

INTERPOLATION-BASED MODEL ORDER REDUCTION FOR POLYNOMIAL SYSTEMS*

PETER BENNER[†] AND PAWAN GOYAL[‡]

Abstract. In this work, we investigate a model-order reduction scheme for polynomial systems. We begin with defining the generalized multivariate transfer functions for the system. Based on this, we aim at constructing a reduced-order system, interpolating the defined generalized transfer functions at a given set of interpolation points. Furthermore, we provide a method, inspired by the Loewner approach for linear and (quadratic-)bilinear systems, to determine a good-quality reduced-order system in an automatic way. We also discuss the computational issues related to the proposed method and a potential application of a CUR matrix approximation in order to further speed up simulation of the reduced-order systems. We test the efficiency of the proposed method via two benchmark examples.

Key words. model order reduction, polynomial dynamical systems, transfer functions, interpolation, tensor algebra, matricization

AMS subject classifications. 15A69, 34C20, 41A05, 49M05, 93A15, 93C10, 93C15

DOI. 10.1137/19M1259171

1. Introduction. Model-order reduction (MOR) is a technique that enables the construction of low-dimensional or reduced-order systems of large-scale dynamical systems. This is achieved by projecting the high-dimensional state vector into a low-dimensional subspace. The obtained reduced-order systems are computationally cheaper and efficient, yet still accurate. Hence, we can replace large-scale dynamical systems by reduced-order systems while performing engineering studies such as optimization, control design, and uncertainty quantification. We refer to [12, 16, 44] for overviews of popular MOR techniques. In this paper, we focus on a class of nonlinear systems, namely polynomial systems of the form

$$(1.1) \quad \begin{aligned} \mathbf{E}\dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \sum_{\xi=2}^d \mathbf{H}_{\xi} \mathbf{x}^{\otimes \xi}(t) + \sum_{\eta=1}^d \mathbf{N}_{\eta}(\mathbf{u}(t) \otimes \mathbf{x}^{\otimes \eta}(t)) + \mathbf{B}\mathbf{u}(t), \quad \mathbf{x}(0) = \mathbf{0}, \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t), \end{aligned}$$

where $\mathbf{E}, \mathbf{A} \in \mathbb{R}^{n \times n}$, $\mathbf{B} \in \mathbb{R}^{n \times m}$, $\mathbf{C} \in \mathbb{R}^{q \times n}$, $\mathbf{H}_{\xi} \in \mathbb{R}^{n \times n^{\xi}}$, $\xi \in \{2, \dots, d\}$, $\mathbf{N}_{\eta} \in \mathbb{R}^{n \times m \cdot n^{\eta}}$, $\eta \in \{1, \dots, d\}$; the state, input and output vectors are denoted by $\mathbf{x}(t) \in \mathbb{R}^n$, $\mathbf{u}(t) \in \mathbb{R}^m$ and $\mathbf{y}(t) \in \mathbb{R}^q$, respectively; $\mathbf{x}^{\otimes \xi} := \mathbf{x} \otimes \dots \otimes \mathbf{x}$, e.g., $\mathbf{x}^{\otimes 3} = \mathbf{x} \otimes \mathbf{x} \otimes \mathbf{x}$. Furthermore, we assume that the matrix pencil (\mathbf{A}, \mathbf{E}) is stable and all the eigenvalues of the pencil are finite. The system (1.1) has polynomial terms of degree up to d ; thus, we refer to it as a *dth degree polynomial system*. The trajectories of the system (1.1) lie in the n -dimensional Euclidean space and we are interested in the situations in which n is large, say $n \in \mathcal{O}(10^5 - 10^6)$, so that a significant amount of computer time will

*Submitted to the journal's Methods and Algorithms for Scientific Computing section April 30, 2019; accepted for publication (in revised form) September 23, 2020; published electronically January 5, 2021.

<https://doi.org/10.1137/19M1259171>

[†]Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstraße 1, 39106 Magdeburg, Germany (benner@mpi-magdeburg.mpg.de) and Otto von Guericke University Magdeburg, Faculty of Mathematics, Magdeburg, Germany.

[‡]Corresponding author. Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstraße 1, 39106 Magdeburg, Germany (goyalp@mpi-magdeburg.mpg.de).

be needed to simulate (1.1) for varying input signals, like in a control, design, or optimization context. We seek to construct a reduced-order system of order $r \ll n$ that not only preserves the polynomial structure but also the output of it is as close as possible to the output of the system (1.1) in an appropriate norm.

Most of the widely used MOR techniques to construct reduced-order systems for nonlinear systems are based on *snapshots*. This means that the state vector $\mathbf{x}(t)$ needs to be evaluated for a given input. In this category, proper orthogonal decomposition is arguably the most favored method. The method relies on determining the dominant subspace for the state vectors through singular value decomposition (SVD) of the collected snapshots, which is generally followed by computing a reduced-order system via Galerkin projection. For more details, we refer to [29]. For nonlinear systems, it is often combined with hyperreduction methods to further reduce computational costs related to the reduced nonlinear terms; see, e.g., [10, 11, 22, 21, 41, 42]. Another widely known method in this category is the *trajectory piecewise linear* method, in which a nonlinear system is approximated by a weighted sum of linearized systems (linearized along the trajectory). Then, each linear system is reduced using popular methods for linear systems such as balanced truncation or iterative methods; see, e.g., [5, 16, 20, 30]. Moreover, reduced basis methods, which are also snapshot-based, have been successfully applied to several nonlinear systems; see, e.g., [43]. Although these methods have been very successful, they share a common drawback of being dependent on snapshots. In other words, the resulting reduced-order systems depend on the trajectories given for predefined inputs. As a result, it is difficult to use these reduced-order systems, e.g., in feedback-control applications, where input variations are not a priori known.

In this work, we focus on MOR methods that allow us to determine reduced-order systems without any prior knowledge of inputs. There are basically two types of such methods, namely, interpolation-based and balancing-related approaches; see [12] for this categorization. Recently, there have been significant efforts to extend these methods from linear to special classes of polynomial systems, namely, bilinear and quadratic-bilinear systems; see, e.g., [7, 14, 15, 18, 19, 25, 27, 34]. In this paper, we investigate an interpolation-based MOR scheme to obtain a reduced-order system for the system (1.1). For this, we first define generalized transfer functions for the system (1.1). Based on this, we aim at constructing a reduced-order system such that its generalized transfer functions interpolate those of the original system at a given set of interpolation points. Furthermore, we propose a scheme, inspired by the Loewner approach for linear and (quadratic-)bilinear systems [7, 27], thus leading to an algorithm that allows us to construct a good quality reduced-order system in an automatic way. Furthermore, we discuss related computational aspects and an application of a pseudoskeletal matrix approximation, the so-called CUR factorization, to further reduce the computational complexity related to the reduced nonlinear terms.

The remaining structure of the paper is as follows. In the following section, we briefly recap interpolatory MOR for linear systems, polynomialization of nonlinear systems, and some basic concepts from tensor algebra. In section 3, we present the generalized transfer functions corresponding to (1.1) and discuss the construction of an interpolating reduced-order system using Petrov–Galerkin projection. Based on this, we propose an algorithm that allows us to determine a good quality reduced-order system in an automatic way. In section 4, we discuss the related computational aspects and investigate an application of CUR matrix approximation to further reduce the complexity of the reduced nonlinear terms. In section 5, we illustrate the efficiency

of the proposed algorithms by means of two benchmark problems. We conclude the paper with a summary of our contributions and future perspectives.

We make use of the following notation in the paper:

- `orth()` returns an orthonormal basis of a given set of vectors.
- The Hadamard and Kronecker products are denoted by “ \circ ” and “ \otimes ,” respectively.
- Using MATLAB notation, $\mathbf{A}(:, 1:r)$ denotes the first r columns of the matrix \mathbf{A} , and $\mathbf{A}(i, j)$ is the (i, j) th element of the matrix \mathbf{A} .
- \mathbf{I}_m is the identity matrix of size $m \times m$.
- \mathcal{V}^{\otimes} is a shorthand notation for

$$\underbrace{\mathcal{V} \otimes \cdots \otimes \mathcal{V}}_{\xi\text{-times}},$$

where \mathcal{V} is a vector/matrix.

2. Background work. In this section, we briefly recap the basic concepts that build a foundation for the rest of the paper. In subsection 2.1, we present an interpolatory MOR scheme for linear systems that we will later on extend to polynomial systems. In subsection 2.2, we discuss the polynomialization of nonlinear systems. In subsection 2.3, we outline a few important concepts from tensor algebra.

2.1. Interpolatory MOR for linear systems. We consider linear time-invariant (LTI) systems of the form

$$(2.1) \quad \begin{aligned} \mathbf{E}\dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \quad \mathbf{x}(0) = 0, \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t), \end{aligned}$$

where $\mathbf{x}(t) \in \mathbb{R}^n$, $\mathbf{u}(t) \in \mathbb{R}^m$, and $\mathbf{y}(t) \in \mathbb{R}^q$, and the systems matrices are of appropriate sizes. Furthermore, we assume that the matrix \mathbf{E} is invertible. Moreover, the system (2.1) is referred to as a single-input single-output (SISO) system when $q = m = 1$; otherwise, it is referred to as a multiinput multioutput system (MIMO). It is known that the analytical solution of the system (2.1) can be given as

$$(2.2) \quad \mathbf{y}(t) = \int_0^t \mathbf{C}e^{\mathbf{E}^{-1}\mathbf{A}\sigma}\mathbf{E}^{-1}\mathbf{B}\mathbf{u}(t - \sigma)d\sigma,$$

where $\mathbf{h}(\sigma) := \mathbf{C}e^{\mathbf{E}^{-1}\mathbf{A}\sigma}\mathbf{E}^{-1}\mathbf{B}$, that can also be seen as the convolution kernel, mapping the input to the output. The Laplace transform of the kernel yields

$$(2.3) \quad \mathbf{H}(s) := \mathbf{C}(s\mathbf{E} - \mathbf{A})^{-1}\mathbf{B},$$

where $s = 2\pi j f$ in which $j = \sqrt{-1}$ and f is the frequency. The function $\mathbf{H}(s)$ is also known as the transfer function of the system (2.1). For a large-scale LTI system, we aim at constructing a reduced-order system of order r :

$$(2.4) \quad \begin{aligned} \widehat{\mathbf{E}}\dot{\widehat{\mathbf{x}}}(t) &= \widehat{\mathbf{A}}\widehat{\mathbf{x}}(t) + \widehat{\mathbf{B}}\mathbf{u}(t), \quad \widehat{\mathbf{x}}(0) = 0, \\ \widehat{\mathbf{y}}(t) &= \widehat{\mathbf{C}}\widehat{\mathbf{x}}(t), \end{aligned}$$

where $\widehat{\mathbf{x}}(t) \in \mathbb{R}^r$, $\widehat{\mathbf{y}}(t) \in \mathbb{R}^q$, and all other matrices are of the appropriate sizes. Let us denote the transfer function of (2.4) by $\widehat{\mathbf{H}}(s)$. We seek to construct a reduced-order system such that its transfer function interpolates that of the original one at predefined frequency points. This problem is very well studied in the literature; see, e.g., [6, 9, 24, 28, 48]. We briefly outline one main result in the following theorem.

THEOREM 2.1. *Consider a SISO LTI system (2.1) of order n . Let $\sigma_i, \mu_i, i \in \{1, \dots, r\}$ be interpolation points such that $s\mathbf{E} - \mathbf{A}$ is invertible at $s \in \{\sigma_i, \mu_i\}, i \in \{1, \dots, r\}$. Furthermore, let the projection matrices \mathbf{V} and \mathbf{W} be as follows:*

$$(2.5a) \quad \mathbf{V} = \text{range} \left((\sigma_1\mathbf{E} - \mathbf{A})^{-1}\mathbf{B}, \dots, (\sigma_r\mathbf{E} - \mathbf{A})^{-1}\mathbf{B} \right),$$

$$(2.5b) \quad \mathbf{W} = \text{range} \left((\mu_1\mathbf{E} - \mathbf{A})^{-T}\mathbf{C}^T, \dots, (\mu_r\mathbf{E} - \mathbf{A})^{-T}\mathbf{C}^T \right).$$

Assume that \mathbf{V} and \mathbf{W} are full column rank matrices. If a reduced-order system of order r is constructed as

$$\hat{\mathbf{E}} = \mathbf{W}^T\mathbf{E}\mathbf{V}, \quad \hat{\mathbf{A}} = \mathbf{W}^T\mathbf{A}\mathbf{V}, \quad \hat{\mathbf{B}} = \mathbf{W}^T\mathbf{B}, \quad \hat{\mathbf{C}} = \mathbf{C}\mathbf{V},$$

then

$$(2.6) \quad \mathbf{H}(s) = \hat{\mathbf{H}}(s), \quad s \in \{\sigma_i, \mu_i\}, i \in \{1, \dots, r\},$$

provided $\sigma_i, \mu_i, i \in \{1, \dots, r\}$ are not in the spectrum of the matrix pencil $(\hat{\mathbf{A}}, \hat{\mathbf{E}})$. Moreover, if $\sigma_i = \mu_i$, then the following are also fulfilled:

$$(2.7) \quad \frac{d}{ds}\mathbf{H}(s) = \frac{d}{ds}\hat{\mathbf{H}}(s), \quad s \in \{\sigma_1, \dots, \sigma_r\}.$$

In the above theorem, we have discussed interpolation of SISO systems. In the MIMO case, the idea of tangential interpolation is often employed; see, e.g., [26, 40]. In this case, we seek to determine a reduced-order system that tangentially interpolates the original system, i.e.,

$$(2.8a) \quad \mathbf{H}(\sigma_i)\tilde{\mathbf{b}}_i = \hat{\mathbf{H}}(\sigma_i)\tilde{\mathbf{b}}_i, \quad i \in \{1, \dots, r\},$$

$$(2.8b) \quad \tilde{\mathbf{c}}_i^T\mathbf{H}(\mu_i) = \tilde{\mathbf{c}}_i^T\hat{\mathbf{H}}(\mu_i), \quad i \in \{1, \dots, r\},$$

where $\tilde{\mathbf{b}}_i$ and $\tilde{\mathbf{c}}_i$ are appropriately chosen directions in \mathbb{R}^m and \mathbb{R}^q , respectively. The goal (2.8) can be achieved when we determine the projection matrices \mathbf{V} and \mathbf{W} as follows:

$$(2.9a) \quad \mathbf{V} = \text{range} \left((\sigma_1\mathbf{E} - \mathbf{A})^{-1}\mathbf{B}\tilde{\mathbf{b}}_1, \dots, (\sigma_r\mathbf{E} - \mathbf{A})^{-1}\mathbf{B}\tilde{\mathbf{b}}_r \right),$$

$$(2.9b) \quad \mathbf{W} = \text{range} \left((\mu_1\mathbf{E} - \mathbf{A})^{-T}\mathbf{C}^T\tilde{\mathbf{c}}_1, \dots, (\mu_r\mathbf{E} - \mathbf{A})^{-T}\mathbf{C}^T\tilde{\mathbf{c}}_r \right).$$

Next, the choice of interpolation points plays an important role in ensuring a good quality of the reduced-order systems. There has been intensive research in this direction; see, e.g., [30, 47, 50]. Moreover, it has been shown in [4, 13] that if we consider enough interpolation points and construct the matrices \mathbf{V} and \mathbf{W} as in (2.9), then the matrices $\mathbf{W}^T\mathbf{E}\mathbf{V}$ and $\mathbf{W}^T\mathbf{A}\mathbf{V}$ encode the complexity of the linear system. In other words, the rank of the matrix $[\mathbf{W}^T\mathbf{E}\mathbf{V}, \mathbf{W}^T\mathbf{A}\mathbf{V}]$ indicates the minimal order of an underlying linear system that exactly represents the dynamics. Moreover, it also provides subspaces that are the most important for the input-output mapping of the linear systems. We refer to [4, 13] for a detailed discussion on this aspect.

