Toward Fitting Structured Nonlinear Systems by Means of Dynamic Mode Decomposition

Ion Victor Gosea and Igor Pontes Duff

Abstract The dynamic mode decomposition (DMD) is a data-driven method used for identifying the dynamics of complex nonlinear systems. It extracts important characteristics of the underlying dynamics using measured time-domain data produced either by means of experiments or by numerical simulations. In the original methodology, the measurements are assumed to be approximately related by a linear operator. Hence, a linear discrete-time system is fitted to the given data. However, often, nonlinear systems modeling physical phenomena have a particular known structure. In this contribution, we propose an identification and reduction method based on the classical DMD approach allowing to fit a structured nonlinear system to the measured data. We mainly focus on two types of nonlinearities: bilinear and quadratic bilinear. By enforcing this additional structure, more insight into extracting the nonlinear behavior of the original process is gained. Finally, we demonstrate the proposed methodology for different examples, such as Burgers' equation and the coupled van der Pol oscillators.

1 Introduction

Mathematical models are commonly used to simulate, optimize, and control the behavior of real dynamical processes. A common way to derive those models is to use the first principles, generally leading to a set of ordinary or partial differential equations. For high complex dynamics, fine discretization leads to high fidelity models, which require numerous equations and variables. In some situations, the high

I. V. Gosea (✉) · I. Pontes Duff
Max Planck Institute, Magdeburg, Germany
e-mail: gosea@mpi-magdeburg.mpg.de

I. Pontes Duff
e-mail: pontes@mpi-magdeburg.mpg.de

P. Benner et al. (eds.), *Model Reduction of Complex Dynamical Systems*,
International Series of Numerical Mathematics 171,
https://doi.org/10.1007/978-3-030-72983-7_3

model is given as a black box setup, i.e., by solvers that allow the computation of the full-model states for a given set of initial conditions and inputs, but does not provide the dynamical system realization. In order to better understand such dynamics processes, it is often beneficial to construct surrogate models using simulated data. This justifies the development of identification or data-driven model reduction methods. Indeed, with the ever-increasing availability of measured/simulated data in different scientific disciplines, the need for incorporating this information in the identification and reduction process has steadily grown. The data-driven model reduction problem consists of determining low-order models from the provided data obtained either by experimentation or numerical simulations. Methods such as Dynamic Mode Decomposition (DMD) have drawn considerable research endeavors.

DMD is a data-driven method for analyzing complex systems. The purpose is to learn/extract the important dynamic characteristics (such as unstable growth modes, resonance, and spectral properties) of the underlying dynamical system by means of measured time-domain data. These can be acquired through experiments in a practical setup or artificially through numerical simulations (by exciting the system). It was initially proposed in [29] in the context of analyzing numerical and experimental fluid mechanics problems. Additionally, it is intrinsically related to the Koopman operator analysis, see [22, 27]. Since its introduction, several extensions have been proposed in the literature, e.g., the exact DMD [31], the extended DMD [11], and the higher order DMD [20]. Also, in order to address control problems, DMD with control inputs was proposed in [25], and then extended to the case where outputs are also considered in [1, 9]. The reader is referred to [19] for a comprehensive monograph on the topic.

Often, nonlinear systems modeling physical phenomena have a particular known structure, such as bilinear and quadratic terms. In the present work, our primary goal is to embed nonlinear structures in the DMD framework. To this aim, we propose an identification and data-driven reduction method based on the classical DMD approach allowing to fit a bilinear and quadratic-bilinear structures to the measured data. The choice to fit such terms is due to the fact most systems with analytical nonlinearities (e.g., rational, trigonometrical, polynomial) can be exactly reformulated as quadratic-bilinear systems [15]. Our work is rooted in the two variants, DMD with control and input-output DMD, and can be considered as an extension of those methodologies.

There exist vast literature on learning nonlinear dynamics from data, and we review the most relevant literature for our work. One approach is the so-called Loewner framework, which enables to construct low-order models from frequency-domain data. It was initially proposed in [21], and later extended to bilinear [3] and quadratic-bilinear case [14]. Another approach is the operator inference, proposed [24]. This approach infers polynomial low-order models as a solution of a least-squares problem based on the initial conditions, inputs, and trajectories of the states. This approach was recently extended to systems with non-polynomials [8]. Also, the authors in [26] show how the use of lifting transformations can be beneficial to identify the system. Finally, the approach proposed in [23] introduces a method based on operator inference enabling to learn exactly the reduced models that are traditionally constructed with model reduction. It is worth mentioning that the operator inference approach [24] can be seen as an extension to DMD for nonlinear systems. Indeed, in this framework, the reduced-order model is allowed to have polynomial terms on the state and its matrices are obtained by solving a least-squares problems. The main difference is that this optimization problem is set using the reduced trajectories as the data (see the introduction of [24] for more details).

In this work, we aim at fitting nonlinear model structures using the DMD setup, i.e., by using the full-model trajectories, which is the main difference from [24]. Additionally, besides the quadratic structure on the state, we also consider reduced-order models having bilinear structure on the state and input.

The rest of the paper is organized as follows. Section 2 recalls some results on the classical DMD, DMD with control, and the input-output DMD. In Sect. 3, we present the main contribution of the paper, which is the incorporation of bilinear and quadratic-bilinear terms in the DMD setup. Finally, in Sect. 4, we demonstrate the proposed methodology for different examples, such as Burgers' equation and the coupled van der Pol oscillators.

2 Dynamic Mode Decomposition

In this section, we briefly recall the classical DMD framework [29]. To this aim, we analyze time-invariant systems of ordinary differential equations (ODEs) written compactly in a continuous-time setting as follows:

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t)), \tag{1}$$

where $\mathbf{x}(t) \in \mathbb{R}^n$ is the state vector and $f : \mathbb{R}^n \to \mathbb{R}^n$ is the system nonlinearity.

By means of sampling the variable $\mathbf{x}$ in (1) at uniform intervals of time, we collect a series of vectors $\mathbf{x}(t_k)$ for sampling times $t_0, t_1, \dots, t_m$. For simplicity, denote $\mathbf{x}_k := \mathbf{x}(t_k)$.

DMD aims at analyzing the relationship between pairs of measurements from a dynamical system. The measurements $\mathbf{x}_k$ and $\mathbf{x}_{k+1}$, as previously introduced, are assumed to be approximately related by a linear operator for all $k \in \{0, 1, \dots, m-1\}$.

$$\mathbf{x}_{k+1} \approx \mathbf{A}\mathbf{x}_k, \tag{2}$$

where $\mathbf{A} \in \mathbb{R}^{n \times n}$. This approximation is assumed to hold for all pairs of measurements. Next, group together the sequence of collected snapshots of the discretized state $\mathbf{x}(t)$ and use the following notations:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_0 \ \mathbf{x}_1 \ \dots \ \mathbf{x}_{m-1} \end{bmatrix} \in \mathbb{R}^{n \times m}, \quad \mathbf{X}_s = \begin{bmatrix} \mathbf{x}_1 \ \mathbf{x}_2 \ \dots \ \mathbf{x}_m \end{bmatrix} \in \mathbb{R}^{n \times m}. \tag{3}$$

The DMD method is based on finding a best-fit solution of an operator A so that the following relation is (approximately) satisfied

$$\mathbf{X}_s = \mathbf{A}\mathbf{X}, \tag{4}$$

which represents the block version of Eq. (2). Moreover, the above relation does not need to hold exactly. Previous work has theoretically justified using this approximating operator on data generated by nonlinear dynamical systems. For more details, see [30]. A best-fit solution is explicitly given as follows:

$$\mathbf{A} = \mathbf{X}_s\mathbf{X}^{\dagger}, \tag{5}$$

where $\mathbf{X}^{\dagger} \in \mathbb{R}^{m\times n}$ is the Moore-Penrose inverse of matrix $\mathbf{X} \in \mathbb{R}^{n\times m}$. In the above statement, by "best-fit" it is meant the solution that minimizes the least-squares error in the Frobenius norm (see [9]). More precisely, the matrix $\mathbf{A}$ in (5) is the solution of the following optimization problem:

$$\underset{\hat{\mathbf{A}}\in\mathbb{R}^{n\times n}}{\arg\min}\left(\|\mathbf{X}_s - \hat{\mathbf{A}}\mathbf{X}\|_{\mathrm{F}}\right). \tag{6}$$

The so-called DMD modes are given by the eigenvectors of matrix $\mathbf{A}$ in (5), collected in matrix $\mathbf{T}$ with $\mathbf{A} = \mathbf{T}\boldsymbol{\Lambda}\mathbf{T}^{-1}$. These spatial modes of system (1) are computed at a single frequency and are connected to the Koopman operator, see [22].

In this work, we will mainly focus on the construction of the reduced-order model rather than the evaluation of the DMD modes.

2.1 Dynamic Mode Decomposition with Control (DMDc)

Dynamic mode decomposition with control (DMDc) was introduced in [25] and it modifies the basic framework characterizing DMD. The novelty is given by including measurements of a control input $u(t) \in \mathbb{R}$. It is hence assumed that the dynamics of the original system of ODEs includes an input dependence, i.e.,

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), u(t)), \tag{7}$$

which represents a directs extension of (1). In (7), it is assumed that $f : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$. Then, continue as in the classical DMD case without control to collect a discretized solution $\mathbf{x}$ at particular time instances.

