applied sciences

MDPI

Article

Exact and Inexact Lifting Transformations of Nonlinear Dynamical Systems: Transfer Functions, Equivalence, and Complexity Reduction

Ion Victor Gosea

Data-Driven System Reduction and Identification Group, Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstrasse 1, 39114 Magdeburg, Germany; gosea@mpi-magdeburg.mpg.de; Tel.: +49-3916110361

Abstract: In this work, we deal with the problem of approximating and equivalently formulating generic nonlinear systems by means of specific classes thereof. Bilinear and quadratic-bilinear systems accomplish precisely this goal. Hence, by means of exact and inexact lifting transformations, we are able to reformulate the original nonlinear dynamics into a different, more simplified format. Additionally, we study the problem of complexity/model reduction of large-scale lifted models of nonlinear systems from data. The method under consideration is the Loewner framework, an established data-driven approach that requires samples of input–output mappings. The latter are known as generalized transfer functions, which are appropriately defined for both bilinear and quadratic-bilinear systems. We show connections between these mappings as well as between the matrices of reduced-order models. Finally, we illustrate the theoretical discussion with two numerical examples.

Keywords: nonlinear systems; complexity model reduction; data-driven methods; bilinear quadratic structure; transfer functions; lifting transformations

check for updates

Citation: Gosea, I.V. Exact and Inexact Lifting Transformations of Nonlinear Dynamical Systems: Transfer Functions, Equivalence, and Complexity Reduction. *Appl. Sci.* **2022**, *12*, 2333. https://doi.org/10.3390/app12052333

Academic Editors: Roman Starosta and Jan Awrejcewicz

Received: 23 January 2022
Accepted: 18 February 2022
Published: 23 February 2022

1. Introduction

Many problems from the applied sciences that involve a dynamic process under study, either to be solved, simulated, or even controlled, are described by models that could contain a considerable amount of variables and/or many inputs and outputs. This is particularly true in the context of fluid dynamics. For instance, equations, such as Navier–Stokes, Burgers', or Kuramoto–Sivashinsky, are nonlinear partial differential equations (PDEs), which are complex and challenging to deal with by means of conventional/direct methods.

Thus, these equations are often discretized and linearized, resulting in linear ordinary differential equations (ODEs) described by a huge amount of states (internal variables). The large dimension results in high computational complexity, which leads to challenges in terms of simulation, analysis, and design. Consequently, there is a need to obtain a simplified model of smaller dimensions, which can still be used to provide accurate information for the simulation, analysis, and control of such systems.

Model reduction is commonly viewed as a methodology used for reducing the computational complexity of large scale complex models in numerical simulations. The goal is to construct a smaller system with the same structure and similar response characteristics as the original. For an overview of conventional model reduction methods, we refer the reader to the books [1–3]. For a fairly general overview on system-theoretical model order reduction (MOR) approaches, we refer the reader to [4]. In this work, we assume that the nonlinear systems to be modeled contains quadratic nonlinearities or it can be equivalently transformed to such a structure.

This class of systems is of interest since most smooth nonlinear systems can be exactly reformulated (lifted) as quadratic or quadratic-bilinear time-invariant systems (QBTIs), provided that the nonlinearities are analytical. Thus, even if the original model does not involve explicitly (only) quadratic nonlinearities, we could assume this structure (as we will see in the next sections). The main lifting approaches involve artificially introducing new variables into the state-space representation, computing derivatives, and rewriting all relations in the required format.

MOR methods specifically tailored for reducing QBTI systems have been proposed in [5–8]. Such approaches are fairly new, and most of them represent extensions of the techniques proposed for model reduction of linear time-invariant (LTI in short) systems and bilinear time-invariant (BTI) systems [9]. The latter class of systems represent a class of mildly nonlinear dynamical systems, which inherits many properties from LTIs; more specifically speaking, BTIs have many systems and control theoretical quantities that are very closely related or can be easily extended from LTIs (infinite Gramians, system norms, stability properties, etc.). Additionally, model reduction of BTIs is a fairly established sub-field of MOR that has been constantly developed in recent decades. We mention here the research endeavors [10–14] that preceded the more recent works in [15–19].

With an ever-increasing availability of measured data in many engineering fields, the need for incorporating measurements in the modeling process has steadily grown over recent decades. The main challenge lies in effectively using the available data in order to construct/learn models that can accurately represent the dynamics of the underlying dynamic process. Sometimes, in order to satisfy accuracy requirements, the fitted/learned models might have large dimensions and, hence, not be suitable for fast numerical simulation.

Thus, it is of interest to learn reliable reduced-order surrogate models to be used instead. Here, we assume that such models have particular dynamic structure, i.e., the nonlinearities include bilinear or quadratic-bilinear terms. As seen in the sequel, this is by no means restrictive. In recent years, many data-driven and learning methods have been designed specifically for the case of BTIs [20,21] and also for QBTIs [22–26]. Here, we mention another prolific method for learning models from data, the so-called dynamic mode decomposition (DMD) [27,28], which was recently extended to deal with BTIs in [29,30].

In this work, we cover a number of recent extensions of the data-driven non-intrusive method known in the MOR community as the Loewner framework (LF), originally introduced in [31]. We show various applications and connections between two prominent extensions of LF. Another goal of this work is to illustrate some properties and similarities of two established reformulation (lifting) strategies:

1. Rewriting the dynamics of a generic nonlinear system (with analytic nonlinearities) into an equivalent form by enforcing quadratic-bilinear terms, and
2. Approximating systems with quadratic-bilinear terms with bilinear systems by neglecting higher-order terms (such as cubic) from the power series expansion.

Lifting and reformulation strategies include classical approaches, such as the ones in [32,33], as well as more modern methods. Some of the latter were already used in connection to MOR applications, such as in [25,34,35]. In this work, by following such processes, we are able to establish connections between the generalized transfer functions (input–output mappings in the frequency domain) of the two classes of lifted systems (bilinear and quadratic-bilinear).

Additionally, we provide methods of connecting the reduced order models in bilinear format with the original matrices corresponding to the quadratic-bilinear (lifted) model. To the best of the author's knowledge, connections, such as the ones mentioned above are indeed new and represent a fair share of this work's novelty factor. Finally, we will show applications of the approaches under consideration for two benchmarks examples (a semi-discretized classical problem of computational fluid dynamics and a nonlinear electrical circuit).

2. Transforming the Original Dynamics to Quadratic or to Bilinear Format

This section discusses several methods for finding simplified or equivalent models for given complex nonlinear problems. Specifically, one requires approximated models written in matrix format (as state-space representation). Then, one could asses how well the approximation performs and how the equivalent model captures the inherent properties of the original, complex, and possibly large-scale nonlinear system. In this section, we present two important classes of methods, which yield reformulated systems with specific structures (bilinear and quadratic-bilinear). We will start with the former in what follows, which is based on truncating power series.

As the Taylor series expansion is used to approximate infinitely differentiable complex-valued functions around a particular expansion point, so do some of the methods presented in this section (for nonlinear systems). Since we require finite-dimensional models, a truncation is in order. This would be the first limitation as truncation automatically introduces an error. The other is that, in some cases, the approximation is usually good enough only around the expansion point but not very accurate in other regions. Finally, the dimensions of the rewritten system are much higher than those of the original nonlinear system (denoted here with n), e.g., polynomial growth in n.

The alternative is equivalently rewriting the dynamics involving analytic nonlinearities by adding variables and/or by computing derivatives. In doing so, it turns out that the reformulation is exact (no approximation is involved) and additionally, the increase in the dimensions of the rewritten system is linear in n. The downside is that this applies only to nonlinear functions with certain properties (however, in practical applications, many relevant cases are indeed covered).

Consider a nonlinear system described in state-space representation by the following system of differential equations:

$$\mathbf{\Sigma}_{\mathrm{N}}: \begin{cases} \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) + \mathbf{g}(\mathbf{x}(t))\mathbf{u}(t), \\ \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t), \end{cases} \tag{1}$$

where $\mathbf{x}: \mathbb{R} \to \mathbb{R}^n$ is the state variable, $\mathbf{u}: \mathbb{R} \to \mathbb{R}$ is the control input, and $\mathbf{y}: \mathbb{R} \to \mathbb{R}$ is the observed output. Additionally, let $t \geqslant 0$, $\mathbf{x}(0) = \mathbf{x}_0$, and nonlinear vector-valued functions $\mathbf{f}, \mathbf{g}: \mathbb{R}^n \to \mathbb{R}^n$ that are analytic in $\mathbf{x}$. For simplicity, assume that the output $\mathbf{y}$ depends linearly on the variable $\mathbf{x}$ (the more general case would be $\mathbf{y}(t) = \mathbf{h}(\mathbf{x}(t))$ with $\mathbf{h}: \mathbb{R}^n \to \mathbb{R}^n$ also analytic in $\mathbf{x}$).

Without loss of generality, we assume that: $\mathbf{x}(0) = \mathbf{0}$, $\mathbf{f}(\mathbf{x}(0)) = \mathbf{f}(0) = \mathbf{0}$. If this does not hold, set $\hat{\mathbf{x}}(t) = \mathbf{x}(t) - \mathbf{x}_0(t)$ where $\mathbf{x}_0(t)$ is the zero input solution. Use the following notation $\hat{\mathbf{f}}(\hat{\mathbf{x}}(t)) = \mathbf{f}(\hat{\mathbf{x}}(t) + \mathbf{x}_0(t)) - \mathbf{f}(\mathbf{x}_0(t))$. Then, rewrite the system equations in terms of the new state:

$$\begin{aligned} \dot{\hat{\mathbf{x}}}(t) &= \dot{\mathbf{x}}(t) - \dot{\mathbf{x}}_0(t) = \mathbf{f}(\mathbf{x}(t)) + \mathbf{g}(\mathbf{x}(t))\mathbf{u}(t) - \mathbf{f}(\mathbf{x}_0(t)) \\ &= \mathbf{f}(\hat{\mathbf{x}}(t) + \mathbf{x}_0(t)) - \mathbf{f}(\mathbf{x}_0(t)) + \mathbf{g}(\mathbf{x}(t))\mathbf{u}(t) = \hat{\mathbf{f}}(\hat{\mathbf{x}}(t)) + \mathbf{g}(\mathbf{x}(t))\mathbf{u}(t), \end{aligned}$$

and also $\hat{\mathbf{y}}(t) = \mathbf{C}(\hat{\mathbf{x}}(t) + \mathbf{x}_0(t))$. Consequently, it follows that

$$\hat{\mathbf{x}}(0) = \mathbf{x}(0) - \mathbf{x}_0(0) = \mathbf{0}, \;\; \hat{\mathbf{f}}(\hat{\mathbf{x}}(0))) = \hat{\mathbf{f}}(0) = \mathbf{f}(\mathbf{x}_0) - \mathbf{f}(\mathbf{x}_0) = \mathbf{0}.$$

Definition 1 ([34])**.** *The function $\mathbf{h}: \mathbb{R} \to \mathbb{R}$ is considered to be elementary if and only if it is a polynomial, rational, exponential, logarithmic, trigonometric, or root function or a composition of any of these classes of functions.*

As mentioned in ([36]), there are a variety of practical applications where these kinds of elementary function appear

1. $\mathbf{h}_1(x) = x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_k^{\alpha_k}$, $\mathbf{x} = [x_1 \; x_2 \; \ldots \; x_k]^T$ in chemical rate equations, metal–oxide semiconductor field-effect transistor (MOSFET) in saturation mode.
2. $\mathbf{h}_2(x) = \frac{x}{k+x}$ in chemical engineering, describing chemical rate equations or smoothing functions.
3. $\mathbf{h}_3(x) = e^x$ in electronical circuit components, such as diodes or bipolar junction transistors.
4. $\mathbf{h}_4(x) = \sin(x)$ in control systems theory (where x is the angle used in steering of an object).