2.2. Polynomialization of nonlinear systems. A class of nonlinear systems, containing nonlinear terms such as exponential, trigonometric, or rational functions,

can be rewritten as a polynomial system (1.1) by introducing some auxiliary variables. This process is very closely related to the McCormick relaxation used in non-convex optimization [39]. Recently, due to advances in the methodologies for MOR for quadratic-bilinear (QB) systems, there has been a substantial focus on rewriting a nonlinear system into the QB form. However, in this subsection, we illustrate with an example of how a polynomialization of a nonlinear system is done by introducing fewer auxiliary variables as compared to its quadratic-bilinearization.

An illustrative example. Let us consider the following one-dimensional nonlinear ODE:

$$(2.10a) \quad \dot{x}(t) = -x(t) - x^3(t) \cdot e^{-x(t)} + u(t),$$

$$(2.10b) \quad y(t) = x(t).$$

Now, we seek to rewrite the system (2.10) as a polynomial system via polynomialization. For this, we introduce an auxiliary variable $z(t) := e^{-x(t)}$ and derive the corresponding differential equation. That is

$$\dot{z}(t) = -e^{-x(t)}\dot{x}(t) = -z(t) \left(-x(t) - x^3(t)z(t) + u(t) \right).$$

Thus, we can equivalently write the input-output system (2.10) as follows:

$$\begin{bmatrix} \dot{x}(t) \\ \dot{z}(t) \end{bmatrix} = \begin{bmatrix} -x(t) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ x(t)z(t) \end{bmatrix} - \begin{bmatrix} x^3(t)z(t) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ x^3(t)z^2(t) \end{bmatrix} - \begin{bmatrix} 0 \\ z(t) \end{bmatrix} u(t) + \begin{bmatrix} u(t) \\ 0 \end{bmatrix},$$

$$y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ z(t) \end{bmatrix}.$$

The nonlinear system (2.10) can be written into a polynomial system (1.1) of degree 5 by introducing a single variable. Furthermore, if one aims at rewriting the system into the QB form, then we need to introduce at least 3 more auxiliary variables. Thus, the nonlinear system (2.10) of order 1 can be written into the QB form of order 5.

We believe that it is not straightforward to conclude whether working with low-order systems of a higher polynomial degree is easier than higher-order systems of a lower polynomial degree. However, we have experienced in our numerical simulation that introducing more auxiliary variables leads to numerical difficulties, e.g., the resulting system becomes stiffer as we increase the number of auxiliary variables. Thus, the goal of the work is to establish a MOR technique to construct reduced-order systems for polynomial systems of any degree that preserve the polynomial structure. Moreover, there are several applications in science and engineering where governing equations inherently have polynomial terms and by the goal of our method, we would be able to keep the same polynomial structure in reduced-order systems.

2.3. Tensor algebra. In this subsection, we recall important results from tensor algebra. Tensor algebra presents an effective tool to simplify algebra involving Kronecker (tensor) products. A tensor is a multidimensional or an N -way array. An N th-order tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times \dots \times n_N}$ is an N -dimensional array with entries $x_{i_1, \dots, i_N} \in \mathbb{R}$, where $i_j \in \{1, \dots, n_j\}$, $j \in \{1, \dots, N\}$. An important tool in tensor calculus is *matricization*. This allows us to unfold a tensor into a matrix, which plays a crucial role in tensor computations. For an N th-order tensor, there are N different ways to unfold as a matrix. In the following, we define mode- m matricization of a tensor \mathcal{X} .

DEFINITION 2.2 (e.g., [33]). *The mode- m matricization of a tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times \dots \times n_N}$, denoted by $\mathbf{X}_{(m)}$, satisfies the following mapping:*

$$\mathbf{X}_{(m)}(i_m, j) = \mathcal{X}_{i_1, \dots, i_N},$$

where $j = 1 + \sum_{k=1, k \neq m}^N (i_k - 1)J_k$ with $J_k = \prod_{z=1, z \neq m}^{k-1} n_z$ and $i_m \in \{1, \dots, n_m\}$.

Like matrix-vector and matrix-matrix products, tensor-tensor, tensor-matrix, and tensor-vector products can be defined. Of a particular interest of the paper, we discuss the m -mode (matrix) product of an N th-order tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times \dots \times n_N}$ and a matrix $\mathbf{U} \in \mathbb{R}^{J \times n_m}$. Such a product, denoted by $\mathcal{X} \times_m \mathbf{U}$, yields also an N th-order tensor, denoted by \mathcal{Y} , of size $n_1 \times \dots \times n_{m-1} \times J \times n_{m+1} \times \dots \times n_N$. Following [33], the elements of the tensor \mathcal{Y} can be given as

$$(2.11) \quad \mathcal{Y}_{i_1 \dots i_{m-1} j i_{m+1} \dots i_N} = \sum_{i_m=1}^{n_m} x_{i_1 i_2 \dots i_N} \mathbf{U}_{j i_m}.$$

The tensor \mathcal{Y} can be interpreted in terms of the product of an unfolded tensor and a matrix. The tensor \mathcal{Y} is such that its mode- m is given by $\mathbf{Y}_{(m)} = \mathbf{U} \mathbf{X}_{(m)}$. Generalizing this, let us now consider the following tensor-matrix product:

$$\mathcal{Y} = \mathcal{X} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \dots \times_N \mathbf{A}^{(N)},$$

where $\mathbf{A}^{(l)} \in \mathbb{R}^{J_l \times n_l}$ and $\mathcal{Y} \in \mathbb{R}^{J_1 \times \dots \times J_N}$. Then, we have the following relations between unfolded tensors and Kronecker products:

$$(2.12) \quad \mathbf{Y}_{(m)} = \mathbf{A}^{(m)} X_{(m)} \left(\mathbf{A}^{(N)} \otimes \dots \otimes \mathbf{A}^{(m+1)} \otimes \mathbf{A}^{(m-1)} \otimes \mathbf{A}^{(1)} \right)^T, \quad m \in \{1, \dots, N\}.$$

Remark 2.3. If $\mathbf{A}^{(k)}, k \in \{1, \dots, n_N\}$ are columns vectors, then the product

$$\mathcal{X} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \dots \times_N \mathbf{A}^{(N)}$$

would be just a scalar. Consequently, we also have

$$\begin{aligned} & \left(\mathbf{A}^{(1)} \right)^T \mathbf{X}_{(1)} \left(\mathbf{A}^{(N)} \otimes \dots \otimes \mathbf{A}^{(2)} \right)^T \\ &= \left(\mathbf{A}^{(2)} \right)^T \mathbf{X}_{(2)} \left(\mathbf{A}^{(N)} \otimes \dots \otimes \mathbf{A}^{(3)} \otimes \mathbf{A}^{(1)} \right)^T = \dots \\ &= \left(\mathbf{A}^{(N)} \right)^T \mathbf{X}_{(N)} \left(\mathbf{A}^{(N-1)} \otimes \dots \otimes \mathbf{A}^{(1)} \right)^T. \end{aligned}$$

For further details on these concepts, we strongly recommend [33] to readers.

3. Construction of interpolating reduced-order systems. In this section, we present the construction of interpolating reduced-order systems for the systems (1.1). We begin with defining the generalized transfer functions and discuss construction of interpolatory reduced-order systems.

3.1. Generalized transfer functions and reduced-order modeling. Let us begin by considering SISO polynomial systems (1.1) and, without loss of generality, assume $\mathbf{E} = \mathbf{I}_n$. As a first step towards developing a MOR scheme for the system, we

aim at defining the generalized multivariate transfer functions. Following the steps as shown in [19] for QB systems, we write the Volterra series corresponding to the system (1.1) as follows:

$$(3.1) \quad \mathbf{x}(t) = \int_0^t e^{\mathbf{A}\sigma_1} \mathbf{B}u(t_{\sigma_1}) d\sigma_1 + \sum_{\xi=2}^d \int_0^t e^{\mathbf{A}\sigma_1} \mathbf{H}_\xi \mathbf{x}^{\otimes \xi}(t_{\sigma_1}) d\sigma_1 \\ + \sum_{\eta=1}^d e^{\mathbf{A}\sigma_1} \int_0^t \mathbf{N}_\eta u(t_{\sigma_1}) \mathbf{x}^{\otimes \eta}(t_{\sigma_1}) d\sigma_1,$$

where $t_{\sigma_1} := t - \sigma_1$. Using (3.1), we obtain the expression for $\mathbf{x}(t_{\sigma_1})$ as

$$(3.2) \quad \mathbf{x}(t_{\sigma_1}) = \int_0^{t_{\sigma_1}} e^{\mathbf{A}\sigma_2} \mathbf{B}u(t_{\sigma_1}-\sigma_2) d\sigma_2 + \sum_{\xi=2}^d \int_0^{t_{\sigma_1}} e^{\mathbf{A}\sigma_2} \mathbf{H}_\xi \mathbf{x}^{\otimes \xi}(t_{\sigma_1}-\sigma_2) d\sigma_2 \\ + \sum_{\eta=1}^d \int_0^{t_{\sigma_1}} e^{\mathbf{A}\sigma_2} \mathbf{N}_\eta u(t_{\sigma_1}-\sigma_2) \mathbf{x}^{\otimes \eta}(t_{\sigma_1}-\sigma_2) d\sigma_2.$$

Substituting the expression in (3.2) for $\mathbf{x}(t_{\sigma_1})$ in (3.1) and multiplying by \mathbf{C} yields

$$\mathbf{y}(t) = \int_0^t \mathbf{C}e^{\mathbf{A}\sigma_1} \mathbf{B}u(t_{\sigma_1}) d\sigma_1 \\ + \sum_{\xi=2}^d \int_0^t \underbrace{\int_0^{t_{\sigma_1}} \dots \int_0^{t_{\sigma_1}}}_{\xi\text{-times}} \mathbf{C}e^{\mathbf{A}\sigma_1} \mathbf{H}_\xi (e^{\mathbf{A}\sigma_2} \mathbf{B} \otimes \dots \otimes e^{\mathbf{A}\sigma_{\xi+1}} \mathbf{B}) d\sigma_1 d\sigma_2 \dots d\sigma_{\xi+1} \\ + \sum_{\eta=1}^d \int_0^t \underbrace{\int_0^{t_{\sigma_1}} \dots \int_0^{t_{\sigma_1}}}_{\eta\text{-times}} \mathbf{C}e^{\mathbf{A}\sigma_1} \mathbf{N}_\eta (e^{\mathbf{A}\sigma_2} \mathbf{B} \otimes \dots \otimes e^{\mathbf{A}\sigma_{\eta+1}} \mathbf{B}) \\ \times (u(t_{\sigma_1})u(t_{\sigma_1}-\sigma_2) \dots u(t_{\sigma_1}-\sigma_{\eta+1})) d\sigma_1 d\sigma_2 \dots d\sigma_{\eta+1} + \dots.$$

The Volterra series, corresponding to the system (1.1), becomes cumbersome and contains infinitely many terms. Thus, in this paper, we restrict ourselves only to the leading *kernels* of the series, which are as follows:

$$(3.3a) \quad f_L(t_1) := \mathbf{C}e^{\mathbf{A}t_1} \mathbf{B},$$

$$(3.3b) \quad f_H^{(\xi)}(t_1, \dots, t_{\xi+1}) := \mathbf{C}e^{\mathbf{A}t_1} \mathbf{H}_\xi (e^{\mathbf{A}t_2} \mathbf{B} \otimes \dots \otimes e^{\mathbf{A}t_{\xi+1}} \mathbf{B}),$$

$$(3.3c) \quad f_N^{(\eta)}(t_1, \dots, t_{\eta+1}) := \mathbf{C}e^{\mathbf{A}t_1} \mathbf{N}_\eta (e^{\mathbf{A}t_2} \mathbf{B} \otimes \dots \otimes e^{\mathbf{A}t_{\eta+1}} \mathbf{B}),$$

where $\xi \in \{2, \dots, d\}$ and $\eta \in \{1, \dots, d\}$. Furthermore, taking the multivariate Laplace transform (see, e.g., [45]) of the above kernels, we get the frequency-domain representations of the kernels as follows:

$$(3.4a) \quad \mathbf{F}_L(s_1) := \mathcal{L}(f_L) = \mathbf{C}\Phi(s_1)\mathbf{B},$$

$$(3.4b) \quad \mathbf{F}_H^{(\xi)}(s_1, \dots, s_{\xi+1}) := \mathcal{L}(f_H^{(\xi)}) = \mathbf{C}\Phi(s_{\xi+1})\mathbf{H}_\xi (\Phi(s_\xi)\mathbf{B} \otimes \dots \otimes \Phi(s_1)\mathbf{B}),$$

$$(3.4c) \quad \mathbf{F}_N^{(\eta)}(s_1, \dots, s_{\eta+1}) := \mathcal{L}(f_N^{(\eta)}) = \mathbf{C}\Phi(s_{\eta+1})\mathbf{N}_\eta (\Phi(s_\eta)\mathbf{B} \otimes \dots \otimes \Phi(s_1)\mathbf{B}),$$

where $\Phi(s) = (s\mathbf{I}_n - \mathbf{A})^{-1}$ is the so-called state transition matrix, and $\mathcal{L}(\cdot)$ denotes the multivariate Laplace transform.