In this setup, a trio of measurements are now assumed to be connected. The goal of DMDc is to analyze the relationship between a future state measurement $\mathbf{x}_{k+1}$ with the current measurement $\mathbf{x}_k$ and the current control u_k.

The motivation for this method is that, understanding the dynamic characteristics of systems that have both internal dynamics and applied external control is of great use for many applications, such as for controller design and sensor placement.

The DMDc method is used to discover the underlying dynamics subject to a driving control input by quantifying its effect to the time-domain measurements corresponding to the underlying dynamical system.

A pair of linear operators represented by matrices $\mathbf{A} \in \mathbb{R}^{n\times n}$ and $\mathbf{B} \in \mathbb{R}^{n}$ provides the following dependence for each trio of measurement data snapshots $(\mathbf{x}_{k+1}, \mathbf{x}_k, \mathbf{u}_k)$

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}u_k, \ \ 0 \leq k \leq m-1. \tag{8}$$

Next, denote the sequence of control input snapshots with

$$\mathbf{U} = \begin{bmatrix} \mathbf{u}_0 \ \mathbf{u}_1 \ \ldots \ \mathbf{u}_{m-1} \end{bmatrix} \in \mathbb{R}^{1\times m}. \tag{9}$$

The first step is to augment the matrix $\mathbf{X}$ with the row vector $\mathbf{U}$ and similarly group together the $\mathbf{A}$ and $\mathbf{B}$ matrices by using the notations:

$$\mathbf{G} = [\mathbf{A} \ \ \mathbf{B}] \in \mathbb{R}^{n\times(n+1)}, \quad \boldsymbol{\Omega} = \begin{bmatrix} \mathbf{X} \\ \mathbf{U} \end{bmatrix} \in \mathbb{R}^{(n+1)\times m}. \tag{10}$$

The matrix $\mathbf{G}$ introduced above will be referred to as the system matrix since it incorporates the matrices corresponding to the system to be fitted.

By letting the index k vary in the range $\{0, 1, \ldots, m-1\}$, one can compactly rewrite the m equations in the following matrix format:

$$\mathbf{X}_s = \mathbf{A}\mathbf{X} + \mathbf{B}\mathbf{U} = [\mathbf{A} \ \ \mathbf{B}] \begin{bmatrix} \mathbf{X} \\ \mathbf{U} \end{bmatrix} := \mathbf{G}\boldsymbol{\Omega}. \tag{11}$$

Thus, similar to standard DMD, compute a pseudo-inverse and solve for $\mathbf{G}$ as

$$\mathbf{G} = \mathbf{X}_s\boldsymbol{\Omega}^{\dagger} \ \ \Rightarrow \ \ [\mathbf{A} \ \ \mathbf{B}] = \mathbf{X}_s \begin{bmatrix} \mathbf{X} \\ \mathbf{U} \end{bmatrix}^{\dagger}. \tag{12}$$

The matrix $\mathbf{G} \in \mathbb{R}^{n\times(n+1)}$ in (12) is actually the solution of the following optimization problem:

$$\underset{\hat{\mathbf{G}}\in\mathbb{R}^{n\times(n+1)}}{\arg\min} \left(\left\| \mathbf{X}_s - \hat{\mathbf{G}} \begin{bmatrix} \mathbf{X} \\ \mathbf{U} \end{bmatrix} \right\|_{\mathrm{F}} \right). \tag{13}$$

To explicitly compute the matrix in (12), we first find the singular value decomposition (SVD) of the augmented data matrix $\boldsymbol{\Omega}$ as follows

$$\boldsymbol{\Omega} = \mathbf{V}\boldsymbol{\Sigma}\mathbf{W}^T \approx \tilde{\mathbf{V}}\tilde{\boldsymbol{\Sigma}}\tilde{\mathbf{W}}^T, \tag{14}$$

where the full-scale and reduced-order matrices have the following dimensions:

$$\begin{cases} \mathbf{V} \in \mathbb{R}^{(n+1)\times(n+1)}, \quad \mathbf{\Sigma} \in \mathbb{R}^{(n+1)\times m}, \quad \mathbf{V} \in \mathbb{R}^{m\times m}, \\ \tilde{\mathbf{V}} \in \mathbb{R}^{(n+1)\times p}, \quad \tilde{\mathbf{\Sigma}} \in \mathbb{R}^{p\times p}, \quad \tilde{\mathbf{V}} \in \mathbb{R}^{m\times r}. \end{cases}$$

The truncation index is denoted with p, where $p \leqslant n$. The pseudo-inverse $\mathbf{\Omega}^{\dagger}$ is computed using the matrices from the SVD in (14), i.e., as $\mathbf{\Omega}^{\dagger} \approx \tilde{\mathbf{W}}\tilde{\mathbf{\Sigma}}^{-1}\tilde{\mathbf{V}}^{T}$.

By splitting up the matrix $\mathbf{V}^{T}$ as $\tilde{\mathbf{V}}^{T} = [\tilde{\mathbf{V}}_{1}^{T} \ \ \tilde{\mathbf{V}}_{2}^{T}]$, recover the system matrices as

$$\overline{\mathbf{A}} = \mathbf{X}_{s}\tilde{\mathbf{W}}\tilde{\mathbf{\Sigma}}^{-1}\tilde{\mathbf{V}}_{1}^{T}, \ \ \overline{\mathbf{B}} = \mathbf{X}_{s}\tilde{\mathbf{W}}\tilde{\mathbf{\Sigma}}^{-1}\tilde{\mathbf{V}}_{2}^{T}. \tag{15}$$

As mentioned in, there is one additional step. By performing another (short) SVD of the matrix $\mathbf{X}_{s}$, write

$$\mathbf{X}_{s} \approx \hat{\mathbf{V}}\hat{\mathbf{\Sigma}}\hat{\mathbf{W}}^{T}, \tag{16}$$

where $\hat{\mathbf{V}} \in \mathbb{R}^{(n+1)\times r}, \quad \hat{\mathbf{\Sigma}} \in \mathbb{R}^{r\times r}, \quad \hat{\mathbf{V}} \in \mathbb{R}^{m\times r}$. Note that the two SVDs will likely have different truncation values. The following reduced-order approximations of $\mathbf{A}$ and $\mathbf{B}$ are hence computed as

$$\tilde{\mathbf{A}} = \hat{\mathbf{V}}^{T}\overline{\mathbf{A}}\hat{\mathbf{V}} = \hat{\mathbf{V}}^{T}\mathbf{X}_{s}\tilde{\mathbf{W}}\tilde{\mathbf{\Sigma}}^{-1}\tilde{\mathbf{V}}_{1}^{T}\hat{\mathbf{V}} \in \mathbb{R}^{r\times r}, \ \tilde{\mathbf{B}} = \hat{\mathbf{V}}^{T}\overline{\mathbf{B}} = \hat{\mathbf{V}}^{T}\mathbf{X}_{s}\tilde{\mathbf{W}}\tilde{\mathbf{\Sigma}}^{-1}\tilde{\mathbf{V}}_{2}^{T} \in \mathbb{R}^{r}. \tag{17}$$

2.2 Input-Output Dynamic Mode Decomposition

In this section, we discuss the technique proposed in [1] known as input-output dynamic mode decomposition (ioDMD). This method constructs an input-output reduced-order model and can be viewed as an extension of DMDc for the case with observed outputs. As stated in the original work [1], this method represents a combination of POD and system identification techniques. The proposed method discussed here is similar in a sense to the algorithms for subspace state-space system identification (N4SID) introduced in [32] and can be also applied to large-scale systems.

We consider as given a system of ODEs whose dynamics is described by the same equations as in (7). Additionally, assume that observations are collected in the variable $y(t) \in \mathbb{R}$, as function of the state variable $\mathbf{x}$ and of the control u, written as

$$y(t) = g(\mathbf{x}(t), u(t)), \tag{18}$$

where $g : \mathbb{R}^{n} \times \mathbb{R} \rightarrow \mathbb{R}$.

As before, the next step is to collect snapshots of both variable $\mathbf{x}(t)$ and of the output $y(t)$ sampled at some positive time instances $t_0, t_1, \ldots t_{m-1}$. Again, for simplicity of the exposition, denote with $y_k := y(t_k)$.