2.1. Bilinear Systems

We analyze bilinear time-invariant systems $\mathbf{\Sigma}_{\mathrm{B}} = (\mathbf{C}, \mathbf{E}, \mathbf{A}, \mathbf{N}, \mathbf{B})$ characterized by the following equations:

$$\mathbf{\Sigma}_{\mathrm{B}} : \begin{cases} \mathbf{E}\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{N}\mathbf{x}(t)\mathbf{u}(t) + \mathbf{B}\mathbf{u}(t), \\ \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t), \end{cases} \tag{2}$$

where $\mathbf{E},\ \mathbf{A},\ \mathbf{N} \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^{n\times m}$, $\mathbf{C} \in \mathbb{R}^{p\times n}$ and $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{u},\ \mathbf{y} \in \mathbb{R}$. Here, we assume that the matrix $\mathbf{E}$ is invertible and can be incorporated in the other matrices appearing in the differential Equation (2) as:

$$\breve{\mathbf{A}} = \mathbf{E}^{-1}\mathbf{A},\ \ \breve{\mathbf{N}} = \mathbf{E}^{-1}\mathbf{N},\ \ \breve{\mathbf{B}} = \mathbf{E}^{-1}\mathbf{B},\ \ \breve{\mathbf{E}} = \mathbf{I}_n,$$

where $\mathbf{I}_n \in \mathbb{R}^{n\times n}$ denotes the identity matrix of length n. However, in practical scenarios in which the model is of large dimensions, we avoid explicitly computing the inverse of matrix $\mathbf{E}$, and incorporate it as above. That is why we will interchange the two (equivalent) representations throughout the paper.

For simplicity of exposition, we will treat the single-input, single-output (SISO) case. The multi-input multi-output case is indeed technically more involved; however, it is based on the same type of reasoning.

2.1.1. Carleman's Linearization

In what follows, we introduce a well-established numerical tool that enforces approximating a nonlinear system with analytical nonlinearities be means of a bilinear system. The technique is commonly known as Carleman's linearization and was originally introduced in [32]. Since this is an approximation technique, it follows that the bilinear system obtained is clearly not equivalent to the initial one (in terms of the input–output response). One main property of this method is that, as the dimension of the bilinear system increases, the original nonlinear system is better approximated. Of course, in many cases, one cannot overly increase the dimensionality, since this results in challenges, such as low computational time or depleted memory storage.

The Kronecker product of the internal state variable $\mathbf{x} = [x_1\ \cdots\ x_n]^T$ with itself is defined as follows:

$$\mathbf{x} \otimes \mathbf{x} = [x_1^2\ \ x_1x_2\ \ \ldots\ \ x_1x_n\ \ \ldots\ \ x_n^2]^T \in \mathbb{R}^{n^2}. \tag{3}$$

Denote the composition of k Kronecker products of the same vector $\mathbf{x}$ with itself with

$$\mathbf{x}^{(k)}(t) = \mathbf{x}(t) \otimes \mathbf{x}(t) \otimes\ \ldots\ \otimes \mathbf{x}(t)\ \in\ \mathbb{R}^{n^k}, \tag{4}$$

where $\mathbf{x}^{(0)} = 1, \mathbf{x}^{(1)} = \mathbf{x},\ \mathbf{x}^{(2)} = \mathbf{x} \otimes \mathbf{x}$ and $\mathbf{x}^{(j)} \otimes \mathbf{x}^{(k)} = \mathbf{x}^{(j+k)}$, $\forall j, k \geqslant 1$. The infinite power series (*Maclaurin series*) expansions in $\mathbf{x}(t)$ for both functions $\mathbf{f}$ and $\mathbf{g}$ are written as follows:

$$\mathbf{f}(\mathbf{x}(t)) = \sum_{k=1}^{\infty} \mathbf{F}_k\mathbf{x}^{(k)}(t), \qquad \mathbf{g}(\mathbf{x})(t) = \sum_{k=0}^{\infty} \mathbf{G}_k\mathbf{x}^{(k)}(t), \tag{5}$$

where $\mathbf{G}_0 \in \mathbb{R}^{n\times 1}, \mathbf{F}_j, \mathbf{G}_j \in \mathbb{R}^{n^j\times n^j},\ j \geqslant 1$. Here, $\mathbf{F}_1, \mathbf{G}_1$ denote the Jacobian matrices for $\mathbf{f}$ and $\mathbf{g}$, respectively. Moreover, $\mathbf{F}_2, \mathbf{G}_2$ denote the matrices containing the second-order derivatives (Hessian matrices), and so on ($\mathbf{F}_k, \mathbf{G}_k$ denote the matrices containing the kth-order derivatives).

Now, by using a truncated version of the Maclaurin series of the functions by keeping only the first N terms for both series, where N represents the truncation index, we can write explicitly:

$$\begin{cases} \mathbf{f}(\mathbf{x}) = \sum_{k=1}^{N} \mathbf{F}_k \mathbf{x}^{(N)} = \mathbf{F}_1\mathbf{x} + \mathbf{F}_2\mathbf{x}^{(2)} + \ldots + \mathbf{F}_N\mathbf{x}^{(N)}, \\ \mathbf{g}(\mathbf{x}) = \sum_{k=0}^{N-1} \mathbf{G}_k \mathbf{x}^{(k)} = \mathbf{G}_0 + \mathbf{G}_1\mathbf{x} + \ldots + \mathbf{G}_{N-1}\mathbf{x}^{(N-1)}. \end{cases} \tag{6}$$

In the first equality in (6), we can neglect the term $\mathbf{F}_0$, without the loss of generality. This is usually chosen to be $\mathbf{0}$ in the case of non-existing forcing terms (and if indeed it is non-zero, it can be incorporated in the state vector by means of shifting). By substituting (6) into the state equation in (1), we write

$$\dot{\mathbf{x}}(t) = \sum_{k=1}^{N} \mathbf{F}_k \mathbf{x}^{(k)}(t) + \sum_{k=0}^{N-1} \mathbf{G}_k \mathbf{x}^{(k)}(t)\mathbf{u}(t). \tag{7}$$

Hence, it follows that the left hand side of the differential equation in (7) contains the state variable vector $\mathbf{x}(t)$, while the right hand side contains various powers of it $(\mathbf{x}^{(2)}(t), \ldots, \mathbf{x}^{(N)}(t))$. The strategy is to modify the state variable by adding higher-order terms that were missing; in this way, the dimension of the state vector increases. First, we compute the derivative of $\mathbf{x}^{(2)}(t)$ explicitly, as:

$$\begin{aligned} \dot{\mathbf{x}}^{(2)}(t) &= \frac{d\mathbf{x}^{(2)}(t)}{dt} = \frac{d\big(\mathbf{x}(t) \otimes \mathbf{x}(t)\big)}{dt} = \dot{\mathbf{x}}(t) \otimes \mathbf{x}(t) + \mathbf{x}(t) \otimes \dot{\mathbf{x}}(t) \\ &= \Big[\sum_{k=1}^{N} \mathbf{F}_k \mathbf{x}^{(k)}(t) + \sum_{k=0}^{N-1} \mathbf{G}_k \mathbf{x}^{(k)}(t)\mathbf{u}(t)\Big] \otimes \mathbf{x}(t) \\ &+ \mathbf{x}(t) \otimes \Big[\sum_{k=1}^{N} \mathbf{F}_k \mathbf{x}^{(k)}(t) + \sum_{k=0}^{N-1} \mathbf{G}_k \mathbf{x}^{(k)}(t)\mathbf{u}(t)\Big] \\ &= \sum_{k=1}^{N-1} [\mathbf{F}_k \otimes \mathbf{I}_n + \mathbf{I}_n \otimes \mathbf{F}_k]\mathbf{x}^{(k+1)}(t) + \sum_{k=0}^{N-2} [\mathbf{G}_k \otimes \mathbf{I}_n + \mathbf{I}_n \otimes \mathbf{G}_k]\mathbf{x}^{(k+1)}(t)\mathbf{u}(t). \end{aligned}$$

We introduce the following summation of j terms, each containing $j-1$ Kronecker products, as follows (for $j \geqslant 2$ and $k \geqslant 1$)

$$\mathbf{F}_{j,k} = \mathbf{F}_k \otimes \mathbf{I}_n \otimes \ldots \otimes \mathbf{I}_n + \mathbf{I}_n \otimes \mathbf{F}_k \otimes \mathbf{I}_n \otimes \ldots \otimes \mathbf{I}_n + \ldots + \mathbf{I}_n \otimes \mathbf{I}_n \otimes \ldots \otimes \mathbf{F}_k,$$

where we set $\mathbf{F}_{1,k} := \mathbf{F}_k$, by definition. For example, the following formulas hold true:

$$\mathbf{F}_{2,k} = \mathbf{F}_k \otimes \mathbf{I}_n + \mathbf{I}_n \otimes \mathbf{F}_k, \;\; \mathbf{F}_{3,k} = \mathbf{F}_k \otimes \mathbf{I}_n \otimes \mathbf{I}_n + \mathbf{I}_n \otimes \mathbf{F}_k \otimes \mathbf{I}_n + \mathbf{I}_n \otimes \mathbf{I}_n \otimes \mathbf{F}_k. \tag{8}$$

In a similar manner, define $\mathbf{G}_{j,k}$ for $j \geqslant 1$, $k \geqslant 0$. It follows that the time derivative of $\mathbf{x}^{(j)}(t)$ can indeed be written in terms of the newly defined matrices as:

$$\dot{\mathbf{x}}^{(j)}(t) = \sum_{k=1}^{N-j+1} \mathbf{F}_{j,k} \mathbf{x}^{(k+1)}(t) + \sum_{k=0}^{N-j} \mathbf{G}_{j,k} \mathbf{x}^{(k+1)}(t)\mathbf{u}(t), \;\; j \geqslant 1.$$

Increase the dimensions of the original state vector from n to $\mathcal{N} = n + n^2 + \ldots + n^N = \frac{n^N - n}{n-1}$, whereas n is the dimension of the original state variable $\mathbf{x}(t)$. In doing so, we also introduce a new state variable $\mathbf{x}^{\otimes}(t)$ defined as:

$$\mathbf{x}^{\otimes}(t) = \begin{bmatrix} \mathbf{x}(t) & \mathbf{x}^{(2)}(t) & \ldots & \mathbf{x}^{(N)}(t) \end{bmatrix}^T \in \mathbb{R}^{n^{(N)}}. \tag{9}$$