Thus far, we have assumed that the mass matrix in front of $\dot{\mathbf{x}}(t)$ in (1.1) is $\mathbf{E} = \mathbf{I}_n$; however, one can also perform the above algebra to derive the multivariate transfer function for $\mathbf{E} \neq \mathbf{I}_n$. In this case, we can also obtain the multivariate transfer functions as in (3.4), where the matrix $\Phi(s)$ will be $(s\mathbf{E} - \mathbf{A})^{-1}$ instead of $(s\mathbf{I}_n - \mathbf{A})^{-1}$. In the rest of the paper, we consider the generalized case with \mathbf{E} being any invertible matrix. We aim at constructing reduced-order systems, having a similar structure to (1.1), as follows:

$$(3.5) \quad \begin{aligned} \widehat{\mathbf{E}}\widehat{\mathbf{x}}(t) &= \widehat{\mathbf{A}}\widehat{\mathbf{x}}(t) + \sum_{\xi=2}^d \widehat{\mathbf{H}}_{\xi}\widehat{\mathbf{x}}^{\otimes \xi}(t) + \sum_{\eta=1}^d \widehat{\mathbf{N}}_{\eta}(\mathbf{u}(t) \otimes \widehat{\mathbf{x}}^{\otimes \eta}(t)) + \widehat{\mathbf{B}}\mathbf{u}(t), \quad \widehat{\mathbf{x}}(0) = 0, \\ \widehat{\mathbf{y}}(t) &= \widehat{\mathbf{C}}\widehat{\mathbf{x}}(t), \end{aligned}$$

where $\widehat{\mathbf{x}}(t) \in \mathbb{R}^r$, $\mathbf{u}(t) \in \mathbb{R}$, and $\widehat{\mathbf{y}}(t) \in \mathbb{R}$ are reduced state, input, and output vectors, respectively, with $r \ll n$, and all other matrices are of appropriate sizes. To that end, our goal is to construct reduced-order systems (3.5) using Petrov–Galerkin projection such that the multivariate transfer functions, as given in (3.4), of the original system match with those of the reduced-order system at a given set of interpolation points. For this, we require projection matrices $\mathbf{V} \in \mathbb{R}^{n \times r}$ and $\mathbf{W} \in \mathbb{R}^{n \times r}$, thus leading to the system matrices of (3.5) as follows:

$$(3.6) \quad \begin{aligned} \widehat{\mathbf{E}} &= \mathbf{W}^T \mathbf{A} \mathbf{V}, & \widehat{\mathbf{A}} &= \mathbf{W}^T \mathbf{A} \mathbf{V}, & \widehat{\mathbf{H}}_{\xi} &= \mathbf{W}^T \mathbf{H}_{\xi} \mathbf{V}^{\otimes \xi}, & \xi &\in \{2, \dots, d\}, \\ \widehat{\mathbf{B}} &= \mathbf{W}^T \mathbf{B}, & \widehat{\mathbf{C}} &= \mathbf{C} \mathbf{V}, & \widehat{\mathbf{N}}_{\eta} &= \mathbf{W}^T \mathbf{N}_{\eta} \mathbf{V}^{\otimes \eta}, & \eta &\in \{1, \dots, d\}, \end{aligned}$$

with $\mathbf{x}(t) \approx \mathbf{V}\widehat{\mathbf{x}}(t)$. The choice of the matrices \mathbf{V} and \mathbf{W} must ensure the desired interpolating properties of the original and reduced-order systems, and the quality of the reduced-order system. Thus, in the following theorem, we reveal the construction of the projection matrices \mathbf{V} and \mathbf{W} , yielding an interpolating reduced-order system.

THEOREM 3.1. *Consider a SISO system as given in (1.1). Let σ_i and μ_i , $i \in \{1, \dots, \tilde{r}\}$, be interpolation points such that $(s\mathbf{E} - \mathbf{A})$ is invertible for all $s = \{\sigma_i, \mu_i\}$, $i \in \{1, \dots, \tilde{r}\}$. Moreover, let the projection matrices \mathbf{V} and \mathbf{W} be as follows:*

$$(3.7a) \quad \mathbf{V}_L = \text{range}(\Phi(\sigma_1)\mathbf{B}, \dots, \Phi(\sigma_{\tilde{r}})\mathbf{B}),$$

$$(3.7b) \quad \mathbf{V}_N = \bigcup_{\eta=1}^d \bigcup_{i=1}^{\tilde{r}} \text{range}(\Phi(\sigma_i)\mathbf{N}_{\eta}(\Phi(\sigma_i)\mathbf{B} \otimes \dots \otimes \Phi(\sigma_i)\mathbf{B})),$$

$$(3.7c) \quad \mathbf{V}_H = \bigcup_{\xi=2}^d \bigcup_{i=1}^{\tilde{r}} \text{range}(\Phi(\sigma_i)\mathbf{H}_{\xi}(\Phi(\sigma_i)\mathbf{B} \otimes \dots \otimes \Phi(\sigma_i)\mathbf{B})),$$

$$(3.7d) \quad \mathbf{W}_L = \text{range}(\Phi(\mu_1)^T \mathbf{C}^T, \dots, \Phi(\mu_{\tilde{r}})^T \mathbf{C}^T),$$

$$(3.7e) \quad \begin{aligned} \mathbf{W}_N &= \bigcup_{\eta=1}^d \bigcup_{i=1}^{\tilde{r}} \text{range}(\Phi(\sigma_i)^T (\mathbf{N}_{\eta})_{(2)} (\Phi(\sigma_i)\mathbf{B} \\ &\quad \otimes \dots \otimes \Phi(\sigma_i)\mathbf{B} \otimes \Phi(\mu_i)^T \mathbf{C}^T)), \end{aligned}$$

$$(3.7f) \quad \mathbf{W}_H = \bigcup_{\xi=2}^d \bigcup_{i=1}^{\tilde{r}} \text{range}(\Phi(\sigma_i)^T (\mathbf{H}_{\xi})_{(2)} (\Phi(\sigma_i)\mathbf{B} \otimes \dots \otimes \Phi(\mu_i)^T \mathbf{C}^T)),$$

$$(3.7g) \quad \mathbf{V} = \text{range}(\mathbf{V}_L, \mathbf{V}_N, \mathbf{V}_H),$$

$$(3.7h) \quad \mathbf{W} = \text{range}(\mathbf{W}_L, \mathbf{W}_N, \mathbf{W}_H),$$

where $\Phi(s) := (s\mathbf{E} - \mathbf{A})^{-1}$, and $(\mathbf{H}_\xi)_{(2)} \in \mathbb{R}^{n \times n^\xi}$ and $(\mathbf{N}_\eta)_{(2)} \in \mathbb{R}^{n \times m \cdot n^\eta}$ are, respectively, the mode-2 matricizations of the $(\xi+1)$ -way tensor $\mathcal{H}_\xi \in \mathbb{R}^{n \times \dots \times n}$ and $(\eta+2)$ -way tensor $\mathcal{N}_\eta \in \mathbb{R}^{n \times \dots \times n}$ whose mode-1 matricizations are \mathbf{H}_ξ and \mathbf{N}_η , respectively. Assume $\mathbf{V} \in \mathbb{R}^{n \times r}$ and $\mathbf{W} \in \mathbb{R}^{n \times r}$ are of full column rank, where $r = \tilde{r} + d\tilde{r} + (d-1)\tilde{r}$. If a reduced-order system of order r is computed as shown in (3.6) using the matrices \mathbf{V} and \mathbf{W} , then the reduced-order system satisfies the following interpolation conditions:

$$(3.8a) \quad \mathbf{F}_L(\sigma_i) = \widehat{\mathbf{F}}_L(\sigma_i),$$

$$(3.8b) \quad \mathbf{F}_L(\mu_i) = \widehat{\mathbf{F}}_L(\mu_i),$$

$$(3.8c) \quad \mathbf{F}_N^{(\eta)}(\sigma_i, \dots, \sigma_i) = \widehat{\mathbf{F}}_N^{(\eta)}(\sigma_i, \dots, \sigma_i),$$

$$(3.8d) \quad \mathbf{F}_N^{(\eta)}(\sigma_i, \dots, \sigma_i, \mu_i) = \widehat{\mathbf{F}}_N^{(\eta)}(\sigma_i, \dots, \sigma_i, \mu_i),$$

$$(3.8e) \quad \mathbf{F}_H^{(\xi)}(\sigma_i, \dots, \sigma_i) = \widehat{\mathbf{F}}_H^{(\xi)}(\sigma_i, \dots, \sigma_i),$$

$$(3.8f) \quad \mathbf{F}_H^{(\xi)}(\sigma_i, \dots, \sigma_i, \mu_i) = \widehat{\mathbf{F}}_H^{(\xi)}(\sigma_i, \dots, \sigma_i, \mu_i),$$

provided none of the interpolation points is the eigenvalue of the matrix pencil $(\widehat{\mathbf{A}}, \widehat{\mathbf{E}})$.

Proof. The relations, given in (3.8a) and (3.8b) directly follow from the linear case; see, e.g., [5]. Therefore, we omit their proofs for the sake of brevity of the paper. However, for the rest of the proof, we note down intermediate results from, e.g., [5], which yield, while proving (3.8a) and (3.8b),

$$(3.9a) \quad \mathbf{V}\widehat{\Phi}(\sigma_i)\widehat{\mathbf{B}} = \Phi(\sigma_i)\mathbf{B}, \quad i \in \{1, \dots, \tilde{r}\},$$

$$(3.9b) \quad \mathbf{W}\widehat{\Phi}(\mu_i)^T\widehat{\mathbf{C}} = \Phi(\mu_i)^T\mathbf{C}^T, \quad i \in \{1, \dots, \tilde{r}\},$$

where $\Phi(s) = (s\mathbf{E} - \mathbf{A})^{-1}$ and $\widehat{\Phi}(s) = (s\widehat{\mathbf{E}} - \widehat{\mathbf{A}})^{-1}\widehat{\mathbf{B}}$. Now, we focus on the relation (3.8c). We begin with

$$\begin{aligned} & \mathbf{V}\widehat{\Phi}(\sigma_i)\widehat{\mathbf{N}}_\eta \left(\widehat{\Phi}(\sigma_i)\widehat{\mathbf{B}} \otimes \dots \otimes \widehat{\Phi}(\sigma_i)\widehat{\mathbf{B}} \right) \\ &= \mathbf{V}\widehat{\Phi}(\sigma_i)\mathbf{W}^T\mathbf{N}_\eta\mathbf{V}^\circledast \left(\widehat{\Phi}(\sigma_i)\widehat{\mathbf{B}} \otimes \dots \otimes \widehat{\Phi}(\sigma_i)\widehat{\mathbf{B}} \right) \quad \left(\text{using } \widehat{\mathbf{N}}_\eta = \mathbf{W}^T\mathbf{N}_\eta\mathbf{V}^\circledast \right) \\ &= \mathbf{V}\widehat{\Phi}(\sigma_i)\mathbf{W}^T\mathbf{N}_\eta \left(\mathbf{V}\widehat{\Phi}(\sigma_i)\widehat{\mathbf{B}} \otimes \dots \otimes \mathbf{V}\widehat{\Phi}(\sigma_i)\widehat{\mathbf{B}} \right) \quad \left(\text{using (3.9a)} \right) \\ &= \mathbf{V}\widehat{\Phi}(\sigma_i)\mathbf{W}^T\mathbf{N}_\eta \left(\Phi(\sigma_i)\mathbf{B} \otimes \dots \otimes \Phi(\sigma_i)\mathbf{B} \right) \\ &= \mathbf{V}\widehat{\Phi}(\sigma_i)\mathbf{W}^T\Phi(\sigma_i)^{-1} \underbrace{\Phi(\sigma_i)\mathbf{N}_\eta \left(\Phi(\sigma_i)\mathbf{B} \otimes \dots \otimes \Phi(\sigma_i)\mathbf{B} \right)}_{\in \mathbf{V}} \\ & \quad \left(\text{introduction of } \mathbf{I}_n = \Phi(\sigma_i)^{-1}\Phi(\sigma_i) \right) \\ (3.10) \quad &= \mathbf{V}\widehat{\Phi}(\sigma_i)\mathbf{W}^T\Phi(\sigma_i)^{-1}\mathbf{V}\mathbf{z}, \end{aligned}$$

where the vector \mathbf{z} is such that $\mathbf{V}\mathbf{z} = \Phi(\sigma_i)\mathbf{N}_\eta \left(\Phi(\sigma_i)\mathbf{B} \otimes \dots \otimes \Phi(\sigma_i)\mathbf{B} \right)$. Additionally, we have

$$\begin{aligned} \widehat{\Phi}(s)\mathbf{W}^T\Phi(s)^{-1}\mathbf{V} &= (s\widehat{\mathbf{E}} - \widehat{\mathbf{A}})^{-1}\mathbf{W}^T(s\mathbf{E} - \mathbf{A})\mathbf{V} \\ &= (s\widehat{\mathbf{E}} - \widehat{\mathbf{A}})^{-1}(s\mathbf{W}^T\mathbf{E}\mathbf{V} - \mathbf{W}^T\mathbf{A}\mathbf{V}) = \mathbf{I}_r. \end{aligned}$$