We enforce the following dependence for each trio of measurement data snapshots given by $(\mathbf{y}_k, \mathbf{x}_k, \mathbf{u}_k)$

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \mathbf{D}u_k, \quad 0 \leq k \leq m-1. \tag{19}$$

Afterward, collect the output values in a row vector as follows:

$$\mathbf{Y} = \begin{bmatrix} \mathbf{y}_0 \; \mathbf{y}_1 \; \dots \; \mathbf{y}_{m-1} \end{bmatrix} \in \mathbb{R}^{1\times m}. \tag{20}$$

The ioDMD method aims at fitting the given set of snapshot measurements collected in matrices $\mathbf{X}_s$, $\mathbf{X}$ and vectors $\mathbf{U}$ and $\mathbf{Y}$ to a linear discrete-time system characterized by the following equations:

$$\begin{aligned} \mathbf{X}_s &= \mathbf{A}\mathbf{X} + \mathbf{B}\mathbf{U}, \\ \mathbf{Y} &= \mathbf{C}\mathbf{X} + \mathbf{D}\mathbf{U}, \end{aligned} \tag{21}$$

where, as before, $\mathbf{A} \in \mathbb{R}^{n\times n}$ and $\mathbf{B} \in \mathbb{R}^n$, and also $\mathbf{C}^T \in \mathbb{R}^n$, $\mathbf{D} \in \mathbb{R}$. Note that the first equation in (21) exactly corresponds to the driving matrix equation of DMDc presented in (12). Moreover, write the system of equations in (21) compactly as

$$\begin{bmatrix} \mathbf{X}_s \\ \mathbf{Y} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ \mathbf{U} \end{bmatrix}. \tag{22}$$

Next, we adapt the definition of the system matrix $\mathbf{G}$ from (10) by incorporating an extra line as follows:

$$\mathbf{G} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \in \mathbb{R}^{(n+1)\times(n+1)}, \tag{23}$$

while $\mathbf{\Omega} = \begin{bmatrix} \mathbf{X} \\ \mathbf{U} \end{bmatrix} \in \mathbb{R}^{(n+1)\times m}$ is as before. Introduce a new notation that will become useful also in the next sections. It represents an augmentation of the shifted state matrix $\mathbf{X}_s$ with the output observation vector $\mathbf{Y}$, i.e.,

$$\mathbf{\Gamma} = \begin{bmatrix} \mathbf{X}_s \\ \mathbf{Y} \end{bmatrix} \in \mathbb{R}^{(n+1)\times m}. \tag{24}$$

Again, the solution of Eq. (22) will be computed as a best-fit type of approach. Hence, similarly to the DMDc case, recover the matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, and $\mathbf{D}$ by computing the pseudo-inverse of matrix $\mathbf{\Omega}$ and writing

$$\mathbf{G} = \mathbf{\Gamma}\mathbf{\Omega}^\dagger \;\; \Rightarrow \;\; \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} = \begin{bmatrix} \mathbf{X}_s \\ \mathbf{Y} \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ \mathbf{U} \end{bmatrix}^\dagger. \tag{25}$$

The matrix $\mathbf{G} \in \mathbb{R}^{(n+1)\times(n+1)}$ in (12) is actually the solution of the following optimization problem:

$$\underset{\hat{\mathbf{G}}\in\mathbb{R}^{(n+1)\times(n+1)}}{\arg\min} \left(\left\| \begin{bmatrix} \mathbf{X}_s \\ \mathbf{Y} \end{bmatrix} - \hat{\mathbf{G}} \begin{bmatrix} \mathbf{X} \\ \mathbf{U} \end{bmatrix} \right\|_{\mathrm{F}} \right). \tag{26}$$

Similar to the procedure covered in Sect. 2.1, one could further lower the dimension of the recovered system matrices by employing an additional SVD of the matrix $\mathbf{\Gamma}$, as was done in (16).

3 The Proposed Extensions

In this section, we present the main contribution of the paper. We propose extensions of the methods previously introduced in Sects. 2.1 and 2.2, e.g., DMDc and, respectively, ioDMD to fit nonlinear structured systems. More specifically, the discrete-time models that are fitted using these procedures will no longer be linear as in (21); the new models will contain nonlinear (bilinear or quadratic) terms.

3.1 Bilinear Systems

Bilinear systems are a class of mildly nonlinear systems for which the nonlinearity is given by the product between the state variable and the control input. More exactly, the characterizing system of ODEs is written as in (7) but for a specific choice of mapping f, i.e., $f(\mathbf{x}, \mathbf{u}) = \mathbf{A}\mathbf{x} + \mathbf{N}\mathbf{x}u + \mathbf{B}u$. Additionally, assume that the observed output y depends linearly on the state $\mathbf{x}$. Hence, in what follows, we will make use of the following description of bilinear systems (with a single input and a single output):

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{N}\mathbf{x}(t)u(t) + \mathbf{B}u(t), \\ y(t) &= \mathbf{C}\mathbf{x}(t), \end{aligned} \tag{27}$$

where the matrix $\mathbf{N} \in \mathbb{R}^{n\times n}$ scales the product of the state variable $\mathbf{x}$ with the control input u. In practice, bilinear control systems are used to approximate nonlinear systems with more general, analytic nonlinearities. This procedure is known as Carleman's linearization; for more details see [28].

Bilinear systems are a class of nonlinear systems that received considerable attention in the last four or five decades. Contributions that range from realization theory in [16], classical system identification in [12], or to subspace identification in [13]. In more recent years (last two decades), model order reduction of bilinear systems (in both continuous- and discrete-time domains) was extensively studied with contributions covering balanced truncation in [33], Krylov subspace methods in [10], interpolation-based $\mathcal{H}_2$ method in [4, 6], or data-driven Loewner approach in [3, 17].

3.1.1 The General Procedure

We start by collecting snapshots of the state $\mathbf{x}$ for multiple time instances t_k. We enforce that the snapshot $\mathbf{x}_{k+1}$ at time t_{k+1} depends on the snapshot $\mathbf{x}_k$ in the following way:

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{N}\mathbf{x}_k u_k + \mathbf{B}u_k, \quad \text{for } 0 \le k \le m-1. \tag{28}$$

We denote the sequence of state and input snapshots as in (3) and in (9). Again, by varying the index k in the interval $\{1, 2, \dots, m-1\}$, one can compactly rewrite the $m-1$ equations in the following matrix format:

$$\mathbf{X}_s = \mathbf{A}\mathbf{X} + \mathbf{N}\mathbf{X}\mathbf{U_D} + \mathbf{B}\mathbf{U}, \tag{29}$$

where $\mathbf{U_D} = \text{diag}(u_0, u_1, \dots, u_{m-1}) \in \mathbb{R}^{m\times m}$. One can hence write $\mathbf{U} = \mathbf{L}\mathbf{U_D}$, with $\mathbf{L} = [1\ 1\ \dots\ 1] \in \mathbb{R}^{1\times m}$ and then introduce the matrix $\mathbf{Z} \in \mathbb{R}^{(n+1)\times m}$ as

$$\mathbf{Z} = \begin{bmatrix} \mathbf{L}\mathbf{U_D} \\ \mathbf{X}\mathbf{U_D} \end{bmatrix} = \begin{bmatrix} \mathbf{L} \\ \mathbf{X} \end{bmatrix} \mathbf{U_D}. \tag{30}$$

The next step is to augment the matrix $\mathbf{X}$ with matrix $\mathbf{Z}$ and denote this new matrix with $\boldsymbol{\Omega} \in \mathbb{R}^{(2n+1)\times m}$ as an extension of the matrix previously introduced in (10), i.e.,

$$\boldsymbol{\Omega} = \begin{bmatrix} \mathbf{X} \\ \mathbf{Z} \end{bmatrix}. \tag{31}$$

For the case in which we extend the DMDc method in Sect. 2.1 to fitting bilinear dynamics (no output observations), we propose a slightly different definition for the matrix $\mathbf{G}$. We hence append the matrix $\mathbf{N}$ to the originally introduced system matrix in (10). Then, Eq. (29) can be written in a factorized way as $\boldsymbol{\Gamma} = \mathbf{G}\boldsymbol{\Omega}$, where the matrices for this particular setup are as follows:

$$\mathbf{G} = \begin{bmatrix} \mathbf{A} & \mathbf{B} & \mathbf{N} \end{bmatrix} \in \mathbb{R}^{n\times(2n+1)}, \quad \boldsymbol{\Gamma} = \mathbf{X}_s. \tag{32}$$

Alternatively, for the case where output observations y_k are also available, we enforce a special bilinear dependence for each trio of measurement data snapshots as

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \mathbf{F}\mathbf{x}_k u_k + \mathbf{D}u_k, \quad 0 \le k \le m-1, \tag{33}$$

where $\mathbf{F}^T \in \mathbb{R}^n$. Note that (33) represents a natural extension of the relation imposed in (19). Therefore, fitting a linear structure is instead enforced.

