction. Computational Science & Engineering. SIAM, Philadelphia, PA (2020). https://doi.org/10.1137/1.9781611976083
3. Bai, Z., Skoogh, D.: A projection method for model reduction of bilinear dynamical systems. Linear Algebra Appl. **415**(2–3), 406–425 (2006). https://doi.org/10.1016/j.laa.2005.04.032
4. Barrault, M., Maday, Y., Nguyen, N.C., Patera, A.T.: An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations. C. R. Math. Acad. Sci. Paris **339**(9), 667–672 (2004). https://doi.org/10.1016/j.crma.2004.08.006
5. Beattie, C.A., Gugercin, S.: Interpolatory projection methods for structure-preserving model reduction. Syst. Control Lett. **58**(3), 225–232 (2009). https://doi.org/10.1016/j.sysconle.2008.10.016
6. Benner, P., Breiten, T.: Two-sided projection methods for nonlinear model order reduction. SIAM J. Sci. Comput. **37**(2), B239–B260 (2015). https://doi.org/10.1137/14097255X
7. Benner, P., Goyal, P.: Balanced truncation model order reduction for quadratic-bilinear control systems. Optimization and Control (math.OC) e-print 1705.00160, arXiv (2017). https://doi.org/10.48550/arXiv.1705.00160
8. Benner, P., Goyal, P.: Interpolation-based model order reduction for polynomial systems. SIAM J. Sci. Comput. **43**(1), A84–A108 (2021). https://doi.org/10.1137/19M1259171
9. Benner, P., Goyal, P., Gugercin, S.: $\mathcal{H}_2$-quasi-optimal model order reduction for quadratic-bilinear control systems. SIAM J. Matrix Anal. Appl. **39**(2), 983–1032 (2018). https://doi.org/10.1137/16M1098280
10. Benner, P., Gugercin, S., Werner, S.W.R.: Structure-preserving interpolation of bilinear control systems. Adv. Comput. Math. **47**(3), 43 (2021). https://doi.org/10.1007/s10444-021-09863-w
11. Benner, P., Gugercin, S., Werner, S.W.R.: A unifying framework for tangential interpolation of structured bilinear control systems. Numer. Math. **155**(3–4), 445–483 (2023). https://doi.org/10.1007/s00211-023-01380-w
12. Chaturantabut, S., Sorensen, D.C.: Nonlinear model reduction via discrete empirical interpolation. SIAM J. Sci. Comput. **32**(5), 2737–2764 (2010). https://doi.org/10.1137/090766498
13. Drmač, Z., Gugercin, S.: A new selection operator for the discrete empirical interpolation method–Improved a priori error bound and extensions. SIAM J. Sci. Comput. **38**(2), A631–A648 (2016). https://doi.org/10.1137/15M1019271
14. Erneux, T.: Applied Delay Differential Equations, Surveys and Tutorials in the Applied Mathematical Sciences, vol. 3. Springer, New York (2009). https://doi.org/10.1007/978-0-387-74372-1
15. Freund, R.W.: Padé-type model reduction of second-order and higher-order linear dynamical systems. In: P. Benner, V. Mehrmann, D.C. Sorensen (eds.) Dimension Reduction of Large-Scale Systems, Lect. Notes Comput. Sci. Eng., vol. 45, pp. 173–189. Springer, Berlin, Heidelberg (2005). https://doi.org/10.1007/3-540-27909-1_8
16. Gosea, I.V., Antoulas, A.C.: Data-driven model order reduction of quadratic-bilinear systems. Numer. Linear Algebra Appl. **25**(6), e2200 (2018). https://doi.org/10.1002/nla.2200
17. Gosea, I.V., Pontes Duff, I., Benner, P., Antoulas, A.C.: Model order reduction of bilinear time-delay systems. In: Proc. of 18th European Control Conference (ECC), pp. 2289–2294 (2019). https://doi.org/10.23919/ECC.2019.8796085
18. Goyal, P., Pontes Duff, I., Benner, P.: Dominant subspaces of high-fidelity nonlinear structured parametric dynamical systems and model reduction. e-print 2301.09484, arXiv (2023). https://doi.org/10.48550/arXiv.2301.09484. Dynamical Systems (math.DS)
19. Gu, C.: 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**(9), 1307–1320 (2011). https://doi.org/10.1109/TCAD.2011.2142184
20. Gurevich, S.V.: Dynamics of localized structures in reaction-diffusion systems induced by delayed feedback. Phys. Rev. E **87**, 052922 (2013). https://doi.org/10.1103/PhysRevE.87.052922
21. Himpe, C., Ohlberger, M.: A unified software framework for empirical Gramians. J. Math. **2013**, 365909 (2013). https://doi.org/10.1155/2013/365909