Substituting the above relation into (3.10) and premultiplying with \mathbf{C} yields the relation (3.8c). Similarly, we can prove the relation (3.8e). Next, we focus on the relation (3.8d). We know that

$$\widehat{\mathbf{N}}_\eta = \mathbf{W}^T \mathbf{N}_\eta \mathbf{V}^\otimes.$$

Hence, using (2.12), we obtain

$$(3.11) \quad \left(\widehat{\mathbf{N}}_\eta\right)_{(2)} = \mathbf{V}^T (\mathbf{N}_\eta)_{(2)} \left(\mathbf{V}^{\otimes(\eta-1)} \otimes \mathbf{W}\right),$$

where $(\widehat{\mathbf{N}}_\eta)_{(2)}$ is the mode-2 matricization of the tensor $\widehat{\mathcal{N}}_\eta$ whose mode-1 matricization is $\widehat{\mathbf{N}}_\eta$. With the relation (3.11), we now consider

$$\begin{aligned} & \mathbf{W} \widehat{\Phi}(\sigma_i)^T \left(\widehat{\mathbf{N}}_\eta\right)_{(2)} \left(\widehat{\Phi}(\sigma_i) \widehat{\mathbf{B}} \otimes \cdots \otimes \Phi(\sigma_i) \widehat{\mathbf{B}} \otimes \widehat{\Phi}(\mu_i)^T \widehat{\mathbf{C}}^T\right) \\ &= \mathbf{W} \widehat{\Phi}(\sigma_i)^T \mathbf{V}^T (\mathbf{N}_\eta)_{(2)} \left(\mathbf{V}^{\otimes(\eta-1)} \otimes \mathbf{W}\right) \left(\widehat{\Phi}(\sigma_i) \widehat{\mathbf{B}} \otimes \cdots \otimes \Phi(\sigma_i) \widehat{\mathbf{B}} \otimes \widehat{\Phi}(\mu_i)^T \widehat{\mathbf{C}}^T\right) \\ & \hspace{20em} \text{(using (3.11))} \\ &= \mathbf{W} \widehat{\Phi}(\sigma_i)^T \mathbf{V}^T (\mathbf{N}_\eta)_{(2)} \left(V \widehat{\Phi}(\sigma_i) \widehat{\mathbf{B}} \otimes \cdots \otimes V \widehat{\Phi}(\sigma_i) \widehat{\mathbf{B}} \otimes W \widehat{\Phi}(\mu_i)^T \widehat{\mathbf{C}}^T\right) \\ &= \mathbf{W} \widehat{\Phi}(\sigma_i)^T \mathbf{V}^T (\mathbf{N}_\eta)_{(2)} \left(\Phi(\sigma_i) \mathbf{B} \otimes \cdots \otimes \Phi(\sigma_i) \mathbf{B} \otimes \Phi(\mu_i)^T \mathbf{C}^T\right) \text{ (using (3.9))} \\ &= \mathbf{W} \widehat{\Phi}(\sigma_i)^T \mathbf{V}^T \Phi(\sigma_i)^{-T} \\ & \quad \times \underbrace{\Phi(\sigma_i)^T (\mathbf{N}_\eta)_{(2)} \left(\Phi(\sigma_i) \mathbf{B} \otimes \cdots \otimes \Phi(\sigma_i) \mathbf{B} \otimes \Phi(\mu_i)^T \mathbf{C}^T\right)}_{\in \mathbf{W}(\cdot =: \mathbf{W}\mathbf{q})} \\ &= \mathbf{W} \widehat{\Phi}(\sigma_i)^T \mathbf{V}^T \Phi(\sigma_i)^{-T} \mathbf{W}\mathbf{q} = \mathbf{W}\mathbf{q} \\ &= \Phi(\sigma_i)^T (\mathbf{N}_\eta)_{(2)} \left(\Phi(\sigma_i) \mathbf{B} \otimes \cdots \otimes \Phi(\sigma_i) \mathbf{B} \otimes \Phi(\mu_i)^T \mathbf{C}^T\right). \end{aligned}$$

Next, we multiply both sides by \mathbf{B}^T to get

$$\begin{aligned} & \widehat{\mathbf{B}}^T \widehat{\Phi}(\sigma_i)^T \left(\widehat{\mathbf{N}}_\eta\right)_{(2)} \left(\widehat{\Phi}(\sigma_i) \widehat{\mathbf{B}} \otimes \cdots \otimes \Phi(\sigma_i) \widehat{\mathbf{B}} \otimes \widehat{\Phi}(\mu_i)^T \widehat{\mathbf{C}}^T\right) \\ &= \mathbf{B} \Phi(\sigma_i)^T (\mathbf{N}_\eta)_{(2)} \left(\Phi(\sigma_i) \mathbf{B} \otimes \cdots \otimes \Phi(\sigma_i) \mathbf{B} \otimes \Phi(\mu_i)^T \mathbf{C}^T\right). \end{aligned}$$

Using the matricization property of tensor-vector multiplications (see Remark 2.3), we get

$$\widehat{\mathbf{C}} \widehat{\Phi}(\mu_i) \widehat{\mathbf{N}}_\eta \left(\widehat{\Phi}(\sigma_i) \widehat{\mathbf{B}} \otimes \cdots \otimes \widehat{\Phi}(\sigma_i) \widehat{\mathbf{B}}\right) = \mathbf{C} \Phi(\mu_i) \mathbf{N}_\eta \left(\Phi(\sigma_i) \mathbf{B} \otimes \cdots \otimes \Phi(\sigma_i) \mathbf{B}\right),$$

which is nothing but the relation given in (3.8d). Using similar steps, we can prove (3.8f); thus, for the sake of brevity, we skip it. This concludes the proof. \square

Remark 3.2. Although we have presented the result in Theorem 3.1 for the SISO case, it can be easily extended to the MIMO case with the help of tangential directions as done in the case of linear systems. Precisely, in this case, $\Phi(\sigma_i) \mathbf{B}$ and $\Phi(\mu_i)^T \mathbf{C}^T$ in (3.7) are replaced by $\Phi(\sigma_i) \mathbf{B} \mathbf{b}_i$ and $\Phi(\mu_i)^T \mathbf{C}^T \mathbf{c}_i$, where $\mathbf{b}_i \in \mathbb{C}^m$ and $\mathbf{c}_i \in \mathbb{C}^q$ are the tangential directions corresponding to the interpolation points σ_i and μ_i . As a result, we get a reduced-order system whose generalized transfer functions tangentially interpolate those of the original ones.

Remark 3.3. There might be some applications where the matrix $\mathbf{A} = 0$. Therefore, we mention that Theorem 3.1 is valid even if $\mathbf{A} = 0$, provided the corresponding Volterra series is convergent. If the matrix $\mathbf{A} = 0$, then it can readily be seen that the projection matrices \mathbf{V} and \mathbf{W} are not full column rank. In the following section, we discuss how to handle such cases, in particular how to compress the redundant information and construct a lower-order realization.

3.2. Connection to the Loewner approach. In recent years, Loewner-based MOR has received a lot of attention. For linear systems, the authors in [38] have discussed the Loewner approach to construct reduced-order systems using transfer function data. Later on, the Loewner approach has been extended to other classes of nonlinear systems, namely, bilinear and QB systems in [7, 27], where data related to generalized transfer functions are required to obtain a reduced-order system.

An important ingredient in the Loewner approach is the construction of the Loewner matrix (\mathbb{L}) and the shifted Loewner matrix (\mathbb{L}_s). One way to construct the matrices \mathbb{L} and \mathbb{L}_s is either by using an experimental setup or by using numerical evaluations of the generalized transfer functions, which is the primary inspiration of the method. However, there is a strong connection with interpolation of (generalized-) transfer functions, corresponding to a given system. As a result, we, alternatively, can construct the latter matrices by projection for a given realization of a system, ensuring the interpolation conditions.

For an example, let us consider 4 frequency measurements $\mathbf{H}(\sigma_1)$, $\mathbf{H}(\sigma_2)$, $\mathbf{H}(\mu_1)$, and $\mathbf{H}(\mu_2)$, where $\mathbf{H}(s) := \mathbf{C}(s\mathbf{E} - \mathbf{A})^{-1}\mathbf{B} \in \mathbb{C}$ is the transfer function of a linear SISO system with the system matrices $(\mathbf{E}, \mathbf{A}, \mathbf{B}, \mathbf{C})$. As shown, e.g., in [8], the matrices \mathbb{L} and \mathbb{L}_s , using the data points and letting $\sigma_{\{1,2\}}$ and $\mu_{\{1,2\}}$ be the right and left interpolation points, can be constructed as follows:

$$(3.13) \quad \mathbb{L}(i, j) = \frac{\mathbf{H}(\mu_i) - \mathbf{H}(\sigma_j)}{\mu_i - \sigma_j}, \quad \mathbb{L}_s(i, j) = \frac{\mu_i \mathbf{H}(\mu_i) - \sigma_j \mathbf{H}(\sigma_j)}{\mu_i - \sigma_j},$$

where $i, j \in \{1, 2\}$. Moreover, if the matrices \mathbf{V} and \mathbf{W} are given as

$$\mathbf{V} = [(\sigma_1 \mathbf{E} - \mathbf{A})^{-1} \mathbf{B}, \quad (\sigma_2 \mathbf{E} - \mathbf{A})^{-1} \mathbf{B}],$$

$$\mathbf{W} = [(\mu_1 \mathbf{E} - \mathbf{A})^{-T} \mathbf{C}^T, \quad (\mu_2 \mathbf{E} - \mathbf{A})^{-T} \mathbf{C}^T],$$

then the matrices \mathbb{L} and \mathbb{L}_s , shown in (3.13), can also be constructed as

$$(3.14) \quad \mathbb{L} = -\mathbf{W}^T \mathbf{E} \mathbf{V}, \quad \mathbb{L}_s = -\mathbf{W}^T \mathbf{A} \mathbf{V}.$$

Analogous features can also be seen for bilinear and QB systems [7, 27]. It is preferable to construct \mathbb{L} and \mathbb{L}_s using the data if the data corresponding to the transfer function can be either computed cheaply by an explicit expression or determined by an experimental setup. In this paper, we focus on the method to determine \mathbb{L} and \mathbb{L}_s by projection shown in (3.14), which can be of greater use when we have a system realization.

In this paper, we assume that a realization of the polynomial system (1.1) is given and let the projection matrices \mathbf{V} and \mathbf{W} be as defined in Theorem 3.1. Thus, we have the projected matrices as follows:

$$\mathbb{L} = -\mathbf{W}^T \mathbf{E} \mathbf{V}, \quad \mathbb{B} = \mathbf{W}^T \mathbf{B}, \quad \mathbb{H}_\xi = \mathbf{W}^T \mathbf{H}_\xi \mathbf{V}^\otimes, \quad \xi \in \{2, \dots, d\},$$

$$\mathbb{L}_s = -\mathbf{W}^T \mathbf{A} \mathbf{V}, \quad \mathbb{C} = \mathbf{C} \mathbf{V}, \quad \mathbb{N}_\eta = \mathbf{W}^T \mathbf{N}_\eta (\mathbf{I}_m \otimes \mathbf{V}^\otimes), \quad \eta \in \{1, \dots, d\}.$$

This allows us to extend the result of bilinear/QB systems [7, 27] to polynomial systems.

LEMMA 3.4. *Let $(\mathbb{L}_s, \mathbb{L})$ be a regular pencil such that none of the interpolation points are its eigenvalues. Then, the sextuple $(\widehat{\mathbf{C}}, \widehat{\mathbf{E}}, \widehat{\mathbf{A}}, \widehat{\mathbf{H}}_\xi, \widehat{\mathbf{N}}_\eta, \widehat{\mathbf{B}})$ defined as*

$$\widehat{\mathbf{E}} = -\mathbb{L}, \quad \widehat{\mathbf{A}} = -\mathbb{L}_s, \quad \widehat{\mathbf{H}}_\xi = \mathbb{H}_\xi, \quad \widehat{\mathbf{N}}_\eta = \mathbb{N}_\eta, \quad \widehat{\mathbf{B}} = \mathbb{B},$$

is a minimal realization that interpolates at given interpolation points.

Furthermore, if we consider far more interpolation points on the imaginary axis than required to capture input-output mapping to construct the matrices \mathbf{V} and \mathbf{W} then the matrix pencil $(\mathbb{L}_s, \mathbb{L})$ is singular. In this case, there exists a lower-order realization that interpolates at given interpolation points. To obtain a lower-order realization, we consider the SVD of the following matrices, composed of \mathbb{L} and \mathbb{L}_s :

$$(3.15) \quad \begin{bmatrix} \mathbb{L} & \mathbb{L}_s \end{bmatrix} = \mathbf{Y}_1 \Sigma_1 \mathbf{X}_1^T,$$

$$(3.16) \quad \begin{bmatrix} \mathbb{L} \\ \mathbb{L}_s \end{bmatrix} = \mathbf{Y}_2 \Sigma_2 \mathbf{X}_2^T,$$

where the diagonal entries of Σ_1 and Σ_2 are in nonincreasing order. Based on the first r columns of \mathbf{Y}_1 and \mathbf{X}_2 , let us construct \mathbf{Y} and \mathbf{X} , i.e., $\mathbf{Y} := \mathbf{Y}_1(:, 1:r)$ and $\mathbf{X} := \mathbf{X}_2(:, 1:r)$ that lead us to the following theorem.

THEOREM 3.5. *If the sextuple $(\widehat{\mathbf{C}}, \widehat{\mathbf{E}}, \widehat{\mathbf{A}}, \widehat{\mathbf{H}}_\xi, \widehat{\mathbf{N}}_\eta, \widehat{\mathbf{B}})$ is given by*

$$(3.17) \quad \begin{aligned} \widehat{\mathbf{E}} &= -\mathbf{Y}^T \mathbb{L} \mathbf{X}, & \widehat{\mathbf{B}} &= \mathbf{Y}^T \mathbb{B}, & \widehat{\mathbf{H}}_\xi &= \mathbf{Y}^T \mathbb{H}_\xi \mathbf{X}^\circledast, & \xi &\in \{2, \dots, d\}, \\ \widehat{\mathbf{A}} &= -\mathbf{Y}^T \mathbb{L}_s \mathbf{X}, & \widehat{\mathbf{C}} &= \mathbb{C} \mathbf{X}, & \widehat{\mathbf{N}}_\eta &= \mathbf{Y}^T \mathbb{N}_\eta (\mathbf{I}_m \otimes \mathbf{X}^\circledast), & \eta &\in \{1, \dots, d\}, \end{aligned}$$

then the realization is such that it approximately interpolates at given interpolation points. Moreover, if r is chosen such that it truncates all 0 singular values, then it constructs an exact interpolant lower-order realization.