Afterward, we collect the equations in (33) for each index k and hence write

$$\mathbf{Y} = \mathbf{C}\mathbf{X} + \mathbf{F}\mathbf{X}\mathbf{U}_{\mathbf{D}} + \mathbf{D}\mathbf{U}, \tag{34}$$

with the same notations as in (30). Then, by combining (29) and (34), we can write all snapshot matrix quantities in the following structured equalities:

$$\begin{aligned} \mathbf{X}_s &= \mathbf{A}\mathbf{X} + \mathbf{N}\mathbf{X}\mathbf{U}_{\mathbf{D}} + \mathbf{B}\mathbf{U}, \\ \mathbf{Y} &= \mathbf{C}\mathbf{X} + \mathbf{F}\mathbf{X}\mathbf{U}_{\mathbf{D}} + \mathbf{D}\mathbf{U}. \end{aligned} \tag{35}$$

This system of equations can be then written in a factorized way as before, i.e., $\boldsymbol{\Gamma} = \mathbf{G}\boldsymbol{\Omega}$, where the matrices for this particular setup are given below:

$$\mathbf{G} = \begin{bmatrix} \mathbf{A} & \mathbf{B} & \mathbf{N} \\ \mathbf{C} & \mathbf{D} & \mathbf{F} \end{bmatrix} \in \mathbb{R}^{(n+1)\times(2n+1)}, \quad \boldsymbol{\Gamma} = \begin{bmatrix} \mathbf{X}_s \\ \mathbf{Y} \end{bmatrix}. \tag{36}$$

Finally, the last step is to recover the matrix $\mathbf{G}$ and split it block-wise in order to put together a system realization. Consequently, this all boils down to solving the equation $\boldsymbol{\Gamma} = \mathbf{G}\boldsymbol{\Omega}$ (in either of the two cases, with or without output observations included). More precisely, the objective matrix $\mathbf{G} \in \mathbb{R}^{(n+1)\times(2n+1)}$ in (36) is the solution of the following optimization problem:

$$\underset{\hat{\mathbf{G}}\in\mathbb{R}^{(n+1)\times(2n+1)}}{\arg\min} \left(\left\| \begin{bmatrix} \mathbf{X}_s \\ \mathbf{Y} \end{bmatrix} - \hat{\mathbf{G}} \begin{bmatrix} \mathbf{X} \\ \mathbf{Z} \end{bmatrix} \right\|_{\mathrm{F}} \right) \Leftrightarrow \underset{\hat{\mathbf{G}}\in\mathbb{R}^{(n+1)\times(2n+1)}}{\arg\min} \left(\left\| \boldsymbol{\Gamma} - \hat{\mathbf{G}}\boldsymbol{\Omega} \right\|_{\mathrm{F}} \right). \tag{37}$$

As shown in the previous sections, solving for $\mathbf{G}$ in (37) involves computing the pseudo-inverse of matrix $\boldsymbol{\Omega} \in \mathbb{R}^{(2n+1)\times m}$ from (31). More precisely, we write the solution as

$$\mathbf{G} = \boldsymbol{\Gamma}\boldsymbol{\Omega}^{\dagger}. \tag{38}$$

Remark 1 Note that the observation map g corresponding to the original dynamical system, as introduced in (18), need not have a bilinear structure as in (33). It could include more complex nonlinearities or could even be linear. In the later case, the recovered matrix $\mathbf{F}$ will typically have a low norm.

3.1.2 Computation of the Reduced-Order Matrices

In this section, we present specific/practical details for retrieving the system matrices in the case of the proposed procedure in Sect. 3.1.1. We solve the equation $\boldsymbol{\Gamma} = \mathbf{G}\boldsymbol{\Omega}$ for which the matrices are given as in (36), i.e., the case containing output observations. We compute an SVD of the augmented data matrix $\boldsymbol{\Omega}$ giving

$$\boldsymbol{\Omega} = \mathbf{V}\boldsymbol{\Sigma}\mathbf{W}^T \approx \tilde{\mathbf{V}}\tilde{\boldsymbol{\Sigma}}\tilde{\mathbf{W}}^T, \tag{39}$$

where the full-scale and reduced-scale matrices derived from SVD are as follows:

$$\begin{cases} \mathbf{V} \in \mathbb{R}^{(2n+1)\times(2n+1)}, & \boldsymbol{\Sigma} \in \mathbb{R}^{(2n+1)\times(m-1)}, \quad \mathbf{V} \in \mathbb{R}^{(m-1)\times(m-1)}, \\ \tilde{\mathbf{V}} \in \mathbb{R}^{(2n+1)\times p}, & \tilde{\boldsymbol{\Sigma}} \in \mathbb{R}^{p\times p}, \quad \tilde{\mathbf{V}} \in \mathbb{R}^{(m-1)\times p}. \end{cases}$$

The truncation index is denoted with p, where $p \leqslant n$. The computation of the pseudo-inverse $\boldsymbol{\Omega}^{\dagger}$ is done by the SVD approach, i.e., $\boldsymbol{\Omega}^{\dagger} \approx \tilde{\mathbf{W}}\tilde{\boldsymbol{\Sigma}}^{-1}\tilde{\mathbf{V}}^{T}$. By splitting up the $\mathbf{V}^T$ matrix as $\tilde{\mathbf{V}}^T = [\tilde{\mathbf{V}}_1^T \;\; \tilde{\mathbf{V}}_2^T \;\; \tilde{\mathbf{V}}_3^T]$, one can recover the system matrices as

$$\begin{aligned} &\overline{\mathbf{A}} = \mathbf{X}_s\tilde{\mathbf{W}}\tilde{\boldsymbol{\Sigma}}^{-1}\tilde{\mathbf{V}}_1^T, \quad \overline{\mathbf{B}} = \mathbf{X}_s\tilde{\mathbf{W}}\tilde{\boldsymbol{\Sigma}}^{-1}\tilde{\mathbf{V}}_2^T, \quad \overline{\mathbf{N}} = \mathbf{X}_s\tilde{\mathbf{W}}\tilde{\boldsymbol{\Sigma}}^{-1}\tilde{\mathbf{V}}_3^T, \\ &\overline{\mathbf{C}} = \mathbf{Y}\tilde{\mathbf{W}}\tilde{\boldsymbol{\Sigma}}^{-1}\tilde{\mathbf{V}}_1^T, \quad \overline{\mathbf{D}} = \mathbf{Y}\tilde{\mathbf{W}}\tilde{\boldsymbol{\Sigma}}^{-1}\tilde{\mathbf{V}}_2^T, \quad \overline{\mathbf{F}} = \mathbf{Y}\tilde{\mathbf{W}}\tilde{\boldsymbol{\Sigma}}^{-1}\tilde{\mathbf{V}}_3^T. \end{aligned} \tag{40}$$

By performing another (short) SVD for the matrix $\mathbf{X}_s$, we can write

$$\mathbf{X}_s \approx \hat{\mathbf{V}}\hat{\boldsymbol{\Sigma}}\hat{\mathbf{W}}^T, \tag{41}$$

where $\hat{\mathbf{V}} \in \mathbb{R}^{(n+1)\times r}$, $\hat{\boldsymbol{\Sigma}} \in \mathbb{R}^{r\times r}$, $\hat{\mathbf{W}} \in \mathbb{R}^{(m-1)\times p}$. Note that the two SVDs could have different truncation values denoted with p and r. Using the transformation $\mathbf{x} = \hat{\mathbf{V}}\tilde{\mathbf{x}}$, the following reduced-order matrices can be computed:

$$\begin{aligned} &\tilde{\mathbf{A}} = \hat{\mathbf{V}}^T\overline{\mathbf{A}}\hat{\mathbf{V}} \in \mathbb{R}^{r\times r}, \quad \tilde{\mathbf{B}} = \hat{\mathbf{V}}^T\overline{\mathbf{B}} \in \mathbb{R}^{r}, \quad \tilde{\mathbf{N}} = \hat{\mathbf{V}}^T\overline{\mathbf{N}}\hat{\mathbf{V}} \in \mathbb{R}^{r\times r}, \\ &\tilde{\mathbf{C}} = \overline{\mathbf{C}}\hat{\mathbf{V}} \in \mathbb{R}^{1\times r}, \quad \tilde{\mathbf{D}} = \overline{\mathbf{D}} \in \mathbb{R}, \quad \tilde{\mathbf{F}} = \hat{\mathbf{V}}^T\overline{\mathbf{F}}\hat{\mathbf{V}} \in \mathbb{R}^{1\times r}. \end{aligned} \tag{42}$$

3.1.3 Conversions Between Discrete-Time and Continuous-Time Representations

The DMD-type approaches available in the literature identify continuous-time systems by means of linear discrete-time models. In this contribution, we make use of the same philosophy, in the sense that the models fitted are discrete time. We extend the DMDc and ioDMD approaches by allowing bilinear or quadratic terms to appear in these models as well.

As also mentioned in [9], one can compute a continuous-time model that represents a first-order approximation of the discrete-time model obtained by DMD-type approaches.

Assume that we are in the bilinear setting presented in Sect. 3.1 and that we already have computed a reduced-order discrete-time model given by matrices $\{\tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{N}}, \tilde{\mathbf{C}}, \tilde{\mathbf{D}}, \tilde{\mathbf{F}}\}$, i.e., following the explicit derivations in (42). Then, a continuous-time model $\{\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{N}}, \hat{\mathbf{C}}, \hat{\mathbf{D}}, \hat{\mathbf{F}}\}$ can also be derived. By assuming that the standard first-order Euler method was used for simulating the original system (with a small enough time step size $0 < \Delta_t \ll 1$), we can write that

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{x}_k + \Delta_t(\hat{\mathbf{A}}\mathbf{x}_k + \hat{\mathbf{B}}\mathbf{u}_k + \hat{\mathbf{N}}\mathbf{x}_k\mathbf{u}_k) \Rightarrow \\ \tilde{\mathbf{A}}\mathbf{x}_k + \tilde{\mathbf{B}}\mathbf{u}_k + \tilde{\mathbf{N}}\mathbf{x}_k\mathbf{u}_k &= \mathbf{x}_k + \Delta_t\hat{\mathbf{A}}\mathbf{x}_k + \Delta_t\hat{\mathbf{B}}\mathbf{u}_k + \Delta_t\hat{\mathbf{N}}\mathbf{x}_k\mathbf{u}_k \Rightarrow \\ &\begin{cases} \hat{\mathbf{A}} = \Delta_t^{-1}(\tilde{\mathbf{A}} - \mathbf{I}), \quad \hat{\mathbf{B}} = \Delta_t^{-1}\hat{\mathbf{B}}, \quad \hat{\mathbf{N}} = \Delta_t^{-1}\hat{\mathbf{N}}, \\ \hat{\mathbf{C}} = \tilde{\mathbf{C}}, \quad \hat{\mathbf{D}} = \tilde{\mathbf{D}}, \quad \hat{\mathbf{F}} = \tilde{\mathbf{F}}. \end{cases} \end{aligned} \tag{43}$$

Observe that for the ioDMD type of approaches, the feed-through terms that appear in the output-state equation are the same in both discrete and continuous representations.