By substituting the newly introduced notation in (9), we obtain a bilinear system with the following realization

$$\begin{cases} \dot{\mathbf{x}}^{\otimes}(t) = \mathbf{A}^{\otimes}\mathbf{x}^{\otimes}(t) + \mathbf{N}^{\otimes}\mathbf{x}^{\otimes}(t)u(t) + \mathbf{B}^{\otimes}\mathbf{u}(t), \\ \quad \mathbf{y} = \mathbf{C}^{\otimes}\mathbf{x}^{\otimes}(t), \end{cases} \tag{10}$$

where $\mathbf{x}^{\otimes}(0) = \mathbf{0}$ and the matrices $\mathbf{A}^{\otimes}, \mathbf{N}^{\otimes} \in \mathbb{R}^{\mathcal{N}\times\mathcal{N}}, \mathbf{B}^{\otimes}, \left(\mathbf{C}^{\otimes}\right)^T \in \mathbb{R}^{\mathcal{N}}$ are as below:

$$\begin{aligned} \mathbf{A}^{\otimes} &= \begin{bmatrix} \mathbf{F}_{1,1} & \mathbf{F}_{1,2} & \dots & \mathbf{F}_{1,N} \\ \mathbf{0} & \mathbf{F}_{2,1} & \dots & \mathbf{F}_{2,N-1} \\ \mathbf{0} & \mathbf{0} & \ddots & \mathbf{F}_{3,N-2} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{F}_{N,1} \end{bmatrix}, \mathbf{N}^{\otimes} = \begin{bmatrix} \mathbf{G}_{1,1} & \mathbf{G}_{1,2} & \dots & \mathbf{G}_{1,N-1} & \mathbf{0} \\ \mathbf{G}_{2,0} & \mathbf{G}_{2,1} & \dots & \mathbf{G}_{2,N-2} & \mathbf{0} \\ \mathbf{0} & \mathbf{G}_{3,0} & \ddots & \mathbf{G}_{3,N-3} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{G}_{N,0} & \mathbf{0} \end{bmatrix}, \\ \mathbf{B}^{\otimes} &= \begin{bmatrix} \mathbf{G}_{10} & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix}, \mathbf{C}^{\otimes} = \begin{bmatrix} \mathbf{C} & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix}. \end{aligned}$$

This procedure involves approximation techniques (based on Taylor series expansion and truncation) to yield a BTI system that approximates the original system with generic nonlinearities. The main challenge with this method is that the dimensions of the derived bilinear system could be considerably higher than those of the original system. Indeed, by increasing the truncation index N will result in a better approximation. However, the increase in dimension of the bilinear system's dimension $\mathcal{N}$ might yield serious memory storage problems (the storage of very large matrices). For many practical applications, one would restrict to low values of N, such as $N = 2$ or $N = 3$.

Example 1. *Consider a non-linear scalar system characterized by the equations:*

$$\mathbf{\Sigma}_{\mathrm{N0}} : \begin{cases} \dot{x}(t) = -3x(t) + x(t)\cos\left(x(t)\right) + u(t), \\ y(t) = 2x(t). \end{cases}$$

Next, we use the Taylor series around the origin for the function $\cos(x(t))$

$$x\cos\left(x(t)\right) = \sum_{k=1}^{\infty} c_k x^k(t) = x(t) - \frac{x^3(t)}{2!} + \frac{x^5(t)}{4!} - \dots, \quad c_k = \begin{cases} 0, \ k \text{ even} \\ \frac{(-1)^{(k-1)/2}}{(k-1)!}, \ k \text{ odd} \end{cases},$$

and hence, write the differential equation as: $\dot{x}(t) = -2x(t) - \sum_{k=3}^{\infty} c_k x^k(t) + u(t)$. *We identify the following values:*

$$\begin{gathered} \mathbf{F}_{j,1} = -2j, \ \mathbf{F}_{j,k} = \begin{cases} 0, \ k \text{ even} \\ j\frac{(-1)^{(k-1)/2}}{(k-1)!}, \ k \text{ odd}, \ k > 1 \end{cases}, \\ \mathbf{G}_{j,0} = j, \ \mathbf{G}_{j,k} = 0, \ j,k \geqslant 1, \ \mathbf{C}_1 = 1, \ \mathbf{C}_h = 0, \ h \geqslant 2. \end{gathered}$$

First, we truncate at $N = 3$ *and by using that* $x^{(k)}(t) = x^k(t)$, *the new state variable is given by* $\mathbf{x}_1^{\otimes}(t) = \begin{bmatrix} x(t) & x^2(t) & x^3(t) \end{bmatrix}^T \in \mathbb{R}^3$; *the dynamics of the bilinear system corresponding to* $N = 3$ *are written as*

$$\dot{\mathbf{x}}_1^{\otimes}(t) = \mathbf{A}_1^{\otimes}\mathbf{x}_1^{\otimes}(t) + \mathbf{N}_1^{\otimes}\mathbf{x}_1^{\otimes}(t)u(t) + \mathbf{B}_1^{\otimes}u(t), \quad y(t) = \mathbf{C}_1^{\otimes}\mathbf{x}_1^{\otimes}(t),$$

with the system matrices computed as below:

$$\mathbf{A}_1^{\otimes} = \begin{bmatrix} -2 & 0 & -\frac{1}{2!} \\ 0 & -4 & 0 \\ 0 & 0 & -6 \end{bmatrix}, \mathbf{N}_1^{\otimes} = \begin{bmatrix} 0 & 0 & 0 \\ 2 & 0 & 0 \\ 0 & 3 & 0 \end{bmatrix}, \mathbf{B}_1^{\otimes} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \mathbf{C}_1^{\otimes} = \begin{bmatrix} 2 & 0 & 0 \end{bmatrix}.$$

Next, we increase the truncation index to $N = 5$, i.e., $\mathbf{x}_2^{\otimes}(t) = \begin{bmatrix} x(t) & x^2(t) & \dots & x^5(t) \end{bmatrix}^T$ with $\mathbf{x}_2^{\otimes}(t) \in \mathbb{R}^5$. Hence, write the state and output equation as

$$\dot{\mathbf{x}}_2^{\otimes}(t) = \mathbf{A}_2^{\otimes}\mathbf{x}_2^{\otimes}(t) + \mathbf{N}_2^{\otimes}\mathbf{x}_2^{\otimes}(t)u(t) + \mathbf{B}_2^{\otimes}u(t), \quad y(t) = \mathbf{C}_2^{\otimes}\mathbf{x}_2^{\otimes}(t),$$

with the following system matrices:

$$\mathbf{A}_2^{\otimes} = \begin{bmatrix} -2 & 0 & -\frac{1}{2!} & 0 & \frac{1}{4!} \\ 0 & -4 & 0 & -\frac{2}{2!} & 0 \\ 0 & 0 & -6 & 0 & -\frac{3}{2!} \\ 0 & 0 & 0 & -8 & 0 \\ 0 & 0 & 0 & 0 & -10 \end{bmatrix}, \ \mathbf{N}_2^{\otimes} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 \end{bmatrix}, \tag{11}$$

$$\mathbf{B}_2^{\otimes} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \ \mathbf{C}_2^{\otimes} = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{12}$$

2.2. *Quadratic-Bilinear Systems*

We analyze QBTI systems $\mathbf{\Sigma}_{\text{QB}} = (\mathbf{E}, \mathbf{A}, \mathbf{Q}, \mathbf{N}, \mathbf{B}, \mathbf{C})$ characterized by the following equations:

$$\mathbf{\Sigma}_{\text{QB}} : \begin{cases} \mathbf{E}\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{Q}\big(\mathbf{x}(t) \otimes \mathbf{x}(t)\big) + \mathbf{N}\mathbf{x}(t)\mathbf{u}(t) + \mathbf{B}\mathbf{u}(t), \\ \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t), \end{cases} \tag{13}$$

where $\mathbf{E}, \ \mathbf{A}, \ \mathbf{N} \in \mathbb{R}^{n \times n}$, $\mathbf{Q} \in \mathbb{R}^{n \times n^2}$, $\mathbf{B}, \mathbf{C}^T \in \mathbb{R}^n$ and $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{u}$, $\mathbf{y} \in \mathbb{R}$.

For simplicity of exposition, we treat the single-input single-output (SISO) case. The multi-input multi-output (MIMO) case is technically more involved but it is based on similar ideas and procedural steps.

A strong argument for studying this problem is that a broad class of nonlinear systems can be written in quadratic-bilinear form by using exact transformations. These transformations can be applied to many different fields, e.g., fluid mechanic systems, electrical circuit models, and biochemical rate equations.

2.2.1. Reformulating Nonlinear Systems as QBTI Systems

Next, we mention a transformation that has been known in the literature as the *McCormick relaxation* [33]. The main result of it is that, by computing derivatives or by adding algebraic equations, the initial nonlinear system can be transformed into a QBTI system. The advantage of this method is that no approximation is performed: the quadratic-bilinear system is equivalent to the initial nonlinear system. The downside is that the procedure works only for certain types of smooth analytic nonliniarities. For application purposes, it works for some cases; however, nevertheless, one has to investigate the applicability of this method to its full extent.

Next, we acknowledge that an algorithm that implements lifting procedures in an automated way was explicitly formulated in [36]. As stated before, this is needed to equivalently transform a general nonlinear system into a QBTI system as in (13). It is to be noted that lifting strategies were already applied in various MOR-related works, such as [25,35].

As originally introduced in [36], one can distinguish two main strategies, i.e., by adding quadratic algebraic equations or by computing derivatives. The first procedure is seldom applicable by its own, and in most cases, one has to combine both of them. To show how the first approach works, we present a simple but self-explanatory example.

Example 2. *Start with the nonlinear differential equation $\dot{x} = x + x^3 + u$. By introducing the new variable $z = x^2$, it follows that $\dot{x} = x + xz + u$, and hence the two are equivalent*

$$\dot{x} = x + x^3 + u \Leftrightarrow \begin{cases} \dot{x} = x + xz + u, \\ 0 = z - x^2. \end{cases}$$

For the second category, we present a probative example that illustrates the lifting approach appropriately.