Note that there are essentially two steps involved in order to get the approximate interpolant realization (3.17). In the first step, we require matrices such as $\mathbb{L}, \mathbb{L}_s, \mathbb{H}_\xi$, and \mathbb{N}_η , which are generally dense, thus becoming unmanageable. This is followed by compressing these matrices by using \mathbf{X} and \mathbf{Y} . However, upon closer inspection, we can directly determine the realization (3.17) without forming matrices such as \mathbb{H}_ξ and \mathbb{N}_η . We can determine reduced-order systems by directly projecting the original system matrices using appropriate projection matrices. If we define the *effective* projection matrices as

$$(3.18) \quad \mathbf{V}_{\text{eff}} := \mathbf{V} \mathbf{X}, \quad \text{and} \quad \mathbf{W}_{\text{eff}} := \mathbf{W} \mathbf{Y},$$

then the reduced-order system can be determined in a traditional projection framework of the original system (1.1) as follows:

$$(3.19) \quad \begin{aligned} \widehat{\mathbf{E}} &= \mathbf{W}_{\text{eff}}^T \mathbf{E} \mathbf{V}_{\text{eff}}, & \widehat{\mathbf{A}} &= \mathbf{W}_{\text{eff}}^T \mathbf{A} \mathbf{V}_{\text{eff}}, & \widehat{\mathbf{H}}_\xi &= \mathbf{W}_{\text{eff}}^T \mathbf{H}_\xi \mathbf{V}_{\text{eff}}^\circledast, \\ \widehat{\mathbf{B}} &= \mathbf{W}_{\text{eff}}^T \mathbf{B}, & \widehat{\mathbf{C}} &= \mathbf{C} \mathbf{V}_{\text{eff}}, & \widehat{\mathbf{N}}_\eta &= \mathbf{V}_{\text{eff}}^T \mathbf{N}_\eta (\mathbf{I}_m \otimes \mathbf{V}_{\text{eff}}^\circledast), \end{aligned}$$

where $\xi \in \{2, \dots, d\}$ and $\eta \in \{1, \dots, d\}$. We point out that it is advantageous to determine reduced system matrices as shown in (3.19); this way, we are not required to form large dense matrices such as \mathbb{H}_ξ and \mathbb{N}_η . We can compute reduced matrices by multiplying efficiently the sparse and supersparse¹ original matrices \mathbf{H}_ξ and \mathbf{N}_η

¹Supersparsity of a matrix is defined as a ratio of the number of nonzero distinct numbers to the total number of nonzero elements; see, e.g., [37, p. 50].

Algorithm 3.1 MOR for polynomial systems (Lb.Ps- Algo).

Input: The system matrices $\mathbf{E}, \mathbf{A}, \mathbf{H}_\xi, \mathbf{N}_\eta, \mathbf{B}, \mathbf{C}$, and interpolation points σ_i, μ_i and corresponding tangential directions \mathbf{b}_i and \mathbf{c}_i , the reduced order r .

Output: The reduced system matrices $\hat{\mathbf{E}}, \hat{\mathbf{A}}, \hat{\mathbf{H}}_\xi, \hat{\mathbf{N}}_\eta, \hat{\mathbf{B}}, \hat{\mathbf{C}}$.

- 1: Determine \mathbf{V} and \mathbf{W} as shown in (3.7).
- 2: Define the Loewner and shifted Loewner matrices as follows:

$$\mathbb{L} = -\mathbf{W}^T \mathbf{E} \mathbf{V}, \quad \mathbb{L}_s = -\mathbf{W}^T \mathbf{A} \mathbf{V}.$$

- 3: Compute SVD of the matrices:

$$[\mathbb{L}, \mathbb{L}_s] = \mathbf{Y}_1 \Sigma_1 \mathbf{X}_1^T, \quad \begin{bmatrix} \mathbb{L} \\ \mathbb{L}_s \end{bmatrix} = \mathbf{Y}_2 \Sigma_2 \mathbf{X}_2^T.$$

- 4: Define $\mathbf{Y} := \mathbf{Y}_1(:, 1:r)$ and $\mathbf{X} := \mathbf{X}_2(:, 1:r)$.

- 5: Determine compact projection matrices:

$$\mathbf{V}_{\text{eff}} := \text{orth}(\mathbf{V} \mathbf{X}) \text{ and } \mathbf{W}_{\text{eff}} := \text{orth}(\mathbf{W} \mathbf{Y}).$$

- 6: Determine the reduced-order system as follows:

$$\begin{aligned} \hat{\mathbf{E}} &= \mathbf{W}_{\text{eff}}^T \mathbf{E} \mathbf{V}_{\text{eff}}, & \hat{\mathbf{A}} &= \mathbf{W}_{\text{eff}}^T \mathbf{A} \mathbf{V}_{\text{eff}}, & \hat{\mathbf{H}}_\xi &= \mathbf{W}_{\text{eff}} \mathbf{H}_\xi \mathbf{V}_{\text{eff}}^{\otimes \xi}, \\ \hat{\mathbf{B}} &= \mathbf{W}_{\text{eff}}^T \mathbf{B}, & \hat{\mathbf{C}} &= \mathbf{C} \mathbf{V}_{\text{eff}}, & \hat{\mathbf{N}}_\eta &= \mathbf{V}_{\text{eff}}^T \mathbf{N}_\eta (\mathbf{I}_m \otimes \mathbf{V}_{\text{eff}}^{\otimes \eta}). \end{aligned}$$

with \mathbf{V}_{eff} and \mathbf{W}_{eff} . Having all these results, we briefly sketch the steps to determine reduced-order systems in Algorithm 3.1. However, an important computational aspect related to tensor computations such as $\mathbf{W}_{\text{eff}} \mathbf{H}_\xi \mathbf{V}_{\text{eff}}^{\otimes \xi}$ still remains, which is discussed in the next section.

4. Computational aspects and application of CUR. In this section, we investigate two important computational aspects. These are related to evaluating the nonlinear terms of the reduced-order systems (3.19) and further approximating the terms using a CUR matrix approximation to accelerate simulation of reduced-order systems.

4.1. Efficient evaluation of the nonlinear terms in the reduced-order models (ROMs). Let us begin with the computational effort related to evaluating, e.g., $\hat{\mathbf{H}}_\xi := \mathbf{W}_{\text{eff}}^T \mathbf{H}_\xi \mathbf{V}_{\text{eff}}^{\otimes \xi}$. It can be noticed that a direct computation of the above terms requires the computation of $\mathbf{V}_{\text{eff}}^{\otimes \xi}$. Generally, the matrix \mathbf{V}_{eff} is a dense matrix; thus, the computation related to $\mathbf{V}_{\text{eff}}^{\otimes \xi}$ is of complexity $\mathcal{O}((n \cdot r)^\xi)$, which easily becomes an unmanageable task. For $\xi = 2$, the authors in [15] have proposed a method using tensor algebra to compute $\hat{\mathbf{H}}_2$ without explicitly forming $\mathbf{V}_{\text{eff}} \otimes \mathbf{V}_{\text{eff}}$. On the other hand, the authors in [19] have aimed at exploiting the structure of the nonlinear operators, typically arising in governing equations, thus also leading to an efficient method to compute $\hat{\mathbf{H}}_2$.

In this paper, we focus on the latter approach, where the explicit nonlinear operator of the governing equations is utilized to compute $\hat{\mathbf{H}}_\xi$. Extending the discussion in [19], in principle, we can write the term $\mathbf{H}_\xi \mathbf{x}^{\otimes \xi}$ in the system (1.1) in the Hadamard product form as follows:

$$(4.1) \quad \mathbf{H}_\xi \mathbf{x}^{\otimes \xi} = \mathcal{A}_1 \mathbf{x} \circ \cdots \circ \mathcal{A}_\xi \mathbf{x},$$

where \circ denotes the Hadamard product and $\mathcal{A}_i \in \mathbb{R}^{n \times n}$ are the constant matrices

depending on the nonlinear operator in the governing equation. In order to reduce these nonlinear terms, resulting in a reduced-order system, we proceed as follows. First, we substitute $\mathbf{x}(t) \approx \mathbf{V}_{\text{eff}} \widehat{\mathbf{x}}(t)$, where $\mathbf{x}(t) \in \mathbb{R}^n$ and $\widehat{\mathbf{x}}(t) \in \mathbb{R}^r$ are the original and reduced state vectors, respectively, and then multiply $\mathbf{W}_{\text{eff}}^T$ from the left-hand side, thus leading to the corresponding nonlinear term:

$$\widehat{\mathbf{H}}_{\xi} \widehat{\mathbf{x}}^{\otimes \xi} = \mathbf{W}_{\text{eff}}^T \left(\left(\widehat{\mathcal{A}}_1 \widehat{\mathbf{x}} \right) \circ \cdots \circ \left(\widehat{\mathcal{A}}_{\xi} \widehat{\mathbf{x}} \right) \right),$$

where $\widehat{\mathcal{A}}_i = \mathcal{A}_i \mathbf{V}_{\text{eff}}$, $i \in \{1, \dots, \xi\}$. Next, we use the relation between the Hadamard product and the Kronecker product, that is,

$$\mathcal{P}p \circ \mathcal{Q}q = \begin{bmatrix} \mathcal{P}(1, :) \otimes \mathcal{Q}(1, :) \\ \vdots \\ \mathcal{P}(n, :) \otimes \mathcal{Q}(n, :) \end{bmatrix} (p \otimes q).$$

Thus, we get

$$(4.2) \quad \mathbf{W}_{\text{eff}}^T \left(\left(\widehat{\mathcal{A}}_1 \widehat{\mathbf{x}} \right) \circ \cdots \circ \left(\widehat{\mathcal{A}}_{\xi} \widehat{\mathbf{x}} \right) \right) = \mathbf{W}_{\text{eff}}^T \underbrace{\begin{bmatrix} \widehat{\mathcal{A}}_1(1, :) \otimes \cdots \otimes \widehat{\mathcal{A}}_{\xi}(1, :) \\ \vdots \\ \widehat{\mathcal{A}}_1(n, :) \otimes \cdots \otimes \widehat{\mathcal{A}}_{\xi}(n, :) \end{bmatrix}}_{=: \widehat{\mathcal{A}}} \widehat{\mathbf{x}}^{\otimes \xi}.$$

It can be seen that $\mathbf{W}_{\text{eff}}^T \widehat{\mathcal{A}} = \widehat{\mathbf{H}}_{\xi}$. Summarizing, we can perform computations related to $\widehat{\mathbf{H}}_{\xi}$ efficiently by utilizing the particular structure of the nonlinear terms, without explicitly forming $\mathbf{V}_{\text{eff}}^{\otimes \xi}$. We will illustrate the procedure using a nonlinear partial differential equation in subsection 4.3.

4.2. CUR matrix approximation and ROMs. Next, we discuss another computational issue, due to which we may not achieve the desired reduction in the simulation time even after reducing the original system (1.1). Explaining this issue further, the reduced matrices such as $\widehat{\mathbf{H}}_{\xi} \in \mathbb{R}^{r \times r^{\xi}}$ are generally dense matrices which are multiplied with $\widehat{\mathbf{x}}^{\otimes \xi}$. Thus, the computation $\widehat{\mathbf{H}}_{\xi} \widehat{\mathbf{x}}^{\otimes \xi}$ is of $\mathcal{O}(r^{2\xi+1})$, which increases rapidly with the order of the reduced system or with the polynomial degree (1.1). As a remedy, in this paper, we propose a new procedure to approximate $\widehat{\mathbf{H}}_{\xi} \widehat{\mathbf{x}}^{\otimes \xi}$, which can be computed cheaply. For this, we make use of the *CUR matrix approximation*; see, e.g., [36, 46, 49]. Using this, we can approximate the matrix $\widehat{\mathcal{A}}$, defined in (4.2), as follows:

$$(4.3) \quad \widehat{\mathcal{A}} \approx \mathbf{C} \mathbf{U} \mathbf{R},$$

where $\mathbf{C} \in \mathbb{R}^{n \times n_c}$ and $\mathbf{R} \in \mathbb{R}^{n_r \times r^{\xi}}$ contain wisely chosen n_c columns and n_r rows of the matrix $\widehat{\mathcal{A}}$, respectively, and $\mathbf{U} \in \mathbb{R}^{n_c \times n_r}$ is determined such that it minimizes $\|\widehat{\mathcal{A}} - \mathbf{C} \mathbf{U} \mathbf{R}\|$ in an appropriate norm. There has been significant research on how to choose columns and rows appropriately, leading to a good, or even optimal in some sense, approximation of the matrix. We refer the reader to [36, 46, 49] and references therein for more details. Substituting the relation (4.3) in (4.2) results in

$$(4.4) \quad \mathbf{W}_{\text{eff}}^T \widehat{\mathcal{A}} \widehat{\mathbf{x}}^{\otimes \xi} \approx \mathbf{W}_{\text{eff}}^T (\mathbf{C} \mathbf{U} \mathbf{R}) \widehat{\mathbf{x}}^{\otimes \xi}.$$

Next, we closely look at the term $\mathbf{R} \widehat{\mathbf{x}}^{\otimes \xi}$, whose columns are given as

$$(4.5) \quad \left(\widehat{\mathcal{A}}_1(i_r, :) \otimes \cdots \otimes \widehat{\mathcal{A}}_{\xi}(i_r, :) \right) \widehat{\mathbf{x}}^{\otimes \xi},$$

where i_r belongs to the columns chosen by the CUR matrix approximation. We know that $\widehat{\mathcal{A}}_1(i_r, :) = \mathcal{A}_1(i_r, :)\mathbf{V}$. Substituting this relation and $\mathbf{x} \approx \mathbf{V}\widehat{\mathbf{x}}$, we get

$$\begin{aligned} \widehat{\mathcal{A}}_1(i_r, :) \otimes \cdots \otimes \widehat{\mathcal{A}}_\xi(i_r, :)\widehat{\mathbf{x}}^{\otimes 3} &= (\mathcal{A}_1(i_r, :) \otimes \cdots \otimes \mathcal{A}_\xi(i_r, :)) \mathbf{V}^{\otimes 3}\widehat{\mathbf{x}}^{\otimes 3} \\ &\approx (\mathcal{A}_1(i_r, :) \otimes \cdots \otimes \mathcal{A}_\xi(i_r, :)) \mathbf{x}^{\otimes 3} := \text{NL}_{i_r}. \end{aligned}$$

Comparing the above quantity with (4.1), it can be noticed that the quantity NL_{i_r} is nothing but the computation of the corresponding nonlinearity of the original system at a particular grid point. Furthermore, the term $\mathbf{W}_{\text{eff}}^T \mathcal{C}\mathcal{U} \in \mathbb{R}^{r \times n_r}$ can be precomputed. This idea is very closely related to empirical interpolation methods, which are commonly used in reduced basis methods or proper orthogonal decomposition for nonlinear systems to reduce the computational cost related to nonlinear terms [11, 22]. We will demonstrate the concept using an example in the following subsection.