3.2 Quadratic-Bilinear Systems

Next, we extend the method in Sect. 2.1 for fitting another class of nonlinear systems, i.e., quadratic-bilinear (QB) systems. Additional to the bilinear terms that enter the differential equations, we assume that quadratic terms are also present. More precisely, the system of the ODEs is written as in (7) but for a specific choice of nonlinear mapping f, i.e.,

$$f(\mathbf{x}, \mathbf{u}) = \mathbf{A}\mathbf{x} + \mathbf{Q}(\mathbf{x} \otimes \mathbf{x}) + \mathbf{N}\mathbf{x}u + \mathbf{B}u,$$

where "$\otimes$" denotes the Kronecker product, the matrix $\mathbf{Q} \in \mathbb{R}^{n \times n^2}$ scales the product of the state $\mathbf{x}$ with itself, and $\mathbf{N} \in \mathbb{R}^{n \times n}$ is as shown in Sect. 3.1.

Quadratic-bilinear systems appear in many applications for which the original system of ODEs inherently has the required quadratic structure. For example, after semi-discretizing Burgers' or Navier-Stokes equations in the spatial domain, one obtains a system of differential equations with quadratic nonlinearities (and also with bilinear terms). Moreover, many smooth analytic nonlinear systems that contain combinations of nonlinearities such as exponential, trigonometric, polynomial functions, etc. can be equivalently rewritten as QB systems. This is performed by employing so-called lifting techniques. More exactly, one needs to introduce new state variables in order to simplify the nonlinearities and hence derive new differential equations corresponding to these variables. Model order reduction of QB systems was a topic of interest in the last years with contributions ranging from projection-based approaches in [5, 15] to optimal $\mathcal{H}_2$-based approximation in [7], or data-driven approaches in the Loewner framework in [2, 14].

Similar to the procedure described in Sect. 3.1, we enforce that the snapshot $\mathbf{x}_{k+1}$ at time t_{k+1} depends on the snapshot $\mathbf{x}_k$ in the following way:

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{Q}(\mathbf{x}_k \otimes \mathbf{x}_k) + \mathbf{N}\mathbf{x}_k u_k + \mathbf{B}u_k, \quad \text{for } 0 \leq k \leq m-1. \tag{44}$$

Next, by varying the k in the range $\{1, 2, \ldots, m-1\}$, compactly rewrite the m equations in (44) in the following matrix format:

$$\mathbf{X}_s = \mathbf{A}\mathbf{X} + \mathbf{Q}(\mathbf{X} \otimes \mathbf{X})\mathbf{H} + \mathbf{N}\mathbf{X}\mathbf{U}_\mathbf{D} + \mathbf{B}\mathbf{U}, \tag{45}$$

with $\mathbf{U_D} = \text{diag}(u_0, u_1, \ldots, u_{m-1}) \in \mathbb{R}^{m\times m}$ and $\mathbf{H} = \left[\mathbf{e}_1 \otimes \mathbf{e}_1 \; \mathbf{e}_2 \otimes \mathbf{e}_2 \; \ldots \; \mathbf{e}_m \otimes \mathbf{e}_m\right] \in \mathbb{R}^{m^2\times m}$. Here, $\mathbf{e}_k$ is the unit vector of length n that contains the 1 on position k. Additionally, we introduce the matrix $\mathbf{T}$ that depends on the state matrix $\mathbf{X}$ as

$$\mathbf{T} = \left[\mathbf{X}_1 \otimes \mathbf{X}_1 \; \mathbf{X}_2 \otimes \mathbf{X}_2 \; \ldots \; \mathbf{X}_m \otimes \mathbf{X}_m\right] \in \mathbb{R}^{n^2\times m}.$$

Note that the equality holds as follows $\mathbf{T} = (\mathbf{X} \otimes \mathbf{X})\mathbf{H}$. Next, we augment the matrix $\mathbf{X}$ with both matrices $\mathbf{Z}$ and $\mathbf{T}$ and group together the matrices $\mathbf{A}$, $\mathbf{Q}$, $\mathbf{N}$ and $\mathbf{B}$ by using the notations:

$$\mathbf{G} = [\mathbf{A}\; \mathbf{B}\; \mathbf{N}\; \mathbf{Q}] \in \mathbb{R}^{n\times(n^2+2n+1)}, \quad \boldsymbol{\Omega} = \begin{bmatrix} \mathbf{X} \\ \mathbf{Z} \\ \mathbf{T} \end{bmatrix} \in \mathbb{R}^{(n^2+2n+1)\times m}, \quad \boldsymbol{\Gamma} = \mathbf{X}_s. \tag{46}$$

Hence, by using the above notations, rewrite Eq. (45) as follows: $\boldsymbol{\Gamma} = \mathbf{G}\boldsymbol{\Omega}$.

More precisely, the objective matrix $\mathbf{G} \in \mathbb{R}^{n\times(n^2+2n+1)}$ in (46) is the solution of the following optimization problem:

$$\underset{\hat{\mathbf{G}}\in\mathbb{R}^{n\times(n^2+2n+1)}}{\arg\min} \left(\left\|\boldsymbol{\Gamma} - \hat{\mathbf{G}}\boldsymbol{\Omega}\right\|_{\text{F}}\right). \tag{47}$$

Thus, one can recover the matrix $\mathbf{G}$ by solving an optimization problem, e.g., the one given in (47). This is explicitly done by computing the Moore-Pseudo pseudo-inverse of matrix $\boldsymbol{\Omega} \in \mathbb{R}^{(n^2+2n+1)\times m}$, and then writing $\mathbf{G} = \boldsymbol{\Gamma}\boldsymbol{\Omega}^\dagger$.

As previously shown in Sect. 3.1, we can again adapt the procedure for fitting QB systems in the ioDMD format by involving output observation measurements y_k. The procedure for quadratic-bilinear systems is similar to that for bilinear systems and we prefer to skip the exact description to avoid duplication. For more details, see the derivation in Sect. 6.1.

Remark 2 Note that the Kronecker product of the vector $\mathbf{x} \in \mathbf{R}^n$ with itself, i.e., $\mathbf{x}^{(2)} = \mathbf{x} \otimes \mathbf{x}$ has indeed duplicate components. For $n = 2$, one can write

$$\mathbf{x}^{(2)} = \left[x_1^2 \; x_1x_2 \; x_2x_1 \; x_2^2\right]^T.$$

Thus, since matrix $\mathbf{G}$ is explicitly written in terms of $\mathbf{Q}$ as in (46), the linear system of equations $\boldsymbol{\Gamma} = \mathbf{G}\boldsymbol{\Omega}$ does not have an unique solution. By using the Moore-Penrose inverse, one implicitly regularizes the least-squares problem in (47). Additionally, note that using a different least-squares solver (with or without regularization) could indeed produce a different result.

Remark 3 It is to be noted that the operator inference procedure avoids the non-uniqueness by accounting for duplicates in the vector $\mathbf{x} \otimes \mathbf{x}$. This is done by introducing a special Kronecker product for which the duplicate terms are removed. For more details, we refer the reader to Sect. 2.3 from [8].

4 Numerical Experiments

4.1 The Viscous Burgers' Equation

Consider the partial differential viscous Burgers' equation:

$$\frac{\partial v(x,t)}{\partial t} + v(x,t)\frac{\partial v(x,t)}{\partial x} = \nu \frac{\partial^2 v(x,t)}{\partial x^2}, \quad (x,t) \in (0,L) \times (0,T)\ ,$$

with i.c. $v(x,0) = 0,\ x \in [0,L],\ v(0,t) = u(t),\ v(L,t) = 0,\ t \geqslant 0$. The viscosity parameter is denoted with ν.

Burgers' equation has a convective term, an unsteady term and a viscous term; it can be viewed as a simplification of the Navier-Stokes equations.