22. Hinze, M., Volkwein, S.: Proper orthogonal decomposition surrogate models for nonlinear dynamical systems: error estimates and suboptimal control. In: Benner, P., Mehrmann, V., Sorensen, D.C. (eds.) Dimension Reduction of Large-Scale Systems, Lect. Notes Comput. Sci. Eng., vol. 45, pp. 261–306. Springer, Berlin, Heidelberg (2005). https://doi.org/10.1007/3-540-27909-1_10
23. Johnson, M.A., Moon, F.C.: Experimental characterization of quasiperiodicity and chaos in a mechanical system with delay. Int. J. Bifurc. Chaos Appl. Sci. Eng. **9**(1), 49–65 (1999). https://doi.org/10.1142/S0218127499000031
24. Kramer, B., Gugercin, S., Borggaard, J.: Nonlinear balanced truncation: Part 1 – Computing energy functions. e-print 2209.07645, arXiv (2022). https://doi.org/10.48550/arXiv.2209.07645. Optimization and Control (math.OC)
25. Kramer, B., Willcox, K.E.: Nonlinear model order reduction via lifting transformations and proper orthogonal decomposition. AIAA J. **57**(6), 2297–2307 (2019). https://doi.org/10.2514/1.J057791
26. Kunisch, K., Volkwein, S.: Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics. SIAM J. Numer. Anal. **40**(2), 492–515 (2002). https://doi.org/10.1137/S0036142900382612
27. Lall, S., Marsden, J.E., Glavaški, S.: Empirical model reduction of controlled nonlinear systems. IFAC Proceedings Volumes (14th IFAC World Congress) **32**(2), 2598–2603 (1999). https://doi.org/10.1016/S1474-6670(17)56442-3
28. Lawrence, D., Myatt, J.H., Camphouse, R.C.: On model reduction via empirical balanced truncation. In: Proc. Am. Control Conf., vol. 5, pp. 3139–3144 (2005). https://doi.org/10.1109/ACC.2005.1470454
29. McCormick, G.P.: Computability of global solutions to factorable nonconvex programs: Part I - Convex underestimating problems. Math. Program. **10**(1), 147–175 (1976). https://doi.org/10.1007/BF01580665
30. Peherstorfer, B.: Model reduction for transport-dominated problems via online adaptive bases and adaptive sampling. SIAM J. Sci. Comput. **42**(5), A2803–A2836 (2020). https://doi.org/10.1137/19M1257275
31. Peherstorfer, B., Willcox, K.: Data-driven operator inference for nonintrusive projection-based model reduction. Comput. Methods Appl. Mech. Eng. **306**, 196–215 (2016). https://doi.org/10.1016/j.cma.2016.03.025
32. Qian, E., Kramer, B., Peherstorfer, B., Willcox, K.: Lift & Learn: physics-informed machine learning for large-scale nonlinear dynamical systems. Phys. D: Nonlinear Phenom. **406**, 132401 (2020). https://doi.org/10.1016/j.physd.2020.132401
33. Ritschel, T.K.S., Weiß, F., Baumann, M., Grundel, S.: Nonlinear model reduction of dynamical power grid models using quadratization and balanced truncation. at-Automatisierungstechnik **68**(12), 1022–1034 (2020). https://doi.org/10.1515/auto-2020-0070
34. Rugh, W.J.: Nonlinear System Theory: The Volterra/Wiener Approach. The Johns Hopkins University Press, Baltimore (1981)
35. Saboureau, P., Foing, J.P., Schanne, P.: Injection-locked semiconductor lasers with delayed optoelectronic feedback. IEEE J. Quantum Electron. **33**(9), 1582–1591 (1997). https://doi.org/10.1109/3.622640
36. Scherpen, J.M.A.: Balancing for nonlinear systems. Syst. Control Lett. **21**(2), 143–153 (1993). https://doi.org/10.1016/0167-6911(93)90117-O
37. Toda, M.: Vibration of a chain with nonlinear interaction. J. Phys. Soc. Jpn. **22**(2), 431–436 (1967). https://doi.org/10.1143/JPSJ.22.431
38. Van de Walle, A., Naets, F., Deckers, E., Desmet, W.: Stability-preserving model order reduction for time-domain simulation of vibro-acoustic FE models. Int. J. Numer. Methods Eng. **109**(6), 889–912 (2017). https://doi.org/10.1002/nme.5323
39. Werner, S.W.R.: Structure-preserving model reduction for mechanical systems. Dissertation, Otto-von-Guericke-Universität, Magdeburg, Germany (2021). https://doi.org/10.25673/38617
40. Werner, S.W.R.: Code, data and results for numerical experiments in "Structured interpolation for multivariate transfer functions of quadratic-bilinear systems" (version 2.0) (2023). https://doi.org/10.5281/zenodo.10116600
41. Willcox, K., Peraire, J.: Balanced model reduction via the proper orthogonal decomposition. AIAA J. **40**(11), 2323–2330 (2002). https://doi.org/10.2514/2.1570