4.3. An illustration using Chafee–Infante equation. In the following, we illustrate the computation of the reduced nonlinear term $\widehat{\mathbf{H}}_\xi$ and the usage of the CUR decomposition with the help of the Chafee–Infante equation. At this stage, we avoid describing the governing equations of the Chafee–Infante equation; we provide a detailed description of it in the numerical section. However, at the moment, we just note that it has cubic nonlinearity, i.e., $-v^3$, where v is the dependent variable. Hence, if the system is written in the form given in (1.1), we have the following nonlinear term:

$$\mathbf{H}_3 \mathbf{x}^{\otimes 3} := -\mathbf{x} \circ \mathbf{x} \circ \mathbf{x}.$$

If the above term is reduced using the projection matrices \mathbf{V}_{eff} and \mathbf{W}_{eff} as shown in (3.19), then we obtain

$$\begin{aligned} \mathbf{W}_{\text{eff}}^T \mathbf{H}_3 \mathbf{V}_{\text{eff}} \widehat{\mathbf{x}}^{\otimes 3} &= \widehat{\mathbf{H}}_3 \widehat{\mathbf{x}}^{\otimes 3} \\ &= \mathbf{W}_{\text{eff}}^T (\mathbf{V}_{\text{eff}} \widehat{\mathbf{x}} \circ \mathbf{V}_{\text{eff}} \widehat{\mathbf{x}} \circ \mathbf{V}_{\text{eff}} \widehat{\mathbf{x}}) \\ (4.6) \quad &= \mathbf{W}_{\text{eff}}^T \underbrace{\begin{bmatrix} \mathbf{V}_{\text{eff}}(1, :) \otimes \mathbf{V}_{\text{eff}}(1, :) \otimes \mathbf{V}_{\text{eff}}(1, :) \\ \vdots \\ \mathbf{V}_{\text{eff}}(n, :) \otimes \mathbf{V}_{\text{eff}}(n, :) \otimes \mathbf{V}_{\text{eff}}(n, :) \end{bmatrix}}_{=:\widetilde{\mathbf{V}}_{\text{eff}}} (\widehat{\mathbf{x}} \otimes \widehat{\mathbf{x}} \otimes \widehat{\mathbf{x}}). \end{aligned}$$

Equation (4.6) shows that instead of explicitly forming $\mathbf{V}_{\text{eff}}^{\otimes 3}$ to determine $\widehat{\mathbf{H}}_3$, we can compute it by a smart choice of rows and perform the Kronecker products. Furthermore, as discussed earlier, the evaluation of the term $\widehat{\mathbf{H}}_3 \widehat{\mathbf{x}}^{\otimes 3}$, in general, is of complexity $\mathcal{O}(r^7)$, which might be expensive if the order of the reduced system (r) is rather large. Also, we stress that the term $\widehat{\mathbf{H}}_3 \widehat{\mathbf{x}}^{\otimes 3}$ needs to be computed at each time step for every simulation. To reduce the computational complexity, we aim at further approximating $\widehat{\mathbf{H}}_3 \widehat{\mathbf{x}}^{\otimes 3}$. For this, we first apply the CUR matrix approximation to the matrix $\widetilde{\mathbf{V}}_{\text{eff}}$, defined in (4.6), to approximate it by using selected columns and rows, that is,

$$(4.7) \quad \widetilde{\mathbf{V}}_{\text{eff}} \approx \mathcal{C}_v \mathcal{U}_v \mathcal{R}_v,$$

where $\mathcal{C}_v \in \mathbb{R}^{r \times n_c}$ and $\mathcal{R}_v \in \mathbb{R}^{n_r \times r^3}$ consist of columns and rows of $\widetilde{\mathbf{V}}_{\text{eff}}$, respectively. Let us assume that $\mathcal{I}_{\mathbf{R}} \subseteq \{1, \dots, n\}$ denotes the indices, leading to the construction

of the matrix \mathcal{R}_v in (4.7), i.e.,

$$\mathcal{R}_v = \tilde{\mathbf{V}}_{\text{eff}}(\mathcal{I}_{\mathbf{R}}, :).$$

As a result, we use the relation (4.7) in (4.6) to obtain

$$\mathbf{W}_{\text{eff}}^T \mathbf{Y}_{\text{eff}} \hat{\mathbf{x}}^{\otimes 3} \approx \mathbf{W}_{\text{eff}}^T \mathcal{C}_v \mathcal{U}_v \mathcal{R}_v \hat{\mathbf{x}}^{\otimes 3} = \underbrace{\mathbf{W}_{\text{eff}}^T \mathcal{C}_v \mathcal{U}_v}_{=: \Psi} \left(\tilde{\mathbf{V}}_{\text{eff}}(\mathcal{I}_{\mathbf{R}}, :)\hat{\mathbf{x}}^{\otimes 3} \right).$$

Now, it can be noticed that the term $\tilde{\mathbf{V}}_{\text{eff}}(\mathcal{I}_{\mathbf{R}}, :)\hat{\mathbf{x}}^{\otimes 3}$ is nothing but evaluating the nonlinearity (in this case, it is a cubic nonlinearity) at indices $\mathcal{I}_{\mathbf{R}}$. As a result, we need to determine the nonlinearity at n_r points. Moreover, the matrix $\Psi \in \mathbb{R}^{r \times r_v}$ can be precomputed. This is exactly the idea of hyperreduction methods such as (discrete) empirical interpolation method ((D)EIM) proposed in [11, 22] in the case of nonlinear MOR. However, a major difference between the methodology in this paper and (D)EIM is that we do not require time-domain snapshots of the nonlinearity as needed in the case of (D)EIM. We approximate the nonlinear terms in the reduced-order systems. Summarizing, for the Chafee–Infante equation in the end, we have

$$(4.8) \quad \mathbf{W}_{\text{eff}}^T \tilde{\mathbf{V}}_{\text{eff}}(\hat{\mathbf{x}} \otimes \hat{\mathbf{x}} \otimes \hat{\mathbf{x}}) \approx \Psi(\tilde{\mathbf{x}} \circ \tilde{\mathbf{x}} \circ \tilde{\mathbf{x}}),$$

where $\tilde{\mathbf{x}} = \mathbf{V}_{\text{eff}}(\mathcal{I}_{\mathbf{R}}, :)\hat{\mathbf{x}}$, which is of complexity $\mathcal{O}(r \cdot n_r^2)$.

Remark 4.1. In the above, we have focused on the computational aspect related to $\hat{\mathbf{H}}_{\xi}$ and $\hat{\mathbf{H}}_{\xi} \hat{\mathbf{x}}^{\otimes 3}$. Analogously, a complexity reduction can be performed for computing $\hat{\mathbf{N}}_{\eta}$ and $\hat{\mathbf{N}}_{\eta} \hat{\mathbf{x}}^{\otimes 3}$.

5. Numerical results. In this section, we illustrate the efficiency of the proposed methods by means of two nonlinear partial differential equations. All the simulations were done on an Intel[®] Core[™]i7-6700 CPU@3.40 GHz, 8 MB cache, 8 GB RAM, Ubuntu 16.04, MATLAB[®] Version 9.1.0.441655(R2016b) 64-bit(glnxa64). In the following, we note some details used in the numerical simulation:

- All original and reduced-order systems are integrated by the routine `ode15s` in MATLAB with relative error and absolute error tolerances of 10^{-10} .
- We measure the output at 500 equidistant points within the time interval $[0, T]$, where T is the end time.
- We choose interpolation points for the frequency (s) in a logarithmic scale for a given frequency range.

5.1. Chafee–Infante equation. In our first example, we deal with a widely considered one-dimensional Chafee–Infante equation. Its governing equation and boundary conditions are given as follows:

$$(5.1) \quad \begin{aligned} \dot{v}(t) &= v_{xx} + v(1 - v^2), & x \in (0, L) \times (0, T), & \quad v(0, \cdot) = u(t), \quad (0, T), \\ v_x(L, \cdot) &= 0, & (0, T), & \quad v(x, 0) = 0, \quad (0, L). \end{aligned}$$

MOR of this example has been considered in various papers [15, 18, 19, 27], where the authors have proposed different methods to reduce it. The governing equation has a cubic nonlinearity. In the literature, a common approach to reduce such a cubic nonlinear system via system-theoretic MOR is twofold. First, it is to rewrite the cubic system into a QB system by introducing auxiliary variables. Thereafter, one can reduce it by employing a MOR scheme for QB systems such as balanced

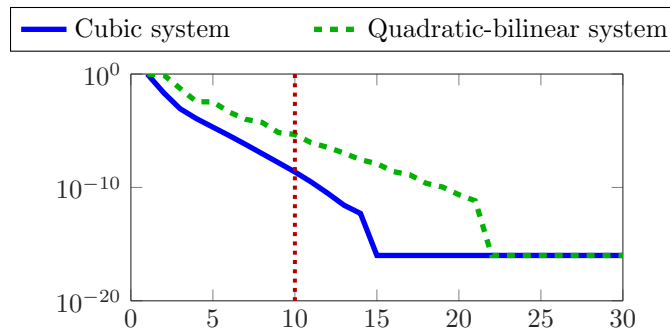


FIG. 5.1. Relative decay of singular values based on the Loewner pencils, obtained using the original cubic system and its equivalent transformed QB system.

truncation [18], and interpolation-based approaches, e.g., [3, 15, 19]. However, in this process, we lose the original cubic nonlinearity structure in the reduced-order system. On the other hand, our proposed method allows us to reduce a cubic system directly, having preserved the polynomial structure in the reduced-order system.

We set the domain length $L = 1$. The system of equations (5.1) is discretized using a finite-difference method by taking $k = 500$ grid points. Next, we aim at constructing a reduced cubic system by applying Algorithm 3.1. For this purpose, we consider the frequency range $[10^{-3}, 10^3]$. For comparison, we also rewrite the cubic system into the QB form, which results in an equivalent QB system of order 1000. We consider the same frequency range in order to employ Algorithm 3.1 to construct a reduced QB system.

First, in Figure 5.1, we observe the decay of the singular values, obtained from the Loewner pencil $(s\mathbb{L} - \mathbb{L}_s)$. We observe that the singular values related to the original cubic system decay faster as compared to its equivalent QB form. Hence, for the same order of the reduced-order system, we can anticipate a better quality reduced system. Next, we construct the reduced cubic and QB systems of order $r = 10$ using Petrov–Galerkin (two-sided) projection. To test the quality of both reduced cubic and QB systems, we perform time-domain simulation using control inputs $u^{(1)}(t) = 10(\sin(\pi t) + 1)$ and $u^{(2)}(t) = 5(te^{-t})$ and compare them in Figures 5.2 and 5.3 by showing the responses and relative errors. As can be seen from these figures, the reduced cubic system captures the dynamics of the original system much better as compared to the QB system; precisely, we gain up to 3 orders of magnitude better accuracy using the new method.

Typically, one expects that if QB and cubic systems are of the same orders, then simulating the QB system may be cheaper. However, we have observed that if a polynomial system is written into QB form, then the QB system typically becomes stiffer. This means that we require finer time steps to obtain the same accuracy of the solution. For completeness, we report the computational time to simulate both reduced-order systems in Table 5.1. We observe that although both reduced-order systems are of different polynomial degrees but have the same orders, the computational costs are very similar.

5.2. The FitzHugh–Nagumo (FHN) model. As a second example, we consider the FHN system, which describes basic neuronal dynamics. This is a coupled cubic nonlinear system, whose governing equations and boundary conditions are as

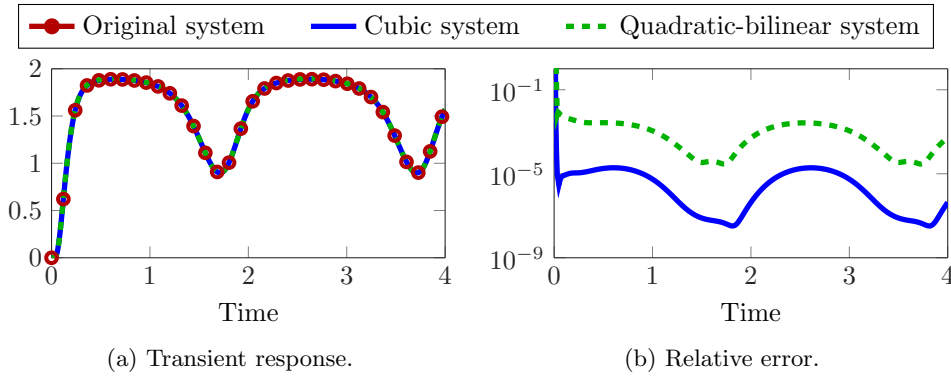


FIG. 5.2. Chafee-Infante equation: a comparison of the original and reduced-order systems for the input $u^{(1)} = 10(\sin(\pi t) + 1)$.