By means of semi-discretization in the space domain, one can obtain the following nonlinear (quadratic) model (see [5]) described by the following system of ODEs:

$$\dot{v}_k = \begin{cases} -\frac{1}{2h}v_1v_2 + \frac{\nu}{h^2}(v_2 - 2v_1) + (\frac{v_1}{2h} + \frac{\nu}{h^2})u, & k = 1, \\ -\frac{v_k}{2h}(v_{k+1} - v_{k-1}) + \frac{\nu}{h^2}(v_{k+1} - 2v_k + v_{k-1}), & 2 \leqslant k \leqslant n_0 - 1, \\ -\frac{1}{2h}v_nv_{n-1} + \frac{\nu}{h^2}(-2v_n + 2v_{n-1}), & k = n_0. \end{cases} \tag{48}$$

Next, by means of the Carleman linearization procedure in [28], one can approximate the above nonlinear system of order n_0 with a bilinear system of order $n = n_0^2 + n_0$. The procedure is as follows: let $\mathbf{v} = [v_1\ v_2\ \dots\ v_n]^T$ be the original state variable in (48). Then, introduce the augmented state variable $\mathbf{x} = \begin{bmatrix} \mathbf{v} \\ \mathbf{v} \otimes \mathbf{v} \end{bmatrix} \in \mathbb{R}^{n_0^2+n_0}$ corresponding to the system described by the following equations:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{N}\mathbf{x}u + \mathbf{B}u, \\ \mathbf{y} &= \mathbf{C}\mathbf{x}. \end{aligned} \tag{49}$$

The continuous-time bilinear model in (49) is going to be used in following the numerical experiments.

Start by choosing the viscosity parameter to be $\nu = 0.01$. Then, choose $n_0 = 40$ as the dimension of the original discretization, and hence the bilinear system in (49) is of order $n = 1640$. Perform a time-domain simulation of this system by approximating the derivative as follows $\dot{\mathbf{x}}(t_k) \approx \frac{\mathbf{x}(t_{k+1}) - \mathbf{x}(t_k)}{t_{k+1} - t_k} = \frac{\mathbf{x}_{k+1} - \mathbf{x}_k}{\Delta_t}$. We use as time step $\delta_t = 10^{-3}$ and the time horizon to be $[0, 10]s$. The control input is chosen to be $u(t) = 0.5\cos(10t)e^{-0.3t}$.

Hence, collect 10^4 snapshots of the trio $(\mathbf{x}_k, u_k, y_k)$ that are arranged in the required matrix format as presented in the previous sections. The first step is to perform an SVD for the matrix $\mathbf{\Omega} \in \mathbb{R}^{3281 \times 10^4}$. The first 200 normalized singular values are presented in Fig. 1. Choose the tolerance value $\tau_p = 10^{-10}$ which corresponds to a truncation order of $p = 86$ (for computing the pseudo-inverse of matrix $\mathbf{\Omega}$). On the same plot

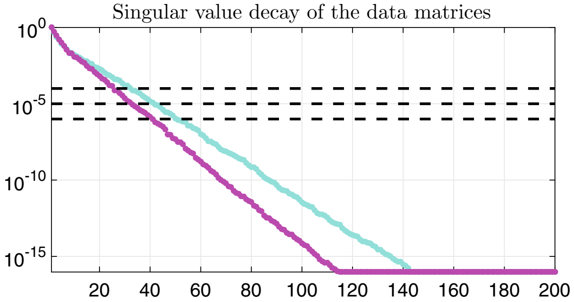

Fig. 1 The normalized first 200 singular values of matrices $\mathbf{\Omega}$ (with cyan) and $\mathbf{\Gamma}$ (with magenta). The three dotted black lines correspond to the three tolerance levels chosen for τ_r

in Fig. 1, we also display the normalized singular values of matrix $\mathbf{\Gamma} \in \mathbb{R}^{1641\times 10^4}$. Note that machine precision is reached at the 112th singular value. We select three tolerance values $\tau_r \in \{10^{-4}, 10^{-5}, 10^{-6}\}$ for truncating matrices obtained from the SVD of $\mathbf{\Gamma}$.

In what follows, we compute reduced-order discrete-time models that have dimension r, as in (42). Next, these models are converted using (43) into a continuous-time model.

4.1.1 Experiment 1—Validating the Trained Models

In this first setup, we perform time-domain simulations of the reduced-order models for the same conditions as in the training stage, i.e., in the time horizon $[0, 10]s$ and by using the control input $u(t) = 0.5\cos(10t)e^{-0.3t}$. Hence, we are validating the trained models on the training data.

Start by choosing the first tolerance value, e.g., $\tau_r = 10^{-4}$. This corresponds to a truncation value of $r = 25$. We compute term $\hat{\mathbf{D}} = 1.1744\text{e} - 14$ and a also bilinear feed-through term with $\|\hat{\mathbf{F}}\|_2 = 6.7734\text{e} - 04$. We simulate both the original large-scale bilinear system and the reduced-order system. The results are presented in Fig. 2. Note that the observed output curves deviate substantially. One way to improve this behavior is to decrease the tolerance value.

For the next experiment, choose the tolerance value to be $\tau_r = 10^{-5}$. This corresponds to a truncation value of $r = 32$. After computing the required matrices, notice that the D term is again numerically 0, while the norm of the matrix $\hat{\mathbf{F}}$ slightly decreases to the value $6.9597e - 04$. Perform numerical simulations and depict the two outputs and the approximation error in Fig. 3. Observe that the approximation quality significantly improved, but there is still room for improvement.

Finally, the tolerance value is chosen as $\tau_r = 10^{-6}$. For this particular choice, it follows that the truncation value is $r = 40$. In this case, the output of the reduced-order model faithfully reproduces the original output, as it can be observed in Fig. 4. Note also that the approximation error stabilizes within the range $(10^{-4}, 10^{-5})$.

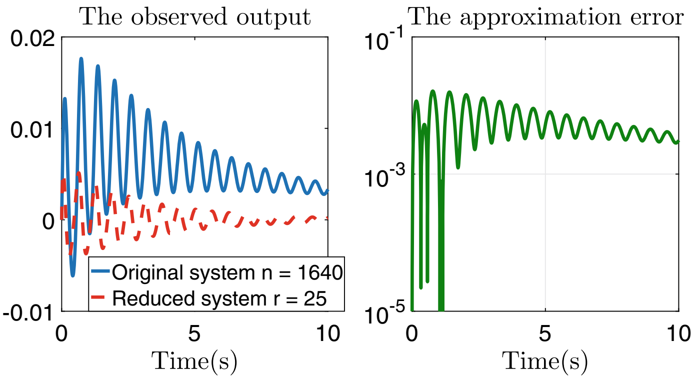

Fig. 2 Left plot: the observed outputs; right plot: the corresponding approximation error

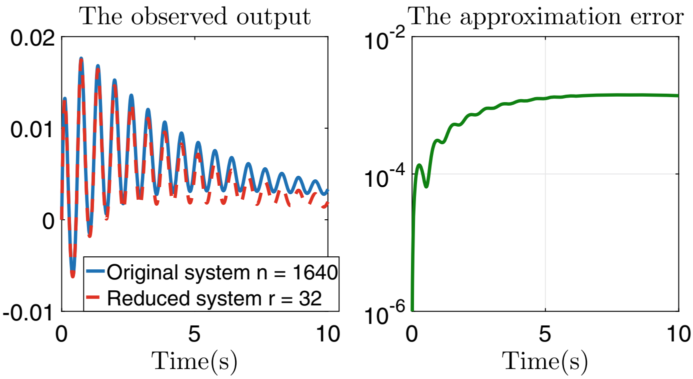

Fig. 3 Left plot: the observed outputs; right plot: the corresponding approximation error

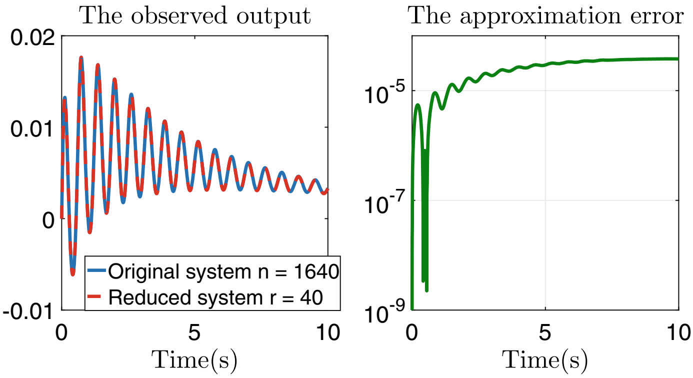

Fig. 4 Left plot: the observed outputs; right plot: the corresponding approximation error

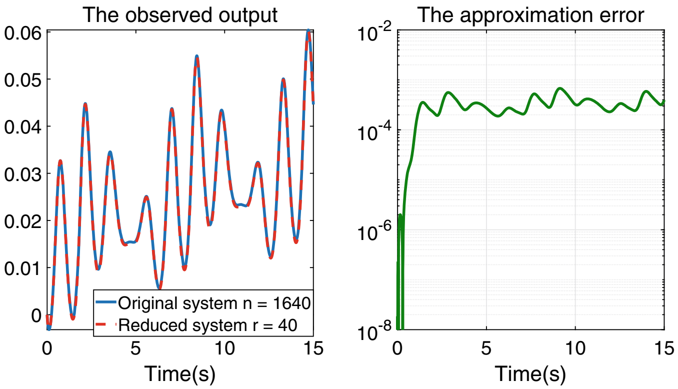

Fig. 5 Left plot: the observed outputs; right plot: the corresponding approximation error

4.1.2 Experiment 2—Testing the Trained Models

In the second setup, we perform time-domain simulations of the reduced-order models for different conditions than those used in the training stage, i.e., the time horizon is extended to $[0, 15]s$ and two other control inputs are used. Moreover, we keep the truncation value to be $r = 40$ (corresponding to tolerance $\tau_r = 10^{-6}$).