Example 3. *Consider the same nonlinear system as before, i.e.,*

$$\mathbf{\Sigma}_{\mathrm{N1}} : \begin{cases} \dot{x}(t) = -3x(t) + x(t)\cos x(t) + u(t), \\ y(t) = 2x(t). \end{cases} \tag{14}$$

By introducing two additional variables: $z(t) = \sin x(t)$ and $w(t) = \cos x(t)$, compute the time derivatives and then substitute the given terms

$$\begin{aligned} \dot{x}(t) &= -3x(t) + x(t)w(t) + u(t), \\ \dot{z}(t) &= \dot{x}(t)\cos x(t) = [-3x(t) + x(t)w(t) + u(t)]w(t), \\ &= -3x(t)w(t) + x(t)w^2(t) + w(t)u(t), \\ \dot{w}(t) &= -x(\dot{t})\sin x(t) = -[-3x(t) + x(t)w(t) + u(t)]z(t) \\ &= 3x(t)z(t) - x(t)z(t)w(t) - z(t)u(t). \end{aligned} \tag{15}$$

We managed to equivalently rewrite the original nonlinear system as a polynomial system but with cubical nonlinearities as can be observed in (15). However, there is one more step until the transformation to QBTI format is completed. We need to choose another two surrogate variables, i.e., $p(t) = x(t)z(t)$ and $r(t) = x(t)w(t)$, to be able to equivalently rewrite the above CTI (cubic time-invariant) system as a QBTI system. Hence, by substituting the new variables into (15), we can re-write the differential equations as:

$$\begin{aligned} \dot{x}(t) &= -3x(t) + r(t) + u(t), \\ \dot{z}(t) &= -3r(t) + r(t)w(t) + w(t)u(t), \\ \dot{w}(t) &= 3p(t) - p(t)w(t) - z(t)u(t). \end{aligned} \tag{16}$$

Then, the next step is to explicitly compute the derivatives of the two new variables $p(t)$ and $r(t)$ as:

$$\begin{aligned} \dot{p}(t) &= \dot{x}(t)z(t) + x(t)\dot{z}(t) \\ &= [-3x(t) + r(t) + u(t)]z(t) + x(t)[-3r(t) + r(t)w(t) + w(t)u(t)] \\ &= -3x(t)z(t) + r(t)z(t) - 3x(t)r(t) + r^2(t) + [z(t) + r(t)]u(t), \end{aligned} \tag{17}$$

and also as

$$\begin{aligned} \dot{r}(t) &= \dot{x}(t)w(t) + x(t)\dot{w}(t) \\ &= [-3x(t) + r(t) + u(t)]w(t) + x(t)[3p(t) - p(t)w(t) - z(t)u(t)] \\ &= -3x(t)w(t) + r(t)w(t) + 3x(t)p(t) - p(t)r(t) + [w(t) - p(t)]u(t). \end{aligned} \tag{18}$$

By combining (16)–(18), it follows that we can construct a QBTI system $n = 5$ in the new variable $\tilde{\mathbf{x}}(t) = \begin{bmatrix} x(t) & z(t) & w(t) & p(t) & r(t) \end{bmatrix}^T$. This lifted system is indeed equivalent to the original scalar system in (14).

3. Deriving Input-Output Mappings for the Two Classes of Systems

3.1. The Variational Equation Approach

Consider a nonlinear system $\mathbf{\Sigma}_{\mathrm{N}}$ as defined in (1). Assume that the nonlinear functions $\mathbf{f}, \mathbf{g} : \mathbb{R}^n \to \mathbb{R}^n$ are analytic in $\mathbf{x}$. Without loss of generality we also enforce that: $\mathbf{x}(0) = \mathbf{0}$, $\mathbf{f}(\mathbf{x}(0)) = \mathbf{f}(0) = \mathbf{0}$.

Next, compute an approximate bilinear model $\mathbf{\Sigma}_{\mathrm{B}}$ via Carleman linearization presented in Section 2.1.1 and an exact quadratic bilinear model $\mathbf{\Sigma}_{\mathrm{QB}}$ via the reformulation procedure presented in Section 2.2.1. The idea is to decouple these two classes of systems and write them equivalently as an infinite linear time-varying representation.

This approach (see [37,38]) studies the response of the systems to control inputs of the form $\mathbf{u}(t) = \alpha\hat{\mathbf{u}}(t)$ (α is an arbitrary scalar, and $\hat{\mathbf{u}}(t)$ is a unitary signal). Since the dynamical systems under considerations contain analytic non-linearities, it follows that the solution $\mathbf{x}(t)$ also exhibits an analytic representation and thus can be written as an infinite power series expansion in the parameter α, e.g.,

$$\mathbf{x}(t) = \sum_{k=1}^{\infty} \alpha^k \mathbf{x}_k(t) = \alpha\mathbf{x}_1(t) + \alpha^2\mathbf{x}_2(t) + \alpha^2\mathbf{x}_3(t) + \ldots \tag{19}$$

We proceed with the bilinear model obtained via Carleman linearizaton, i.e., as introduced in (2)). By substituting (19) and the control input $\mathbf{u}(t) = \alpha\hat{\mathbf{u}}(t)$ onto the differential equation in (2), it follows that:

$$\begin{aligned}\mathbf{E}\sum_{k=1}^{\infty}\alpha^k\dot{\mathbf{x}}_k(t) &= \mathbf{A}\sum_{k=1}^{\infty}\alpha^k\mathbf{x}_k(t) + \mathbf{N}\Big(\sum_{k=1}^{\infty}\alpha^k\mathbf{x}_k(t)\Big)\alpha\hat{\mathbf{u}}(t) + \alpha\mathbf{B}\hat{\mathbf{u}}(t)\\ &= \Big(\sum_{k=1}^{\infty}\alpha^k\mathbf{A}\mathbf{x}_k(t)\Big) + \Big(\sum_{k=2}^{\infty}\alpha^k\mathbf{N}\mathbf{x}_{k-1}(t)\mathbf{u}(t)\Big) + \alpha\mathbf{B}\hat{\mathbf{u}}(t).\end{aligned}$$

Now, since this differential equation and the equation for the initial state must hold for any value of α, coefficients of same powers of α can be equated. By selecting the terms corresponding to α^k from both sides, we start with the equality $\mathbf{E}\dot{\mathbf{x}}_1(t) = \mathbf{A}\mathbf{x}_1(t) + \mathbf{B}\hat{\mathbf{u}}(t)$ (for $k = 1$). In general, by collecting all terms corresponding to α^k from both sides, for $k > 1$, we obtain equalities, such as $\mathbf{E}\dot{\mathbf{x}}_k(t) = \mathbf{A}\mathbf{x}_k(t) + \mathbf{N}\mathbf{x}_{k-1}(t)\hat{\mathbf{u}}(t)$. Each equality can be rewritten as $\mathbf{E}\dot{\mathbf{x}}_k(t) = \mathbf{A}x_k(t) + \mathbf{B}_k(t)\hat{\mathbf{u}}(t)$, by introducing $\mathbf{B}_k(t) := \mathbf{N}\mathbf{x}_{k-1}(t)\mathbf{u}(t)$, $k > 1$. Hence, a linear differential equation with time-varying coefficients (the vector $\mathbf{B}_k(t)$).

The next step is to decouple the differential equation corresponding to the BTI system into infinite many time-varying linear differential equations. We can, hence, rewrite the initial dynamics equivalently as follows:

$$\mathbf{E}\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{N}\mathbf{x}(t)\mathbf{u}(t)\mathbf{B}\hat{\mathbf{u}}(t) \;\;\Leftrightarrow\;\; \begin{cases}\mathbf{E}\dot{\mathbf{x}}_1(t) &= \mathbf{A}\mathbf{x}_1(t) + \mathbf{B}\hat{\mathbf{u}}(t),\\ \mathbf{E}\dot{\mathbf{x}}_2(t) &= \mathbf{A}x_2(t) + \mathbf{N}\mathbf{x}_1(t)\hat{\mathbf{u}}(t),\\ &\vdots\\ \mathbf{E}\dot{\mathbf{x}}_k(t) &= \mathbf{A}x_k(t) + \mathbf{N}\mathbf{x}_{k-1}(t)\hat{\mathbf{u}}(t),\\ &\vdots\end{cases}$$

Denote with $\mathbf{y}_k(t) = \mathbf{C}\mathbf{x}_k(t)$, $\forall\, k \geqslant 1$. The initial BTI system is written equivalently as infinitely many coupled linear subsystems. Denote with $\mathbf{\Sigma}_{\mathrm{B}}^{(k)}$ the kth subsystem whose dynamics are determined by $\mathbf{x}_k$ (for $k \geqslant 2$), i.e.,

$$\mathbf{\Sigma}_{\mathrm{B}}^{(k)} : \begin{cases}\mathbf{E}\dot{\mathbf{x}}_k(t) = \mathbf{A}\mathbf{x}_k(t) + \mathbf{N}\mathbf{x}_{k-1}(t)\hat{\mathbf{u}}(t),\\ \mathbf{y}_k = \mathbf{C}\mathbf{x}_k.\end{cases} \tag{20}$$

Note that $\mathbf{\Sigma}_{\mathrm{B}}^{(1)}$ is indeed a LTI system, while $\mathbf{\Sigma}_{\mathrm{B}}^{(k)}$ is a LTV (linear time-varying) system for all $k \geq 2$.

Next, we continue with splitting the dynamics of the QBTI model in (13), similar to the BTI models above. The following equality holds true:

$$\mathbf{Q}\big(\mathbf{x}(t) \otimes \mathbf{x}(t)\big) = \mathbf{Q}\Big(\sum_{k=1}^{\infty} \alpha^k \mathbf{x}_k(t)\Big) \otimes \Big(\sum_{k=1}^{\infty} \alpha^k \mathbf{x}_k(t)\Big) = \sum_{k=2}^{\infty} \alpha^k \Big[\sum_{j=1}^{k-1} \mathbf{Q}\big(\mathbf{x}_j(t) \otimes \mathbf{x}_{k-j}(t)\big)\Big].$$

By substituting (19) and $\mathbf{u}(t) = \alpha \hat{\mathbf{u}}(t)$ into the differential equation in (13), we write

$$\begin{aligned}
\mathbf{E}\sum_{k=1}^{\infty} \alpha^k \dot{\mathbf{x}}_k(t) &= \mathbf{A}\sum_{k=1}^{\infty} \alpha^k \mathbf{x}_k(t) + \mathbf{Q}\Big(\sum_{k=1}^{\infty} \alpha^k \mathbf{x}_k(t)\Big) \otimes \Big(\sum_{k=1}^{\infty} \alpha^k \mathbf{x}_k(t)\Big) \\
&\quad + \mathbf{N}\Big(\sum_{k=1}^{\infty} \alpha^k \mathbf{x}_k(t)\Big)\alpha \hat{\mathbf{u}}(t) + \alpha \mathbf{B}\hat{\mathbf{u}}(t) \\
&= \sum_{k=1}^{\infty} \alpha^k \mathbf{A} x_k(t) + \sum_{k=2}^{\infty} \alpha^k \Big[\sum_{j=1}^{k-1} \mathbf{Q}\big(x_j(t) \otimes \mathbf{x}_{k-j}(t)\big)\Big] \\
&\quad + \sum_{k=2}^{\infty} \alpha^k \mathbf{N}\mathbf{x}_{k-1}(t)\mathbf{u}(t) + \alpha \mathbf{B}\hat{\mathbf{u}}(t).
\end{aligned}$$

Now, similarly as for the bilinear case, since this differential equation is valid for all $\alpha \in \mathbb{R}$, coefficients corresponding to the same powers of α can be equated and doing so for the coefficients of α^k, $k > 1$, we can write

$$\mathbf{E}\dot{\mathbf{x}}_k(t) = \mathbf{A}\mathbf{x}_k(t) + \sum_{j=1}^{k-1} \mathbf{Q}\big(\mathbf{x}_j(t) \otimes \mathbf{x}_{k-j}(t)\big) + \mathbf{N}\mathbf{x}_{k-1}(t)\hat{\mathbf{u}}(t). \tag{21}$$