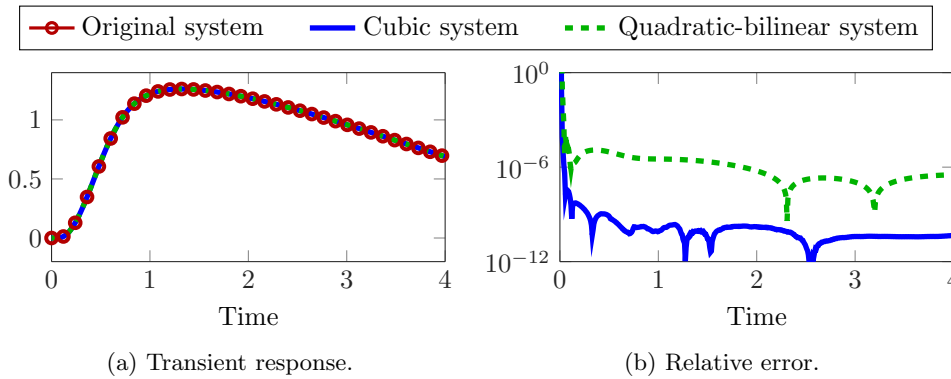


FIG. 5.3. Chafee-Infante equation: a comparison of the original and reduced-order systems for the input $u^{(2)} = 5(te^{-t})$.

TABLE 5.1

Chafee-Infante equation: CPU time comparison of reduced-order systems for both inputs.

Input	Cubic systems	QB system
$u^{(1)}(t)$	0.3103 s	0.2709 s
$u^{(2)}(t)$	0.1013 s	0.0835 s

follows:

$$(5.2) \quad \begin{aligned} \epsilon v_t &= \epsilon^2 v_{xx} + v(v - 0.1)(1 - v) - w + q, \\ w_t &= hv - \gamma w + q \end{aligned}$$

with boundary conditions

$$\begin{aligned} v(x, 0) &= 0, & w(x, 0) &= 0, & x &\in (0, L), \\ v_x(0, t) &= i_0(t), & v_x(1, t) &= 0, & t &\geq 0, \end{aligned}$$

where $h = 0.05, \gamma = 2, q = 0.05, L = 0.1$, and i_0 acts as an actuating control input which takes the values $5 \cdot 10^4 t^3 e^{-t}$. In brief, the variables v and w denote the activation and deactivation of a neuron, respectively. We discretize the governing equation using

a finite difference method by taking 100 grid points. This leads to a cubic system of order $n = 200$. We use the same output setting as used, e.g., in [19]. The system has two inputs and two outputs, thus, is an MIMO system. The MOR problem related to the FHN system has been considered by several researchers; see, e.g., [18, 19, 22]. Similarly to the previous example, system-theoretic MOR of the FHN system has also been considered by first rewriting it into a QB system and employing MOR schemes such as interpolation based and balanced truncation to reduce it. Thus, we obtain an equivalent QB system of order 300. However, by doing so, we lose the original nonlinear structure.

We apply Algorithm 3.1 to obtain reduced-order systems for the original cubic and its equivalent QB systems; we choose 200 points in the frequency range $[10^{-2}, 10^2]$. We construct reduced cubic and QB systems of order $r = 20$, which are denoted by cubic (two-sided) and QB (two-sided) systems for future reference. To determine the quality of the reduced systems, we perform time-domain simulation. We observe that the obtained cubic two-sided reduced system captures the dynamics of the original system very well, whereas the reduced QB (two-sided) system is unstable. This illustrates a common shortcoming of Algorithm 3.1 that it does not always result in a stable reduced system, and also shows that preserving the polynomial structure can be beneficial.

As a remedy, we propose to obtain a reduced-order system using Galerkin (one-sided) projection. For this, we determine the matrix \mathbf{V} at step 1 in Algorithm 3.1 and set $\mathbf{W} = \mathbf{V}$. This is followed by determining \mathbf{X} as shown in step 4 of the algorithm and determine the projection matrix \mathbf{V}_{eff} . Subsequently, we set $\mathbf{W}_{\text{eff}} = \mathbf{V}_{\text{eff}}$ and compute a reduced-order system. As a result, we have a reduced-order system by one-sided projection instead of two-sided. An advantage of doing one-sided projection is the (local) stability of the reduced-order system in some cases. Next, we also compute reduced systems of order $r = 20$ using the cubic and its equivalent QB form, using one-sided projection, that are referred to as cubic (one-sided) and QB (one-sided) systems, respectively.

For comparisons, we first plot the decay of singular values in Figure 5.4, indicating a faster decay for cubic systems as compared to the equivalent QB systems. Furthermore, we compare the transient response of all reduced-order systems in Figure 5.5, which shows that the reduced cubic systems (both one sided and two sided) perform much better as compared to the reduced QB system (one sided), and as stated earlier, the reduced QB system (two sided) is unstable. Interestingly, we observe that the reduced cubic systems, using two-sided and one-sided projection, tend to perform equally well as the time progresses but in the beginning, the reduced system obtained using two-sided projection performs better. Furthermore, we mention that the same order of accuracy as the reduced QB system (one sided) of order $r = 20$ can be obtained from a reduced cubic system of order $r = 6$ only.

Note that one of the characteristic features of the FHN model is the limit cycle that exhibits the activation and deactivation of the neuron. As we know, the model is a cubic system. Thus, when we aim at constructing a reduced cubic system that exhibits the similar limit cycles, we can construct a reduced cubic system of order as low as $r = 2$. On the other hand, if we first rewrite the cubic system into the QB form, we need the minimum order $r = 15$ for a reduced QB system to capture the limit cycle behavior. This illustrates that keeping the original polynomial structure into reduced-order systems can lead to much better reduced-order systems.

Furthermore, similarly to the previous example, we compare computational cost to simulate reduced-order systems in Table 5.2. We observe that the reduced QB

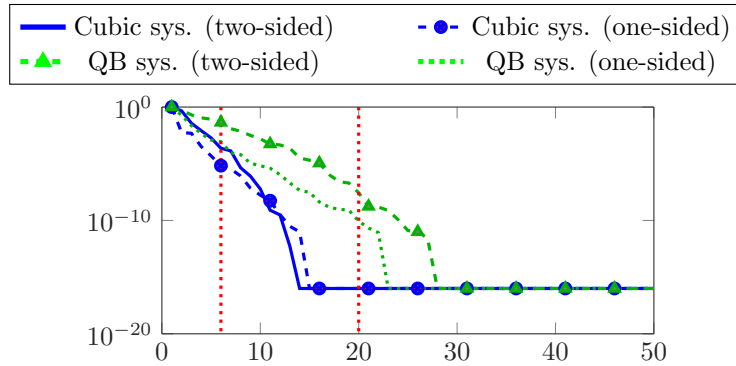


FIG. 5.4. *FHN model: relative decay of singular values based on the Loewner pencils, obtained via one-sided and two-sided projections of the corresponding systems.*

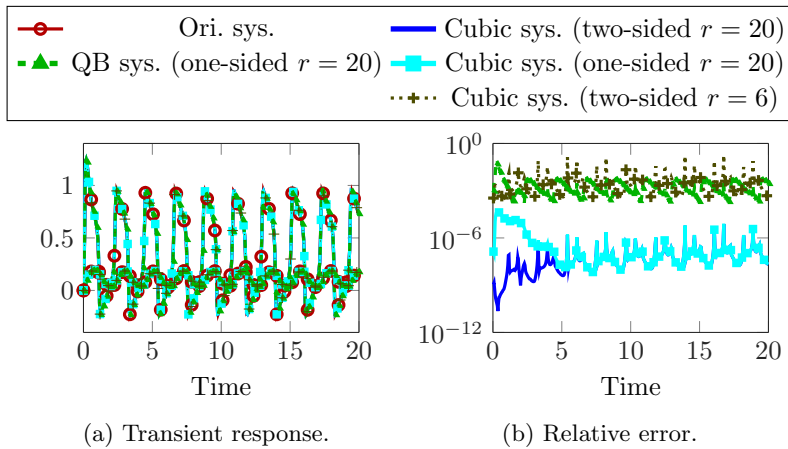


FIG. 5.5. *FHN model: a comparison of the original and reduced cubic and QB systems using one-sided and two-sided projections, having employed Algorithm 3.1.*

TABLE 5.2
FHN model: CPU time comparison of reduced-order systems.

Cubic (two-sided)	Cubic (one-sided)	QB (one-sided)	Cubic ($r = 6$)
1.5232 s	1.4816 s	1.2704 s	0.9753 s

system (one sided) is slightly faster compared to the reduced cubic systems when the orders of the reduced systems are kept the same. But the reduced cubic system of order 6 and the reduced QB system of order 20 have a similar accuracy in terms of time-domain simulation, but the reduced cubic system ($r = 6$) requires nearly 20% less time as compared to the reduced QB system ($r = 20$).

5.3. Usage of CUR in ROM. In this section, we illustrate the usage of the CUR matrix approximation to further approximate the nonlinear reduced terms. For this, we again consider the Chafee–Infante equation. Now, we aim at determining reduced cubic systems using one-sided and two-sided projections. First, in Figure 5.6, we plot the relative decay of the singular values based on the Loewner pencil, ob-

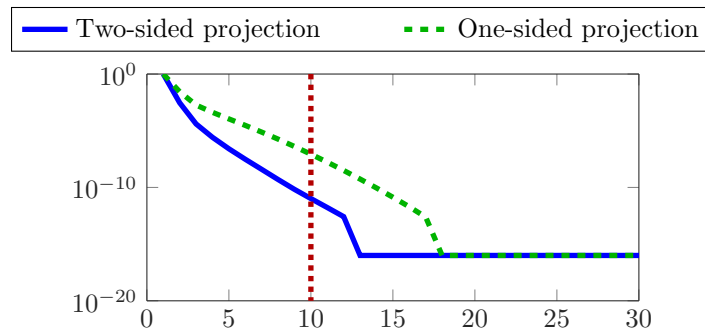


FIG. 5.6. Chafee-Infante equation: relative decay of singular values using the Loewner pencil, obtained via one-sided and two-sided projections.

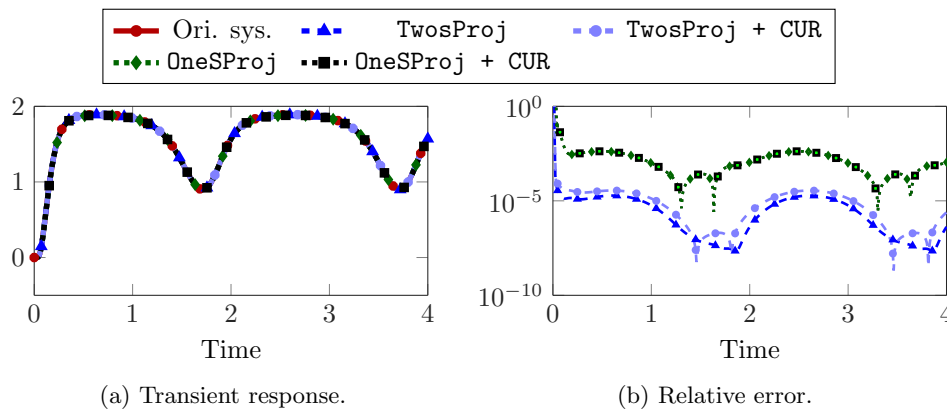


FIG. 5.7. Chafee-Infante equation: a comparison of the original and (*CUR* combined) reduced-order systems for the input $\mathbf{u}^{(1)} = 10(\sin(\pi t) + 1)$.

tained using the one-sided and two-sided projection matrices. We observe that the singular values based on the two-sided projection decay faster relative to the one-sided projection as can be expected.

Next, we construct reduced-order systems of order $r = 10$ using one-sided and two-sided projections using Algorithm 3.1, preserving the polynomial structure. As discussed in section 4, we can further approximate the reduced nonlinear terms by making use of a *CUR* matrix approximation. For *CUR* matrix approximation, we choose 60 rows and 60 columns of \mathcal{V} (defined in (4.6)), which are chosen based on an adaptive sampling proposed in [49]. We would like to mention that the number 60 for row and columns is determined based on a trial and error method. An appropriate automatic method for *CUR* matrix approximation, being suitable for MOR, needs further research. To this end, we have four reduced systems as follows:

- One-sided projection (OneSProj).
- One-sided projection with *CUR* approximation (OneSProj + CUR).
- Two-sided projection (TwoSProj).
- Two-sided projection with *CUR* approximation (TwoSProj + CUR).

To compare the quality of these reduced-order systems, we perform the time-domain simulation of these systems with the original systems for two control inputs, the same

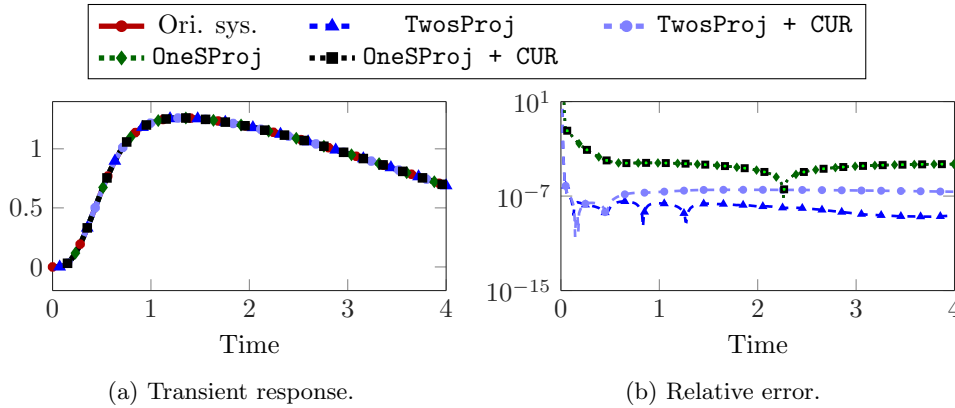


FIG. 5.8. *Chafee-Infante equation: a comparison of the original and (CUR combined) reduced-order systems for the input $\mathbf{u}^{(2)} = 5(e^{-t})$.*

as considered in subsection 5.1 which are then compared in Figures 5.7 and 5.8. We observe that two-sided projection yields the best reduced-order systems among the four reduced-order systems. Furthermore, when the two-sided reduced-order system is combined with CUR matrix approximation, then we notice that the quality of the reduced-order system decreases a little but still provides a very good approximation of the original system. Interestingly, we also notice that CUR matrix approximation applied to the one-sided reduced-order system also performs very well and keeps the quality of the reduced-order models in the same order.

6. Conclusions. In this paper, we have discussed the construction of interpolating reduced-order systems for polynomial systems that preserve the polynomial structure. For this purpose, we have introduced generalized multivariate transfer functions for the systems and have proposed algorithms, inspired by the Loewner approach, to generate good quality reduced-order systems in an automatic way. We have discussed related computational issues and also the usage of the CUR matrix approximation in the simulation of reduced systems. We have illustrated the efficiency of the approaches via several numerical experiments, where we have observed that preserving the polynomial structure in reduced-order systems leads to better reduced-order systems.