First, choose the testing control input to be $u_1(t) = \sin(4t)/4 - \cos(5t)/5$. The time-domain simulations showing the observed outputs are depicted in Fig. 5. Moreover, on the same figure, the magnitude of the approximation is presented. We observe that the output of the learned reduced model accurately approximates the output of the original system.

Afterward, choose the testing control input to be $u_2(t) = \frac{\text{square}(2t)}{5(t+1)}$. Note that $\text{square}(2t)$ is a square wave with period π. The time-domain simulations showing the observed outputs are depicted in Fig. 6. Moreover, on the same figure, the magnitude of the approximation is presented. We observe that the output of the learned reduced model does not approximate the output of the original system as well as in the previous experiments.

4.2 Coupled van der Pol Oscillators

Consider the coupled van der Pol oscillators along a limit cycle example given in [18]. The dynamics are characterized by the following six differential equations with linear and nonlinear (cubic) terms:

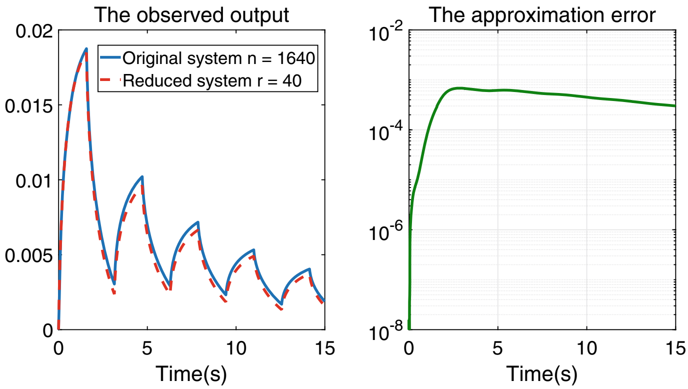

Fig. 6 Left plot: the observed outputs; right plot: the corresponding approximation error

$$\begin{aligned}
\dot{x}_1 &= x_2,\\
\dot{x}_2 &= -x_1 - \mu(x_1^2-1)x_2 + a(x_3-x_1) + b(x_4-x_2),\\
\dot{x}_3 &= x_4,\\
\dot{x}_4 &= -x_3 - \mu(x_3^2-1)x_4 + a(x_1-x_3) + b(x_2-x_4),\\
&\quad + a(x_5-x_3) + b(x_6-x_4) + u,\\
\dot{x}_5 &= x_6,\\
\dot{x}_6 &= -x_5 - \mu(x_5^2-1)x_6 + a(x_3-x_5) + b(x_4-x_6).
\end{aligned} \tag{50}$$

Choose the output to be $y = x_3$. Hence, the state-output equation is written as $y = \mathbf{C}\mathbf{x}$ with $\mathbf{C} = \begin{bmatrix} 0\ 0\ 1\ 0\ 0\ 0 \end{bmatrix}$ and $\mathbf{x} = \begin{bmatrix} x_1\ x_2\ x_3\ x_4\ x_5\ x_6 \end{bmatrix}^T$. Choose the parameters in (50) as follows: $\mu = 0.5,\ a = 0.5$ and $b = 0.2$.

Note that by introducing three additional surrogate states, e.g., $x_7 = x_1^2$, $x_8 = x_2^2$, and $x_9 = x_3^2$, one can rewrite the cubic nonlinear system in (50) of order $n = 6$ as an order $n_q = 9$ quadratic-bilinear system.

Perform time-domain simulations of the cubic system of order $n = 6$ and collect data from 500 snapshots using the explicit Euler method with step size $\Delta_t = 0.01$. The chosen time horizon is hence [0,5]s. The control input is a square wave with period $\pi/5$ and amplitude 30, i.e., $u(t) = 30$ square(10t).

Compute the pseudo-inverse of matrix $\mathbf{\Omega} \in \mathbb{R}^{49\times 500}$ and select as truncation value $p = 19$ (the 20th normalized singular value drops below machine precision).

We compute a reduced-order quadratic-bilinear model of order $r = 5$. We made this choice since the fifth normalized singular value of matrix $\mathbf{\Gamma} \in \mathbb{R}^{7\times 500}$ is 5.8651e-04 while the sixth is numerically 0, i.e., 3.5574e-16. We hence fit an order $r = 5$ quadratic-bilinear system that approximates the original order $n = 6$ cubic polynomial system. Note that the only nonzero feed-through quantity in the recovered state-output equation is given by $\hat{\mathbf{C}} = \begin{bmatrix} -0.1067\ 0.5580\ 0.0797\ -0.4145\ -0.7065 \end{bmatrix}$.

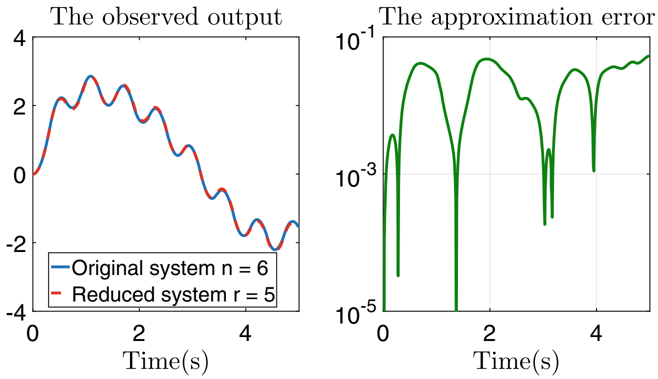

Fig. 7 Left plot: the observed outputs; Right plot: the corresponding approximation error

Next, we perform time-domain simulations in the same manner as in Sect. 4.1.1, i.e., by validating the reduced models on the training data. The results are depicted in Fig. 5. One can observe that the two outputs match well. In this particular setup, it follows that the response of the sixth-order cubic system (that can be equivalently written as a ninth-order QB system) can be accurately approximated with the response of a fifth-order QB system. The approximation error is presented in Fig. 7.

5 Conclusion

In this paper, we have proposed extensions of the DMDc and ioDMD recently proposed methods. The philosophy is similar to that of the original methods, but instead of fitting discrete-time linear systems, we impose a more complex structure to the fitted models. More precisely, we fit bilinear or quadratic terms to augment the existing linear quantities (both in the differential and in the output equations). The numerical results presented were promising, and they have shown the strength of the method. Indeed, there is a clear trade-off to be made between approximation quality and the dimension of the fitted model.

Nevertheless, this represents a first step toward extending DMD-type methods, and a more involved analysis of the method's advantages and disadvantages could represent an appealing future endeavor. Moreover, another contribution could be made by comparing the proposed methods in this work with the recently introduced operator inference-type methods. For the quadratic-bilinear case, additional challenges arise when storing the large-scale matrices involved and also when computing the classical SVD for such big non-sparse matrices.

6 Appendix

6.1 Computation of the Reduced-Order Matrices for the Quadratic-Bilinear Case

In this section, we present practical details for retrieving the system matrices in the case of the proposed procedure in Sect. 3.2. We solve the equation $\mathbf{\Gamma} = \mathbf{G}\mathbf{\Omega}$ for which the matrices are given as in (46), i.e., the case without output observations. We again utilize an SVD, now performed on the matrix $\mathbf{\Omega}$, i.e.,

$$\mathbf{\Omega} = \mathbf{V}\mathbf{\Sigma}\mathbf{W}^T \approx \tilde{\mathbf{V}}\tilde{\mathbf{\Sigma}}\tilde{\mathbf{W}}^T, \tag{51}$$

where the full-scale and reduced-order SVD matrices have the following dimensions:

$$\begin{cases} \mathbf{V} \in \mathbb{R}^{(n^2+2n+1)\times(n^2+2n+1)}, \ \ \mathbf{\Sigma} \in \mathbb{R}^{(n^2+2n+1)\times(m-1)}, \ \ \mathbf{W} \in \mathbb{R}^{(m-1)\times(m-1)}, \\ \tilde{\mathbf{V}} \in \mathbb{R}^{(n^2+2n+1)\times p}, \ \ \tilde{\mathbf{\Sigma}} \in \mathbb{R}^{p\times p}, \ \ \tilde{\mathbf{W}} \in \mathbb{R}^{(m-1)\times p}. \end{cases}$$

The truncation index is denoted with r, and written as before $\mathbf{\Omega}^{\dagger} \approx \tilde{\mathbf{W}}\tilde{\mathbf{\Sigma}}^{-1}\tilde{\mathbf{V}}^T$.