Decouple the differential equation above into infinitely many differential equations as

$$\begin{aligned}
&\mathbf{E}\dot{x}_1(t) = \mathbf{A}x_1(t) + \mathbf{B}\hat{\mathbf{u}}(t), \\
&\mathbf{E}\dot{x}_2(t) = \mathbf{A}x_2(t) + \mathbf{Q}\big(\mathbf{x}_1(t) \otimes \mathbf{x}_1(t)\big) + \mathbf{N}x_1(t)\hat{\mathbf{u}}(t), \\
&\qquad\vdots \\
&\mathbf{E}\dot{\mathbf{x}}_k(t) = \mathbf{A}\mathbf{x}_k(t) + \mathbf{Q}\big(\mathbf{x}_1(t) \otimes \mathbf{x}_{k-1}(t) + \ldots + \mathbf{x}_{k-1}(t) \otimes \mathbf{x}_1(t)\big) + \mathbf{N}\mathbf{x}_{k-1}(t)\hat{\mathbf{u}}(t), \\
&\qquad\vdots
\end{aligned}$$

so that (19) holds true. Hence, the initial QBTI system is written equivalently as infinitely many coupled LTV subsystems. Denote with $\mathbf{\Sigma}_{\mathrm{QB}}^{(k)}$ the kth subsystem in variable $\mathbf{x}_k$:

$$\mathbf{\Sigma}_{\mathrm{QB}}^{(k)} : \begin{cases} \mathbf{E}\dot{\mathbf{x}}_k(t) = \mathbf{A}\mathbf{x}_k(t) + \sum_{j=1}^{k-1} \mathbf{Q}\big(\mathbf{x}_j(t) \otimes \mathbf{x}_{k-j}(t)\big) + \mathbf{N}\mathbf{x}_{k-1}(t)\hat{\mathbf{u}}(t), \\ \mathbf{y}_k = \mathbf{C}\mathbf{x}_k. \end{cases} \tag{22}$$

The advantage of the *variational equation approach* is that the solution $\mathbf{x}(t)$ can be derived by iteratively solving a series of LTV systems, which are non-linearly coupled. For instance, starting with the first LTI subsystem, we can directly write the solution as $\mathbf{x}_1(t) = \int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{B}u(\tau)d\tau$ (assuming $\mathbf{x}_1(0) = \mathbf{0}$). By considering this expression as a pseudo-input for the second subsystem, an expression for $\mathbf{x}_2(t)$ can be directly derived, and so on. By continuing this iterative procedure, one obtains the so-called Volterra series representation [38] associated with the system $\mathbf{\Sigma}_{\mathrm{QB}}$.

3.2. Frequency Domain Mappings

The unilateral (or one-sided) Laplace transform of a univariate function $\mathbf{f} : \mathbb{R} \to \mathbb{R}$ is defined as

$$\mathbf{F}(s) = \mathcal{L}\{\mathbf{f}(t)\} = \int_0^\infty \mathbf{f}(t)e^{-st}dt, \tag{23}$$

while the extension of it directly follows for the multivariate case:

Definition 2. *Let* $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}$ *be a multivariate function in n variables. The Laplace transform of* $\mathbf{f}$ *is defined as*

$$\begin{aligned}\mathbf{F}(s_1, s_2, \ldots, s_n) &= \mathcal{L}\{\mathbf{f}(t_1, t_2, \ldots, t_n)\}\\ &= \int_0^\infty \ldots \int_0^\infty f(t_1, t_2, \ldots, t_n)e^{-s_1t_1}e^{-s_2t_2}\ldots e^{-s_nt_n}dt_1\ldots dt_n.\end{aligned}$$

Definition 3. *The transfer function* $\mathbf{H}(s)$ *of an LTI system* $\mathbf{\Sigma}_\mathrm{L}$ *(with* $\mathbf{E} = \mathbf{I}_n$*) is defined as the Laplace transfer of the impulse response* $h(t) = \mathbf{C}e^{\mathbf{A}t}\mathbf{B}$*. Hence, the transfer function can be written in terms of the system matrices as follows* $\mathbf{H}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}$*. By replacing the identity matrix with* $\mathbf{E}$*, we write the more general definition*

$$\mathbf{H}(s) = \mathbf{C}(s\mathbf{E} - \mathbf{A})^{-1}\mathbf{B}. \tag{24}$$

The exact derivation follows

$$\begin{aligned}\mathbf{H}(s) &= \mathcal{L}\{\mathbf{h}(t)\} = \int_0^\infty \mathbf{h}(t)e^{-st}dt = \int_0^\infty \mathbf{C}e^{\mathbf{A}t}\mathbf{B}e^{-st}dt = \int_0^\infty \mathbf{C}e^{-(s\mathbf{I}-\mathbf{A})t}\mathbf{B}dt\\ &= \mathbf{C}\Big(\int_0^\infty e^{-(s\mathbf{I}-\mathbf{A})t}dt\Big)\mathbf{B} = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}.\end{aligned}$$

Proposition 1 ([38])**.** *If the mapping* $\mathbf{f} : \mathbb{C}^n \to \mathbb{C}$ *can be written as a product of two multivariate factors* $\mathbf{x}$ *and* $\mathbf{y}$ *that do not depend on any common variables, i.e.,* $\mathbf{f}(t_1, \ldots, t_n) = \mathbf{x}(t_1, \ldots, t_j)\mathbf{y}(t_{j+1}, \ldots, t_n)$*, then by applying the multi-variate Laplace transform to this equality, it follows that* $\mathbf{F}(s_1, \ldots, s_n) = \mathbf{X}(s_1, \ldots, s_j)\mathbf{Y}(s_{j+1}, \ldots, s_n)$*.*

Definition 4. *The* ℓ*th (regular) transfer function associated with the system* $\mathbf{\Sigma}_\mathrm{B}$ *is defined by the Laplace transform of the* ℓ*th regular kernel. It can be expressed in terms of the system matrices as*

$$\mathbf{H}_\ell^{\mathbf{w}}(s_0, s_1, \ldots, s_\ell) = \mathbf{C}(s_0\mathbf{E} - \mathbf{A})^{-1}\mathbf{N}(s_1\mathbf{E} - \mathbf{A})^{-1}\mathbf{N}\cdots\mathbf{N}(s_\ell\mathbf{E} - \mathbf{A})^{-1}\mathbf{B}, \tag{25}$$

where $\ell \geqslant 1$ *and* $\mathbf{w} = (\underbrace{\mathbf{N}, \mathbf{N}, \ldots, \mathbf{N}}_{(\ell-1)\ \text{times}})$*, and for* $\ell = 0$*, we have* $\mathbf{H}_0(s_0) = \mathbf{C}(s_0\mathbf{E} - \mathbf{A})^{-1}\mathbf{B}$*.*

In what follows, we introduce a special class of rational functions that can be interpreted as generalized transfer functions of QBTIs and were originally introduced in [23,39]. The advantage of using these functions is that they are written as one product only, requiring chain multiplication of certain matrices, similar to the case of bilinear systems. Other types of transfer functions have been used for MOR of QBTI systems, such as the class of symmetric ones in [40]. Next, a series of definitions is set up following [23].

Definition 5. *For* $\Upsilon = \{\mathbf{N}, \mathbf{Q}\}$*,* $b \in \{1,2\}$ *and* $\mathcal{S} \in \mathbb{C}^b$ *consider the function* $\mathbf{\Gamma} : \Upsilon \times \mathbb{C}^b \to \mathbb{C}^{n^b \times n}$*, defined as follows:*

$$\mathbf{\Gamma}(w, \mathcal{S}) = \begin{cases}\mathbf{\Phi}(s_1),\ \textit{if } w = \mathbf{N},\ \mathcal{S} = (s_1) \textit{ and } b = 1,\\ \mathbf{\Phi}(s_1)\mathbf{B} \otimes \mathbf{\Phi}(s_2),\ \textit{if } w = \mathbf{Q},\ \mathcal{S} = (s_1, s_2) \textit{ and } b = 2.\end{cases} \tag{26}$$

More precisely, in (26), it follows that $\mathbf{\Gamma}(\mathbf{N}, s_1) = (s_1\mathbf{E} - \mathbf{A})^{-1}$ *and by* $\mathbf{\Gamma}(\mathbf{Q}, (s_1, s_2)) = (s_1\mathbf{E} - \mathbf{A})^{-1}\mathbf{B} \otimes (s_2\mathbf{E} - \mathbf{A})^{-1}$*.*

Denote with Y^{ℓ} the set of all ordered tuples of length ℓ with entries from Y, i.e., for all $\ell \geqslant 1$, write $\mathrm{Y}^{\ell} = \{(w_1, w_2, \ldots, w_\ell) | w_k \in \mathrm{Y}, 1 \leqslant k \leqslant \ell\}$. Moreover, $\mathrm{Y}^0 = \{\epsilon\}$ contains only the null symbol.

Definition 6. *Let $\ell, h \geqslant 0$ be two integers so that $\ell \leqslant h \leqslant 2\ell$ and consider the ordered tuple $\mathcal{S} = (s_0, s_1, \ldots, s_h)$, where $s_0, s_1, \ldots, s_h \in \mathbb{C}$. Additionally, let $\mathcal{S}_k$, $1 \leqslant k \leqslant \ell$ be a tuple of 1 or 2 complex numbers from the set $\{s_0, s_1, \ldots, s_h\}$ so that $\mathcal{S}_1 \circledcirc \mathcal{S}_2 \circledcirc \cdots \circledcirc \mathcal{S}_\ell = \mathcal{S}$. Finally, let $\mathbf{w} = (w_1, w_2, \ldots, w_\ell) \in \mathrm{Y}^{\ell}$ be a tuple of length $\ell \geqslant 0$. Associate the following level ℓ rational transfer function to the tuple $\mathbf{w} \in \mathrm{Y}^{\ell}$:*

$$\mathbf{H}_{\ell}^{\mathbf{w}}(s_0, s_1, \ldots, s_h) = \begin{cases} \mathbf{C}\mathbf{\Phi}(s_0)\mathbf{B}, \ \ell = 0, \ \mathbf{w} = \epsilon, \\ \mathbf{C}\mathbf{\Phi}(s_0) w_1 \mathbf{\Gamma}(w_1, \mathcal{S}_1) \cdots w_\ell \mathbf{\Gamma}(w_\ell, \mathcal{S}_\ell)\mathbf{B}, \ \ell \geqslant 1. \end{cases} \tag{27}$$

The rational functions introduced in (27) are hence divided into subcategories or levels. It follows that a number of 2^{ℓ} functions are associated with level $\ell \geqslant 0$. More precisely, the one corresponding to level 0 is the linear transfer function $\mathbf{H}_0^{\epsilon}(s) = \mathbf{C}\,\mathbf{\Phi}(s)\,\mathbf{B}$, which can also be denoted with $\mathbf{H}(s)$. Furthermore, the two functions corresponding to level 1 are written as:

$$\mathbf{H}_1^{\mathbf{N}}(s_0, s_1) = \mathbf{C}\,\mathbf{\Phi}(s_0)\,\mathbf{N}\,\mathbf{\Phi}(s_1)\mathbf{B}, \quad \mathbf{H}_1^{\mathbf{Q}}(s_0, s_1, s_2) = \mathbf{C}\,\mathbf{\Phi}(s_0)\,\mathbf{Q}\,\big(\mathbf{\Phi}(s_1)\mathbf{B} \otimes \mathbf{\Phi}(s_2)\mathbf{B}\big).$$

Note that the rational functions above can indeed be computed by applying the multivariate Laplace transform of time-domain kernels as explained in [38,39].