In our numerical experiments, we have chosen interpolation points logarithmically in given intervals for frequency. However, choosing these interpolation points wisely can ease the computational burden. In this direction, \mathcal{H}_2 -optimal framework [14, 25, 19] and an adaptive choice of interpolation points based on an error estimate [1, 23] can be extended to polynomial systems. Moreover, in section 4, we have discussed the computational aspect related to $\widehat{\mathbf{H}}_\xi \widehat{\mathbf{x}}^\ominus$ which can be eased with the help of CUR matrix approximation. However, we do not take the projection matrix \mathbf{W} into account for an approximation of the latter term. Thus, it would be valuable to employ the projection matrix \mathbf{W} as well, which could improve the approximation quality. Moreover, it will be of great interest to the MOR community to extend the proposed methodology to other classes of nonlinear systems such as rational nonlinear systems, i.e., those systems containing nonlinear functions, e.g., $\frac{1}{1+x}$ or $e^{-1/x}$. Such systems, for example, arise, in batch chromatography reactors [32] and reactor models

[35]. Such systems can be rewritten as a polynomial system by introducing auxiliary variables as discussed in subsection 2.2, but the goal would be to preserve the original structure of the nonlinearity in the reduced-order systems. Last but not least, one can think of extending the methodologies discussed, e.g., in [2, 17, 31] for descriptor systems to descriptor polynomial systems.

Code availability. A MATLAB implementation of Algorithm 1 can be found on Gitlab under the link https://gitlab.mpi-magdeburg.mpg.de/goyalp/interpolatorymor_polynomials.

REFERENCES

- [1] M. I. AHMAD, P. BENNER, AND L. FENG, *Interpolatory model reduction for quadratic-bilinear systems using error estimators*, Eng. Comput., 36 (2018), pp. 25–44, <https://doi.org/10.1108/EC-04-2018-0162>.
- [2] M. I. AHMAD, P. BENNER, P. GOYAL, AND J. HEILAND, *Moment-matching based model reduction for Navier-Stokes type quadratic-bilinear descriptor systems*, ZAMM Z. Angew. Math. Mech., 97 (2017), pp. 1252–1267, <https://doi.org/10.1002/zamm.201500262>.
- [3] M. I. AHMAD, P. BENNER, AND I. JAIMOUKHA, *Krylov subspace projection methods for model reduction of quadratic-bilinear systems*, IET Control Theory Appl., 10 (2016), pp. 2010–2018, <https://doi.org/10.1049/iet-cta.2016.0415>.
- [4] B. D. O. ANDERSON AND A. C. ANTOULAS, *Rational interpolation and state-variable realizations*, Linear Algebra Appl., 137/138 (1990), pp. 479–509.
- [5] A. C. ANTOULAS, *Approximation of Large-Scale Dynamical Systems*, Adv. Des. Control 6, SIAM, Philadelphia, 2005, <https://doi.org/10.1137/1.9780898718713>.
- [6] A. C. ANTOULAS, C. A. BEATTIE, AND S. GUGERCIN, *Interpolatory model reduction of large-scale dynamical systems*, in Efficient Modeling and Control of Large-Scale Systems, J. Mohammadpour and K. M. Grigoriadis, eds., Springer, New York, 2010, pp. 3–58, https://doi.org/10.1007/978-1-4419-5757-3_1.
- [7] A. C. ANTOULAS, I. V. GOSEA, AND A. C. IONITA, *Model reduction of bilinear systems in the Loewner framework*, SIAM J. Sci. Comput., 38 (2016), pp. B889–B916.
- [8] A. C. ANTOULAS, S. LEFTERIU, AND A. C. IONITA, *A tutorial introduction to the Loewner framework for model reduction*, in Model Reduction and Approximation: Theory and Algorithms, P. Benner, A. Cohen, M. Ohlberger, and K. Willcox, eds., Comput. Sci. Eng. 15, SIAM, Philadelphia, 2017, pp. 335–376, <https://doi.org/10.1137/1.9781611974829.ch8>.
- [9] A. C. ANTOULAS, D. C. SORENSEN, AND S. GUGERCIN, *A survey of model reduction methods for large-scale systems*, Contemp. Math., 280 (2001), pp. 193–219.
- [10] P. ASTRID, S. WEILAND, K. WILLCOX, AND T. BACKX, *Missing point estimation in models described by proper orthogonal decomposition*, IEEE Trans. Automat. Control, 53 (2008), pp. 2237–2251.
- [11] M. BARRAULT, Y. MADAY, N. C. NGUYEN, AND A. T. PATERA, *An ‘empirical interpolation’ method: Application to efficient reduced-basis discretization of partial differential equations*, C. R. Math. Acad. Sci. Paris, 339 (2004), pp. 667–672.
- [12] U. BAUR, P. BENNER, AND L. FENG, *Model order reduction for linear and nonlinear systems: A system-theoretic perspective*, Arch. Comput. Methods Eng., 21 (2014), pp. 331–358, <https://doi.org/10.1007/s11831-014-9111-2>.
- [13] P. BENNER, P. GOYAL, AND I. PONTES DUFF, *Identification of Dominant Subspaces for Linear Structured Parametric Systems and Model Reduction*, preprint, <https://arxiv.org/abs/1910.13945>, 2019.
- [14] P. BENNER AND T. BREITEN, *Interpolation-based \mathcal{H}_2 -model reduction of bilinear control systems*, SIAM J. Matrix Anal. Appl., 33 (2012), pp. 859–885.
- [15] P. BENNER AND T. BREITEN, *Two-sided projection methods for nonlinear model order reduction*, SIAM J. Sci. Comput., 37 (2015), pp. B239–B260, <https://doi.org/10.1137/14097255X>.
- [16] P. BENNER, A. COHEN, M. OHLBERGER, AND K. WILLCOX, EDs., *Model Reduction and Approximation: Theory and Algorithms*, Comput. Sci. Eng. 15, SIAM, Philadelphia, 2017, <https://doi.org/10.1137/1.9781611974829>.
- [17] P. BENNER AND P. GOYAL, *Multipoint interpolation of Volterra series and \mathcal{H}_2 -model reduction for a family of bilinear descriptor systems*, Systems Control Lett., 97 (2016), pp. 1–11, <https://doi.org/10.1016/j.sysconle.2016.08.008>.

- [18] P. BENNER AND P. GOYAL, *Balanced Truncation Model Order Reduction for Quadratic-bilinear Systems*, e-prints 1705.00160, <https://arxiv.org/abs/1705.00160>, 2017.
- [19] P. BENNER, P. GOYAL, AND S. GUGERCIN, \mathcal{H}_2 -quasi-optimal model order reduction for quadratic-bilinear control systems, *SIAM J. Matrix Anal. Appl.*, 39 (2018), pp. 983–1032, <https://doi.org/10.1137/16M1098280>.
- [20] P. BENNER, V. MEHRMANN, AND D. C. SORENSEN, *Dimension Reduction of Large-Scale Systems*, *Lect. Notes Comput. Sci. Eng.* 45, Springer, Berlin, 2005.
- [21] K. CARLBERG, C. FARHAT, J. CORTIAL, AND D. AMSALLEM, *The GNAT method for nonlinear model reduction: Effective implementation and application to computational fluid dynamics and turbulent flows*, *J. Comput. Phys.*, 242 (2013), pp. 623–647.
- [22] S. CHATURANTABUT AND D. C. SORENSEN, *Nonlinear model reduction via discrete empirical interpolation*, *SIAM J. Sci. Comput.*, 32 (2010), pp. 2737–2764, <https://doi.org/10.1137/090766498>.
- [23] S. CHELLAPPA, L. FENG, AND P. BENNER, *An adaptive sampling approach for the reduced basis method*, in *Realization and Model Reduction of Dynamical Systems – A Festschrift in Honor of 70th Birthday of Thanos Antoulas*, C. Beattie, P. Benner, M. Embree, S. Gugercin, and S. Lefteriu, eds., Springer, 2020, to appear.
- [24] P. FELDMANN AND R. W. FREUND, *Efficient linear circuit analysis by Padé approximation via the Lanczos process*, *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, 14 (1995), pp. 639–649.
- [25] G. FLAGG AND S. GUGERCIN, *Multipoint Volterra series interpolation and \mathcal{H}_2 optimal model reduction of bilinear systems*, *SIAM J. Matrix Anal. Appl.*, 36 (2015), pp. 549–579, <https://doi.org/10.1137/130947830>.
- [26] K. GALLIVAN, A. VANDENDORPE, AND P. VAN DOOREN, *Model reduction of MIMO systems via tangential interpolation*, *SIAM J. Matrix Anal. Appl.*, 26 (2004), pp. 328–349.
- [27] I. V. GOSEA AND A. C. ANTOULAS, *Data-driven model order reduction of quadratic-bilinear systems*, *Numer. Linear Algebra Appl.*, 25 (2018), e2200, <https://doi.org/10.1002/nla.2200>.
- [28] E. J. GRIMME, *Krylov Projection Methods for Model Reduction*, Ph.D. Thesis, University of Illinois at Urbana-Champaign, Urbana-Champaign, IL, 1997.
- [29] M. GUBISCH AND S. VOLKWEIN, *Proper orthogonal decomposition for linear-quadratic optimal control*, in *Model Reduction and Approximation: Theory and Algorithms*, P. Benner, A. Cohen, M. Ohlberger, and K. Willcox, eds., *Comput. Sci. Eng.* 15, SIAM, Philadelphia, 2017, pp. 3–63, <https://doi.org/10.1137/1.9781611974829.ch1>.
- [30] S. GUGERCIN, A. C. ANTOULAS, AND C. BEATTIE, \mathcal{H}_2 model reduction for large-scale linear dynamical systems, *SIAM J. Matrix Anal. Appl.*, 30 (2008), pp. 609–638, <https://doi.org/10.1137/060666123>.
- [31] S. GUGERCIN, T. STYKEL, AND S. WYATT, *Model reduction of descriptor systems by interpolatory projection methods*, *SIAM J. Sci. Comput.*, 35 (2013), pp. B1010–B1033, <https://doi.org/10.1137/130906635>.
- [32] G. GUIOCHON, A. FELINGER, AND D. G. SHIRAZI, *Fundamentals of preparative and nonlinear chromatography*, Academic Press, Boston, 2006.
- [33] T. G. KOLDA, *Multilinear Operators for Higher-Order Decompositions*, Technical report SAND2006-2081, Sandia National Laboratories, Albuquerque, NM, <https://doi.org/10.2172/923081>.
- [34] B. KRAMER AND K. WILLCOX, *Balanced Truncation Model Reduction for Lifted Nonlinear Systems*, preprint, <https://arxiv.org/abs/1907.12084>, 2019.
- [35] B. KRAMER AND K. WILLCOX, *Nonlinear model order reduction via lifting transformations and proper orthogonal decomposition*, *AIAA J.*, 57 (2019), pp. 2297–2307, <https://doi.org/10.2514/1.J057791>.
- [36] M. W. MAHONEY AND P. DRINEAS, *CUR matrix decompositions for improved data analysis*, *Proc. Natl. Acad. Sci.*, 106 (2009), pp. 697–702.
- [37] I. MAROS, *Computational Techniques of the Simplex Method*, *Internat. Ser. Oper. Res. Manag.* 61, Springer, New York, 2003.
- [38] A. J. MAYO AND A. C. ANTOULAS, *A framework for the solution of the generalized realization problem*, *Linear Algebra Appl.*, 425 (2007), pp. 634–662.
- [39] G. P. MCCORMICK, *Computability of global solutions to factorable nonconvex programs: Part I—Convex underestimating problems*, *Math. Program.*, 10 (1976), pp. 147–175.
- [40] S. MELCHIOR, V. LEGAT, AND P. VAN DOOREN, *Gramian based model reduction of nonlinear MIMO systems*, in *Mathematical Theory of Networks and Systems*, 2012, <http://www.mtns2012.conference.net.au/Extended%20Abstract/MTNS2012.0170-paper.pdf>. (Extended Abstract).

- [41] N. NGUYEN, A. T. PATERA, AND J. PERAIRE, *A best points interpolation method for efficient approximation of parametrized functions*, Internat. J. Numer. Methods Engrg., 73 (2008), pp. 521–543.
- [42] B. PEHERSTORFER, D. BUTNARU, K. WILLCOX, AND H.-J. BUNGARTZ, *Localized discrete empirical interpolation method*, SIAM J. Sci. Comput., 36 (2014), pp. A168–A192.
- [43] A. QUARTERONI, A. MANZONI, AND F. NEGRI, *Reduced Basis Methods for Partial Differential Equations*, Unitext 92, Springer, Milan, 2016.
- [44] A. QUARTERONI AND G. ROZZA, *Reduced Order Methods for Modeling and Computational Reduction*, MS&A. Model. Simul. Appl. 9, Springer, Cham, Switzerland, 2014.
- [45] W. J. RUGH, *Nonlinear System Theory*, Johns Hopkins University Press Baltimore, MD, 1981.
- [46] D. C. SORENSEN AND M. EMBREE, *A DEIM induced CUR factorization*, SIAM J. Sci. Comput., 38 (2016), pp. A1454–A1482.
- [47] P. VAN DOOREN, K. GALLIVAN, AND P.-A. ABSIL, *\mathcal{H}_2 -optimal model reduction of MIMO systems*, Appl. Math. Lett., 21 (2008), pp. 1267–1273, <https://doi.org/10.1016/j.aml.2007.09.015>.
- [48] D. C. VILLEMAGNE AND R. E. SKELTON, *Model reduction using a projection formulation*, Internat. J. Control, 46 (1987), pp. 2141–2169.
- [49] S. WANG AND Z. ZHANG, *Improving CUR matrix decomposition and the Nyström approximation via adaptive sampling*, J. Mach. Learn. Res., 14 (2013), pp. 2729–2769.
- [50] Y. XU AND T. ZENG, *Optimal \mathcal{H}_2 model reduction for large scale MIMO systems via tangential interpolation*, Int. J. Numer. Anal. Model., 8 (2011), pp. 174–188.