By splitting up the matrix $\mathbf{V}^T$ as $\tilde{\mathbf{V}}^T = [\tilde{\mathbf{V}}_1^T \ \ \tilde{\mathbf{V}}_2^T \ \ \tilde{\mathbf{V}}_3^T \ \ \tilde{\mathbf{V}}_4^T]$, with

$$\tilde{\mathbf{V}}_1, \tilde{\mathbf{V}}_3 \in \mathbb{R}^{n\times r}, \ \ \tilde{\mathbf{V}}_2 \in \mathbb{R}^{1\times r}, \ \ \tilde{\mathbf{V}}_4 \in \mathbb{R}^{n^2\times r},$$

recover the matrices

$$\overline{\mathbf{A}} = \mathbf{X}_s\tilde{\mathbf{W}}\tilde{\mathbf{\Sigma}}^{-1}\tilde{\mathbf{V}}_1^T, \ \ \overline{\mathbf{B}} = \mathbf{X}_s\tilde{\mathbf{W}}\tilde{\mathbf{\Sigma}}^{-1}\tilde{\mathbf{V}}_2^T, \ \ \overline{\mathbf{N}} = \mathbf{X}_s\tilde{\mathbf{W}}\tilde{\mathbf{\Sigma}}^{-1}\tilde{\mathbf{V}}_3^T, \ \ \overline{\mathbf{Q}} = \mathbf{X}_s\tilde{\mathbf{W}}\tilde{\mathbf{\Sigma}}^{-1}\tilde{\mathbf{V}}_4^T. \tag{52}$$

Again, perform an additional SVD, e.g., $\mathbf{X}_s \approx \hat{\mathbf{V}}\tilde{\mathbf{\Sigma}}\hat{\mathbf{W}}^T$, where $\hat{\mathbf{V}} \in \mathbb{R}^{(n+1)\times r}$, $\hat{\mathbf{\Sigma}} \in \mathbb{R}^{r\times r}$, $\hat{\mathbf{V}} \in \mathbb{R}^{(m-1)\times r}$. Using the transformation $\mathbf{x} = \hat{\mathbf{V}}\tilde{\mathbf{x}}$, the following reduced-order approximations are computed:

$$\begin{aligned}
\tilde{\mathbf{A}} &= \hat{\mathbf{V}}^T\overline{\mathbf{A}}\hat{\mathbf{V}} = \hat{\mathbf{V}}^T\mathbf{X}_s\tilde{\mathbf{W}}\tilde{\mathbf{\Sigma}}^{-1}\tilde{\mathbf{V}}_1^T\hat{\mathbf{V}} \in \mathbb{R}^{r\times r}, \\
\tilde{\mathbf{B}} &= \hat{\mathbf{V}}^T\overline{\mathbf{B}} = \hat{\mathbf{V}}^T\mathbf{X}_s\tilde{\mathbf{W}}\tilde{\mathbf{\Sigma}}^{-1}\tilde{\mathbf{V}}_2^T \in \mathbb{R}^{r}, \\
\tilde{\mathbf{N}} &= \hat{\mathbf{V}}^T\overline{\mathbf{A}}\hat{\mathbf{V}} = \hat{\mathbf{V}}^T\mathbf{X}_s\tilde{\mathbf{W}}\tilde{\mathbf{\Sigma}}^{-1}\tilde{\mathbf{V}}_3^T\hat{\mathbf{V}} \in \mathbb{R}^{r\times r}, \\
\tilde{\mathbf{Q}} &= \hat{\mathbf{V}}^T\overline{\mathbf{Q}}(\hat{\mathbf{V}} \otimes \hat{\mathbf{V}}) = \hat{\mathbf{V}}^T\mathbf{X}_s\tilde{\mathbf{W}}\tilde{\mathbf{\Sigma}}^{-1}\tilde{\mathbf{V}}_2^T(\hat{\mathbf{V}} \otimes \hat{\mathbf{V}}) \in \mathbb{R}^{r\times r^2}.
\end{aligned}$$

References

1. Annoni, J., Gebraad, P., Seiler, P.: Wind farm flow modeling using an input-output reduced-order model. In: 2016 American Control Conference (ACC), pp. 506–512. IEEE (2016)
2. Antoulas, A.C., Gosea, I.V., Heinkenschloss, M.: On the Loewner framework for model reduction of Burgers' equation. In: King, R. (ed.) Active Flow and Combustion Control, pp. 255–270. Springer (2018)
3. Antoulas, A.C., Gosea, I.V., Ionita, A.C.: Model reduction of bilinear systems in the Loewner framework. SIAM J. Sci. Comput. **38**(5), B889–B916 (2016)
4. Benner, P., Breiten, T.: Interpolation-based $\mathcal{H}_2$-model reduction of bilinear control systems. SIAM J. Matrix Anal. Appl. **33**(3), 859–885 (2012)
5. Benner, P., Breiten, T.: Two-sided projection methods for nonlinear model order reduction. SIAM J. Sci. Comput. **37**(2), B239–B260 (2015)
6. Benner, P., Breiten, T., Damm, T.: Generalised tangential interpolation for model reduction of discrete-time mimo bilinear systems. Int. J. Control **84**(8), 1398–1407 (2011)
7. 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)
8. Benner, P., Goyal, P., Kramer, B., Peherstorfer, B., Willcox, K.: Operator inference for non-intrusive model reduction of systems with non-polynomial nonlinear terms. arXiv preprint arXiv:2002.09726 (2020)
9. Benner, P., Himpe, C., Mitchell, T.: On reduced input-output dynamic mode decomposition. Adv. Comput. Math. **44**, 1751–1768 (2018)
10. Breiten, T., Damm, T.: Krylov subspace methods for model order reduction of bilinear control systems. Syst. Control Lett. **59**(8), 443–450 (2010)
11. Chen, K.K., Tu, J.H., Rowley, C.W.: Variants of dynamic mode decomposition: boundary condition, koopman, and fourier analyses. J. Nonlinear Sci. **22**(6), 887–915 (2012)
12. Dorissen, H.T.: A method for bilinear system identification, pp. 143–148 (1990)
13. Favoreel, W., de Moor, B., Van Overschee, P.: Subspace identification of bilinear systems subject to white inputs. IEE Trans. Autom. Control **44**(6), 1157–1165 (1999)
14. Gosea, I.V., Antoulas, A.C.: Data-driven model order reduction of quadratic-bilinear systems. Numer. Linear Algebra Appl. **25**(6), e2200 (2018)
15. Gu, C.: A projection-based nonlinear model order reduction approach using quadratic-linear representation of nonlinear systems. IEEE Trans. Comput.-Aided Design Integr. Circuits Syst. **30**, 1307–1320 (2011)
16. Isidori, A., Ruberti, A.: Realization theory of bilinear systems. In: Mayne, D.Q., Brockett, R.W. (eds.) Geometric Methods in System Theory. Springer, Dordrecht (1973)
17. Karachalios, D.S., Gosea, I.V., Antoulas, A.C.: On bilinear time domain identification and reduction in the Loewner framework. In: Model Reduction of Complex Dynamical Systems, International Series of Numerical Mathematics. Springer (2020). Accepted September 2020
18. Kawano, Y., Scherpen, J.: Empirical differential balancing for nonlinear systems, pp. 6326–6331 (2017)
19. Kutz, J.N., Brunton, S.L., Brunton, B.W., Proctor, J.L.: Dynamic Mode Decomposition: Data-driven Modeling of Complex Systems. SIAM (2016)
20. Le Clainche, S., Vega, J.M.: Higher order dynamic mode decomposition. SIAM J. Appl. Dyn. Syst. **16**(2), 882–925 (2017)
21. Mayo, A.J., Antoulas, A.C.: A framework for the solution of the generalized realization problem. Linear Algebra Appl. **425**(2–3), 634–662 (2007)
22. Mezić, I.: Analysis of fluid flows via spectral properties of the koopman operator. Annu. Rev. Fluid Mech. **45**, 357–378 (2013)
23. Peherstorfer, B.: Sampling low-dimensional markovian dynamics for pre-asymptotically recovering reduced models from data with operator inference. arXiv preprint arXiv:1908.11233 (2019)
24. Peherstorfer, B., Willcox, K.: Data-driven operator inference for nonintrusive projection-based model reduction. Comput. Methods Appl. Mech. Eng. **306**, 196–215 (2016)

25. Proctor, J.L., Brunton, S.L., Kutz, J.N.: Dynamic mode decomposition with control. SIAM J. Appl. Dyn. Syst. **15**(1), 142–161 (2016)
26. Qian, E., Kramer, B., Marques, A.N., Willcox, K.: Transform & learn: a data-driven approach to nonlinear model reduction. In: AIAA Aviation 2019 Forum, p. 3707 (2019)
27. Rowley, C.W., Mezić, I., Bagheri, S., Schlatter, P., Henningson, D.S.: Spectral analysis of nonlinear flows. J. Fluid Mech. **641**, 115–127 (2009)
28. Rugh, W.J.: Nonlinear System Theory - The Volterra/Wiener Approach. University Press (1981)
29. Schmid, P.J.: Dynamic mode decomposition of numerical and experimental data. J. Fluid Mech. **656**, 5–28 (2010)
30. Tu, J.H., Luchtenburg, D.M., Rowley, C.W., Brunton, S.L., Kutz, J.N.: On dynamic mode decomposition: theory and applications. J. Comput. Dyn. **1**, 391–421 (2014)
31. Tu, J.H., Rowley, C.W., Luchtenburg, D.M., Brunton, S.L., Kutz, J.N.: On dynamic mode decomposition: theory and applications. J. Comput. Dyn. **1**, 391–421 (2014)
32. Van Overschee, P., de Moor, B.: N4SID: subspace algorithms for the identification of combined deterministic-stochastic systems. Automatica **30**(1), 75–93 (1994)
33. Zhang, L., Lam, J., Huang, B., Yang, G.H.: On gramians and balanced truncation of discrete-time bilinear systems. Int. J. Control **76**(2003), 414–427 (1999)