In general, a ℓth level transfer function defined in (27) is a multivariate rational function depending on h variables $\{s_1, \ldots, s_h\}$ for $\ell + 1 \leqslant h \leqslant 2\ell + 1$.

Note that, if $\mathbf{w} = (\mathbf{N}, \mathbf{N}, \ldots, \mathbf{N}) \in \mathrm{Y}^{\ell+1}, \ell > 0$ and $\mathcal{S}_\ell = (s_\ell)$, then the transfer function $\mathbf{H}_{\ell+1}^{\mathbf{w}}$ exactly coincides to the bilinear transfer functions proposed in [15], i.e.,

$$\mathbf{H}_{\ell}^{\mathbf{w}}(s_0, \ldots, s_\ell) = \mathbf{C}\mathbf{\Phi}(s_0)\mathbf{N}\mathbf{\Gamma}(\mathbf{N}, s_1) \cdots \mathbf{N}\mathbf{\Gamma}(\mathbf{N}, s_{\ell+1})\mathbf{B} = \mathbf{C}\mathbf{\Phi}(s_0)\mathbf{N}\mathbf{\Phi}(s_1) \cdots \mathbf{N}\mathbf{\Phi}(s_\ell)\mathbf{B}.$$

3.3. A Conenction between the Transfer Functions of the Two Classes of Systems

The original quadratic system is transformed into a bilinear system by means of Carleman linearization [32], as originally shown in [41] and later in [42], for particular models from electrical or mechanical engineering.

Now, let us revisit the previous derivation by using the original equations of the original quadratic-bilinear system written as

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{Q}(\mathbf{x}(t) \otimes \mathbf{x}(t)) + \mathbf{N}\mathbf{x}(t)\mathbf{u}(t) + \mathbf{B}\mathbf{u}(t) \\ &= \mathbf{A}\mathbf{x}(t) + \mathbf{Q}\mathbf{x}^{(2)}(t) + \mathbf{N}\mathbf{x}(t)\mathbf{u}(t) + \mathbf{B}\mathbf{u}(t). \end{aligned} \tag{28}$$

Then, we can write that

$$\begin{aligned} \dot{\mathbf{x}}^{(2)}(t) = \frac{d\mathbf{x}^{(2)}(t)}{dt} &= \frac{d\big(\mathbf{x}(t) \otimes \mathbf{x}(t)\big)}{dt} = \dot{\mathbf{x}}(t) \otimes \mathbf{x}(t) + \mathbf{x}(t) \otimes \dot{\mathbf{x}}(t) \\ &= \Big(\mathbf{A}\mathbf{x}(t) + \mathbf{Q}\mathbf{x}^{(2)}(t) + \mathbf{N}\mathbf{x}(t)\mathbf{u}(t) + \mathbf{B}\mathbf{u}(t)\Big) \otimes \mathbf{x}(t) \\ &+ \mathbf{x}(t) \otimes \Big(\mathbf{A}\mathbf{x}(t) + \mathbf{Q}\mathbf{x}^{(2)}(t) + \mathbf{N}\mathbf{x}(t)\mathbf{u}(t) + \mathbf{B}\mathbf{u}(t)\Big) \end{aligned} \tag{29}$$

By neglecting the higher-order cubic terms $\mathbf{x}^{(3)}(t)$ in the equality above, we can write

$$\dot{\mathbf{x}}^{(2)}(t) \approx (\mathbf{A}\mathbf{x}(t) + \mathbf{N}\mathbf{x}(t)\mathbf{u}(t) + \mathbf{B}\mathbf{u}(t)) \otimes \mathbf{x}(t) + \mathbf{x}(t) \otimes (\mathbf{A}\mathbf{x}(t) + \mathbf{N}\mathbf{x}(t)\mathbf{u}(t) + \mathbf{B}\mathbf{u}(t)). \tag{30}$$

It follows that the matrices of the BTI system computed from (13) via Carleman's linearization are given as below:

$$\begin{aligned}
\mathbf{A}^{\otimes} &= \begin{bmatrix} \mathbf{A} & \mathbf{Q} \\ \mathbf{0} & \mathbf{A}\otimes\mathbf{I}_n + \mathbf{I}_n\otimes\mathbf{A} \end{bmatrix}, \ \mathbf{B}^{\otimes} = \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix}, \\
\mathbf{N}^{\otimes} &= \begin{bmatrix} \mathbf{N} & \mathbf{0} \\ \mathbf{B}\otimes\mathbf{I}_n + \mathbf{I}_n\otimes\mathbf{B} & \mathbf{N}\otimes\mathbf{I}_n + \mathbf{I}_n\otimes\mathbf{N} \end{bmatrix}, \ \mathbf{C}^{\otimes} = \begin{bmatrix} \mathbf{C} & \mathbf{0} \end{bmatrix},
\end{aligned} \tag{31}$$

while the state vector is given by $\mathbf{x}^{\otimes}(t) = \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{x}(t)\otimes\mathbf{x}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{x}^{(1)}(t) \\ \mathbf{x}^{(2)}(t) \end{bmatrix}$.

A first observation is that the poles of the BTI model in (31), derived by applying Carleman's linearization, and the poles of the original QBTI model are clearly connected. By denoting with $\{\sigma_1, \ldots, \sigma_n\} := \sigma_{\mathbf{A}}$ the set of eigenvalues of matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, since $\mathbf{A}^{\otimes} = \begin{bmatrix} \mathbf{A} & \mathbf{Q} \\ \mathbf{0} & \mathbf{A}\otimes\mathbf{I}_n + \mathbf{I}_n\otimes\mathbf{A} \end{bmatrix}$, it follows that the set of eigenvalues of matrix $\mathbf{A}^{\otimes} \in \mathbb{R}^{\mathcal{N}\times\mathcal{N}}$ are written as $\sigma_{\mathbf{A}} \cup \{\sigma_{\mathbf{A}} \oplus \sigma_{\mathbf{A}}\}$; here, the set $\{\sigma_{\mathbf{A}} \oplus \sigma_{\mathbf{A}}\}$ includes all possible sums between two elements of $\sigma_{\mathbf{A}}$, by allowing repetitions (poles with multiplicities). For example, when $n = 2$, we have that $\{\sigma_{\mathbf{A}} \oplus \sigma_{\mathbf{A}}\} = \{2\sigma_1, \sigma_1 + \sigma_2, \sigma_2 + \sigma_1, 2\sigma_2\}$.

Next, we derive explicit formulas involving the transfer functions of the BTI system characterized by the matrices in (31), in terms of the matrices of the original QBTI system. We will be using the following matrix identity based on the Schur complement in order to explicitly compute the inverse of matrix $s\mathbf{I}_{\mathcal{N}} - \mathbf{A}^{\otimes}$, i.e.,

$$\begin{bmatrix} \mathbf{X} & \mathbf{Y} \\ \mathbf{Z} & \mathbf{T} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{X}^{-1} + \mathbf{X}^{-1}\mathbf{Y}\mathbf{S}^{-1}\mathbf{Z}\mathbf{X}^{-1} & -\mathbf{X}^{-1}\mathbf{Y}\mathbf{S}^{-1} \\ -\mathbf{S}^{-1}\mathbf{Z}\mathbf{X}^{-1} & \mathbf{S}^{-1} \end{bmatrix} \text{where } \mathbf{S} \overset{\text{def}}{=} \mathbf{T} - \mathbf{Z}\mathbf{X}^{-1}\mathbf{Y}. \tag{32}$$

Choose now $\mathbf{X} = s\mathbf{I}_n - \mathbf{A}$, $\mathbf{Y} = -\mathbf{Q}$, $\mathbf{Z} = \mathbf{0}$ and $\mathbf{T} = s\mathbf{I}_{n^2} - (\mathbf{A}\otimes\mathbf{I}_n + \mathbf{I}_n\otimes\mathbf{A})$. Hence, $\mathbf{S} = \mathbf{T} = s\mathbf{I}_{n^2} - (\mathbf{A}\otimes\mathbf{I}_n + \mathbf{I}_n\otimes\mathbf{A})$. It follows that:

$$\left(s\mathbf{I}_{\mathcal{N}} - \mathbf{A}^{\otimes}\right)^{-1} = \begin{bmatrix} (s\mathbf{I}_n - \mathbf{A})^{-1} & (s\mathbf{I}_n - \mathbf{A})^{-1}\mathbf{Q}[s\mathbf{I}_{n^2} - (\mathbf{A}\otimes\mathbf{I}_n + \mathbf{I}_n\otimes\mathbf{A})]^{-1} \\ \mathbf{0} & [s\mathbf{I}_{n^2} - (\mathbf{A}\otimes\mathbf{I}_n + \mathbf{I}_n\otimes\mathbf{A})]^{-1} \end{bmatrix} \tag{33}$$

We can now compute the first transfer function of the bilinear model as in (4):

$$\mathbf{H}_0(s_0) = \mathbf{C}^{\otimes}\left(s_0\mathbf{I}_{\mathcal{N}} - \mathbf{A}^{\otimes}\right)^{-1}\mathbf{B}^{\otimes} = \mathbf{C}(s_0\mathbf{I}_n - \mathbf{A})^{-1}\mathbf{B}, \tag{34}$$

which is the transfer function of classical LTI systems (here, corresponding to the linear sub-model of the original QBTI system). Additionally, we can write that

$$\begin{aligned}
\mathbf{H}_1^{\mathbf{N}^{\otimes}}(s_0, s_1) &= \mathbf{C}^{\otimes}\left(s_0\mathbf{I}_{\mathcal{N}} - \mathbf{A}^{\otimes}\right)^{-1}\mathbf{N}^{\otimes}\left(s_1\mathbf{I}_{\mathcal{N}} - \mathbf{A}^{\otimes}\right)^{-1}\mathbf{B}^{\otimes} \\
&= \underbrace{\mathbf{C}(s_0\mathbf{I}_n - \mathbf{A})^{-1}\mathbf{N}(s_1\mathbf{I}_n - \mathbf{A})^{-1}\mathbf{B}}_{\mathbf{H}_1^{\mathbf{N}}(s_0,s_1)} + \mathbf{K}_1(s_0, s_1),
\end{aligned} \tag{35}$$

where $\mathbf{K}_1(s_0, s_1)$ is a bivariate function defined as:

$$\begin{aligned}
\mathbf{K}_1(s_0, s_1) = \mathbf{C}(s_0\mathbf{I}_n - \mathbf{A})^{-1}\mathbf{Q}[s_0\mathbf{I}_{n^2} - (\mathbf{A}\otimes\mathbf{I}_n + \mathbf{I}_n\otimes\mathbf{A})]^{-1} \\
(\mathbf{B}\otimes\mathbf{I}_n + \mathbf{I}_n\otimes\mathbf{B})(s_1\mathbf{I}_n - \mathbf{A})^{-1}\mathbf{B}.
\end{aligned} \tag{36}$$

Now, by using basic properties of the Kronecker product, such as $(\mathbf{v} \otimes \mathbf{I}_n)\mathbf{w} = \mathbf{v} \otimes \mathbf{w}$ and $(\mathbf{I}_n \otimes \mathbf{v})\mathbf{w} = \mathbf{w} \otimes \mathbf{v}$ for any vectors $\mathbf{v}, \mathbf{w} \in \mathbb{C}^n$, one can write that:

$$\begin{aligned} \mathbf{K}_1(s_0, s_1) &= \mathbf{C}(s_0\mathbf{I}_n - \mathbf{A})^{-1}\mathbf{Q}[s_0\mathbf{I}_{n^2} - (\mathbf{A} \otimes \mathbf{I}_n + \mathbf{I}_n \otimes \mathbf{A})]^{-1} \\ &\quad \Big[\mathbf{B} \otimes (s_1\mathbf{I}_n - \mathbf{A})^{-1}\mathbf{B} + (s_1\mathbf{I}_n - \mathbf{A})^{-1}\mathbf{B} \otimes \mathbf{B}\Big] \\ &= \mathbf{C}(s_0\mathbf{I}_n - \mathbf{A})^{-1}\mathbf{Q}[s_0\mathbf{I}_{n^2} - (\mathbf{A} \otimes \mathbf{I}_n + \mathbf{I}_n \otimes \mathbf{A})]^{-1} \\ &\quad \Big[\mathbf{I}_n \otimes (s_1\mathbf{I}_n - \mathbf{A})^{-1} + (s_1\mathbf{I}_n - \mathbf{A})^{-1} \otimes \mathbf{I}_n\Big](\mathbf{B} \otimes \mathbf{B}). \end{aligned} \tag{37}$$

Next, by using the identity $(\mathbf{X} \otimes \mathbf{Y})^{-1} = \mathbf{X}^{-1} \otimes \mathbf{Y}^{-1}$ that holds for all invertible matrices $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{n \times n}$, it follows that

$$\begin{aligned} \mathbf{I}_n \otimes (s_1\mathbf{I}_n - \mathbf{A})^{-1} &= (s_1\mathbf{I}_{n^2} - \mathbf{I}_n \otimes \mathbf{A})^{-1}, \\ (s_1\mathbf{I}_n - \mathbf{A})^{-1} \otimes \mathbf{I}_n &= (s_1\mathbf{I}_{n^2} - \mathbf{A} \otimes \mathbf{I}_n)^{-1}, \\ (s_1\mathbf{I}_n - \mathbf{A})^{-1} \otimes (s_1\mathbf{I}_n - \mathbf{A})^{-1} &= (s_1^2\mathbf{I}_{n^2} - (\mathbf{I}_n \otimes \mathbf{A} + \mathbf{A} \otimes \mathbf{I}_n) + \mathbf{A}^2)^{-1}. \end{aligned} \tag{38}$$

Hence, it follows that the function $\mathbf{K}_1(s_0, s_1)$ above can be further simplified and written in a similar format as to that of the quadratic transfer function $\mathbf{H}_1^{\mathbf{Q}}(s_0, s_1, s_1)$ given by:

$$\mathbf{H}_1^{\mathbf{Q}}(s_0, s_1, s_1) = \mathbf{C}(s_0\mathbf{I}_n - \mathbf{A})^{-1}\mathbf{Q}\Big[(s_1\mathbf{I}_n - \mathbf{A})^{-1}\mathbf{B} \otimes (s_1\mathbf{I}_n - \mathbf{A})^{-1}\mathbf{B}\Big] \tag{39}$$

$$= \mathbf{C}(s_0\mathbf{I}_n - \mathbf{A})^{-1}\mathbf{Q}\Big[(s_1\mathbf{I}_n - \mathbf{A})^{-1} \otimes (s_1\mathbf{I}_n - \mathbf{A})^{-1}\Big](\mathbf{B} \otimes \mathbf{B}). \tag{40}$$

4. Extensions of the Loewner Framework

The Loewner framework [31] constructs ROMs that approximate the transfer function of the underlying model, by using data, i.e., samples of the transfer function at particular evaluation points. We refer the reader to the recent handbooks [43–45] for a more comprehensive description.

Extensions of the Loewner framework have been proposed in recent years (some of them collected in [39]); these were mainly based on matching values of the generalized transfer functions introduced in Section 3, at certain evaluation points. Here, we present a simplified presentation of the main approaches and refer to the original works in [20,23] for a more detailed analysis, the general interpolation scheme, and many other derivations and complementary results.

As in the classical linear case [31], the interpolation points are partitioned into two disjoint sets of left and right interpolation points. Since the transfer functions corresponding to the original bilinear or to the quadratic-bilinear system depend on multiple frequencies, the interpolation points need to be arranged in a suitable way. To simplify the presentation, we show the procedure for BTIs only, and we refer the reader to [23] for the algorithm in the case of QBTIs.

Additionally, assume that $k = 2\tilde{k}$ left and right interpolation points are available, which are grouped as follows:

$$\begin{aligned} \text{left points}: \ & \mu_1^{(1)}, \mu_2^{(1)}, \ldots, \mu_1^{(\tilde{k})}, \mu_2^{(\tilde{k})}, \\ \text{right points}: \ & \lambda_1^{(1)}, \lambda_2^{(1)}, \ldots, \lambda_1^{(\tilde{k})}, \lambda_2^{(\tilde{k})}. \end{aligned} \tag{41}$$

Next, the left and right interpolation points are grouped in multi-tuples for $j = 1, \ldots, \tilde{k}$:

$$\boldsymbol{\mu}^{(j)} = \{(\mu_1^{(j)}),\ (\mu_1^{(j)},\ \mu_2^{(j)})\}, \quad \boldsymbol{\lambda}^{(j)} = \{(\lambda_1^{(j)}),\ (\lambda_2^{(j)},\ \lambda_1^{(j)})\}. \tag{42}$$

The generalized controllability matrix $\mathcal{R} \in \mathbb{C}^{n\times k}$ associated with the right multi-tuples $\boldsymbol{\lambda}^{(1)}, \boldsymbol{\lambda}^{(2)}, \ldots, \boldsymbol{\lambda}^{(\tilde{k})}$ is

$$\mathcal{R} = \left[\mathcal{R}^{(1)},\ \mathcal{R}^{(2)},\ \cdots,\ \mathcal{R}^{(\tilde{k})}\right], \tag{43}$$

where the matrices $\mathcal{R}^{(j)} \in \mathbb{C}^{n\times 2}$, $j = 1, \ldots, \tilde{k}$, are associated with the j-th multi-tuple $\boldsymbol{\lambda}^{(j)}$ in (42) are given by

$$\mathcal{R}^{(j)} = \left[(\lambda_1^{(j)}\mathbf{E} - \mathbf{A})^{-1}\,\mathbf{B},\ \ (\lambda_2^{(j)}\mathbf{E} - \mathbf{A})^{-1}\,\mathbf{N}\,(\lambda_1^{(j)}\mathbf{E} - \mathbf{A})^{-1}\,\mathbf{B},\right]. \tag{44}$$

Similarly, the generalized observability matrix $\mathcal{O} \in \mathbb{C}^{k\times n}$ associated with the left multi-tuples $\boldsymbol{\mu}^{(1)}, \boldsymbol{\mu}^{(2)}, \ldots, \boldsymbol{\mu}^{(\bar{k})}$ is given by

$$\mathcal{O} = \left[(\mathcal{O}^{(1)})^T,\ (\mathcal{O}^{(2)})^T,\ \ldots (\mathcal{O}^{(\bar{k})})^T\right]^T \in \mathbb{C}^{k\times n}, \tag{45}$$

where $\mathcal{O}^{(i)} \in \mathbb{C}^{2\times n}$, $i = 1, \ldots, \bar{k}$, correspond to the i-th multi-tuple $\boldsymbol{\mu}^{(i)}$ in (42) and

$$\mathcal{O}^{(i)} = \begin{bmatrix} \mathbf{C}^T\,(\mu_1^{(i)}\mathbf{E} - \mathbf{A})^{-1} \\ \mathbf{C}^T\,(\mu_1^{(i)}\mathbf{E} - \mathbf{A})^{-1}\,\mathbf{N}\,(\mu_2^{(i)}\mathbf{E} - \mathbf{A})^{-1} \end{bmatrix}. \tag{46}$$

Next, the Loewner matrix $\mathbb{L}$ and the shifted Loewner matrix $\mathbb{L}_s$ are defined using the generalized controllability (43) and observability (45) matrices as

$$\mathbb{L} = -\mathcal{O}\,\mathbf{E}\,\mathcal{R},\ \ \mathbb{L}_s = -\mathcal{O}\,\mathbf{A}\,\mathcal{R}. \tag{47}$$

The fact that the Loewner matrices are factorized in terms of the pairs of matrices $(\mathbf{E}, \mathbf{A})$ and $(\mathcal{O}, \mathcal{R})$ is an inherent property of the Loewner framework, which holds true for both the bilinear and quadratic-bilinear extensions of the method.

Now, using the structure in (44) and in (46), it follows that:

$$\mathbb{L}^{(i,j)} = -\mathcal{O}^{(i)}\,\mathbf{E}\,\mathcal{R}^{(j)} \tag{48}$$

$$= \begin{bmatrix} \frac{\mathbf{H}_0(\mu_1^{(i)}) - \mathbf{H}_0(\lambda_1^{(j)})}{\mu_1^{(i)} - \lambda_1^{(j)}} & \frac{\mathbf{H}_1^{\mathbf{N}}(\mu_1^{(i)},\lambda_1^{(j)}) - \mathbf{H}_1^{\mathbf{N}}(\lambda_2^{(j)},\lambda_1^{(j)})}{\mu_1^{(i)} - \lambda_2^{(j)}} \\ \frac{\mathbf{H}_1^{\mathbf{N}}(\mu_1^{(i)},\mu_2^{(i)}) - \mathbf{H}_1^{\mathbf{N}}(\mu_1^{(i)},\lambda_1^{(j)})}{\mu_2^{(i)} - \lambda_1^{(j)}} & \frac{\mathbf{H}_2^{\mathbf{N},\mathbf{N}}(\mu_1^{(i)},\mu_2^{(i)},\lambda_1^{(j)}) - \mathbf{H}_2^{\mathbf{N},\mathbf{N}}(\mu_1^{(i)},\lambda_2^{(j)},\lambda_1^{(j)})}{\mu_2^{(i)} - \lambda_2^{(j)}} \end{bmatrix}, \tag{49}$$

and similarly for ${\mathbb{L}_s}^{(i,j)}$. Hence, matrices $\mathbb{L}$ and $\mathbb{L}_s$ are indeed data matrices, since all their entries are samples of the system's transfer functions denoted with $\mathbf{H}_0(s_0), \mathbf{H}_1^{\mathbf{N}}(s_0, s_1)$, and $\mathbf{H}_2^{\mathbf{N},\mathbf{N}}(s_0, s_1, s_2)$ (evaluated at particular grid of points). Next, introduce matrices:

$$\mathbb{V} = \mathcal{O}\mathbf{B},\ \ \mathbb{W} = \mathbf{C}\mathcal{R},\ \ \mathbb{T} = \mathcal{O}\mathbf{N}\mathcal{R}, \tag{50}$$

which can also be shown to be composed solely in terms of data, i.e., evaluations of the same transfer functions, introduced in (25). For matrix $\mathbb{T}$, one can show that the (i, j) block is explicitly written as follows:

$$\mathbb{T}^{(i,j)} = \mathcal{O}^{(i)}\,\mathbf{N}\,\mathcal{R}^{(j)} = \begin{bmatrix} \mathbf{H}_1^{\mathbf{N}}(\mu_1^{(i)}, \lambda_1^{(j)}) & \mathbf{H}_2^{\mathbf{N}}(\mu_1^{(i)}, \lambda_2^{(j)}, \lambda_1^{(j)}) \\ \mathbf{H}_2^{\mathbf{N}}(\mu_1^{(i)}, \mu_2^{(i)}, \lambda_1^{(j)}) & \mathbf{H}_3^{\mathbf{N}}(\mu_1^{(i)}, \mu_2^{(i)}, \lambda_2^{(j)}, \lambda_1^{(j)}) \end{bmatrix}. \tag{51}$$

As in the linear case, we focus on the case characterized by a redundant amount of data, which typically applies in practice. In such scenarios, the power of the Loewner framework relies on compressing and extracting the relevant information from the (possibly large) amount of data. Consequently, the singular value decomposition (SVD) is used as in the sequel.

However, other factorization schemes, such as the QR factorization, or the CUR factorization [44], can indeed be used. Next, consider the (short) SVDs of the Loewner matrices (47). The matrices $\mathbf{Y}, \mathbf{X} \in \mathbb{C}^{k \times r}$ are obtained by selecting the first r columns of the matrices $\mathbf{Y}_1$ and $\mathbf{X}_2$. We define the following projection matrices (see also [46]):

$$\mathbf{V} = \mathcal{R}\mathbf{X} \in \mathbb{C}^{n \times r}, \quad \mathbf{W} = \mathcal{O}^*\mathbf{Y} \in \mathbb{C}^{n \times r}. \tag{52}$$

The Loewner ROM is given by the following matrices:

$$\begin{aligned} \widehat{\mathbf{E}} &= -\mathbf{Y}^*\mathbb{L}\mathbf{X} = \mathbf{W}^*\mathbf{E}\mathbf{V}, \widehat{\mathbf{A}} = -\mathbf{Y}^*\mathbb{L}_s\mathbf{X}^* = \mathbf{W}^*\mathbf{E}\mathbf{V}, \\ \widehat{\mathbf{N}} &= \mathbf{W}^*\mathbf{N}\mathbf{V}, \ \ \widehat{\mathbf{B}} = \mathbf{W}^*\,\mathbf{B}, \widehat{\mathbf{C}} = \mathbf{V}^*\,\mathbf{C}. \end{aligned} \tag{53}$$

As in the linear case treated in [31], the projection matrices $\mathbf{V}, \mathbf{W}$, and the ROM have complex entries; however, one can enforce real-valued matrices of the ROM if the sets of left and of right interpolation points also contain the conjugate complex data as shown in [43] (one also needs to apply some specific transformations).

As shown in (53), the Loewner ROM (53) can be computed using a Petrov–Galerkin projection (see also [1] for more details on this classical and widespread approach). However, this requires explicit access to the matrices of the original system. As in the linear case, it is possible to generate the same Loewner ROM directly from measurements of the generalized transfer functions. We refer the reader to [23,46] for details on the extensions and recent applications of the Loewner framework to QBTI systems.

Petrov–Galerkin Projections Applied to the Bilinear Model Computed via Carleman's Linearization

In what follows, we will connect the structure of the projection matrices introduced in (52) to the particular compressed quantities of the original QBTI system. For this, consider that the projection matrices are of required dimension, i.e., $\mathbf{V}, \mathbf{W} \in \mathbb{C}^{\mathcal{N} \times r}$ with $\mathcal{N} = n^2 + n$, corresponding to the dimension of the bilinear model computed via Carleman's linearization (for $N = 2$). Additionally, the projection matrices are split as:

$$\mathbf{V} = \begin{bmatrix} \mathbf{V}_1 \\ \mathbf{V}_2 \end{bmatrix}, \ \mathbf{W} = \begin{bmatrix} \mathbf{W}_1 \\ \mathbf{W}_2 \end{bmatrix}, \tag{54}$$

with $\mathbf{V}_1, \mathbf{W}_1 \in \mathbb{C}^{n \times r}$ and $\mathbf{V}_2, \mathbf{W}_2 \in \mathbb{C}^{n^2 \times r}$. These two matrices are used to construct the realization of a bilinear system of reduced dimension computed by means of Petrov–Galerkin projections applied to the bilinear model in (31):

$$\boldsymbol{\Sigma}_{\mathrm{B}}^{\otimes} : \begin{cases} \dot{\hat{\mathbf{x}}}^{\otimes}(t) = \hat{\mathbf{A}}^{\otimes}\hat{\mathbf{x}}^{\otimes}(t) + \hat{\mathbf{N}}^{\otimes}\hat{\mathbf{x}}^{\otimes}(t)\mathbf{u}(t) + \hat{\mathbf{B}}^{\otimes}\mathbf{u}(t), \\ \quad \hat{\mathbf{y}}^{\otimes} = \hat{\mathbf{C}}^{\otimes}\hat{\mathbf{x}}^{\otimes}(t), \end{cases} \tag{55}$$

where the matrices are computed as follows:

$$\begin{aligned} \hat{\mathbf{A}}^{\otimes} &= \begin{bmatrix} \mathbf{W}_1^* & \mathbf{W}_2^* \end{bmatrix} \begin{bmatrix} \mathbf{A} & \mathbf{Q} \\ \mathbf{0} & \mathbf{A} \otimes \mathbf{I}_n + \mathbf{I}_n \otimes \mathbf{A} \end{bmatrix} \begin{bmatrix} \mathbf{V}_1 \\ \mathbf{V}_2 \end{bmatrix} \\ &= \mathbf{W}_1^*\mathbf{A}\mathbf{V}_1 + \mathbf{W}_1^*\mathbf{Q}\mathbf{V}_2 + \mathbf{W}_2^*(\mathbf{A} \otimes \mathbf{I}_n + \mathbf{I}_n \otimes \mathbf{A})\mathbf{V}_2, \\ \hat{\mathbf{N}}^{\otimes} &= \begin{bmatrix} \mathbf{W}_1^* & \mathbf{W}_2^* \end{bmatrix} \begin{bmatrix} \mathbf{N} & \mathbf{0} \\ \mathbf{B} \otimes \mathbf{I}_n + \mathbf{I}_n \otimes \mathbf{B} & \mathbf{N} \otimes \mathbf{I}_n + \mathbf{I}_n \otimes \mathbf{N} \end{bmatrix} \begin{bmatrix} \mathbf{V}_1 \\ \mathbf{V}_2 \end{bmatrix} \\ &= \mathbf{W}_1^*\mathbf{N}\mathbf{V}_1 + \mathbf{W}_2^*(\mathbf{B} \otimes \mathbf{I}_n + \mathbf{I}_n \otimes \mathbf{B})\mathbf{V}_1 + \mathbf{W}_2^*(\mathbf{N} \otimes \mathbf{I}_n + \mathbf{I}_n \otimes \mathbf{N})\mathbf{V}_2, \end{aligned} \tag{56}$$

while the other matrices are given by:

$$\begin{aligned} \hat{\mathbf{B}}^{\otimes} &= \begin{bmatrix} \mathbf{W}_1^* & \mathbf{W}_2^* \end{bmatrix} \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix} = \mathbf{W}_1^* \mathbf{B}, \ \hat{\mathbf{C}}^{\otimes} = \begin{bmatrix} \mathbf{C} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{V}_1 \\ \mathbf{V}_2 \end{bmatrix} = \mathbf{C}\mathbf{V}_1, \\ \hat{\mathbf{E}}^{\otimes} &= \begin{bmatrix} \mathbf{W}_1^* & \mathbf{W}_2^* \end{bmatrix} \mathbf{I}_{n^2+n} \begin{bmatrix} \mathbf{V}_1 \\ \mathbf{V}_2 \end{bmatrix} = \mathbf{W}_1^* \mathbf{V}_1 + \mathbf{W}_2^* \mathbf{V}_2. \end{aligned} \tag{57}$$

It is to be noted that matrices $\mathbf{V}$ and $\mathbf{W}$ in (54) can indeed be chosen so that the relation $\mathbf{W}^*\mathbf{V} = \mathbf{I}_r$ holds. This implies that $\mathbf{W}_1^*\mathbf{V}_1 + \mathbf{W}_2^*\mathbf{V}_2 = \mathbf{I}_r$ and hence, it follows that $\hat{\mathbf{E}}^{\otimes} = \mathbf{I}_r$. The formulas in (56) and (57) show that the system matrices of the reduced-order bilinear model $\mathbf{\Sigma}_{\mathrm{B}}^{\otimes}$ in (55) can be written as sums of appropriately compressed quantities, as enumerated below:

1. Using the projection matrices $\mathbf{W}_1$ and $\mathbf{V}_1$ by projecting the quantities of the bilinear sub-block of the original QBTI system: $\mathbf{W}_1^*\mathbf{A}\mathbf{V}_1, \mathbf{W}_1^*\mathbf{N}\mathbf{V}_1, \mathbf{W}_1^*\mathbf{B}$, and $\mathbf{C}\mathbf{V}_1$.
2. Using the projection matrices $\mathbf{W}_2$ and $\mathbf{V}_2$: $\mathbf{W}_2^*(\mathbf{A} \otimes \mathbf{I}_n + \mathbf{I}_n \otimes \mathbf{A})\mathbf{V}_2$, and $\mathbf{W}_2^*(\mathbf{N} \otimes \mathbf{I}_n + \mathbf{I}_n \otimes \mathbf{N})\mathbf{V}_2$.
3. Using the projection matrices $\mathbf{W}_2$ and $\mathbf{V}_1$ or $\mathbf{W}_1$ and $\mathbf{V}_2$ (mixed terms): $\mathbf{W}_2^*(\mathbf{B} \otimes \mathbf{I}_n + \mathbf{I}_n \otimes \mathbf{B})\mathbf{V}_1$ and $\mathbf{W}_1^*\mathbf{Q}\mathbf{V}_2$.

Hence, we demonstrated that the problem of projection-based model reduction of BTI models of bi-linearized QBTI systems (using Carleman's approach), boils down to block-wise reduction of the original QBTI models.

5. Numerical Examples

5.1. The Viscous Burgers' Equation

We start with an illustrative example to show the quadratic and bilinear modeling and reduction concepts proposed in the previous sections: *the viscous Burgers' equation*. The equation was first introduced by Harry Bateman in 1915 and later studied by Johannes Martinus Burgers in 1948 [47]. In the sequel, we follow the description from [42]. As was stated there, the dynamics are characterized by the following PDE:

$$\frac{\partial v(x,t)}{\partial t} + v(x,t)\frac{\partial v(x,t)}{\partial x} = \frac{\partial}{\partial x}\Big(\nu \frac{v(x,t)}{\partial x}\Big), \quad (x,t) \in (0,1) \times (0,T)\ , \tag{58}$$

subject to the initial and boundary conditions given (in our application) by:

$$v(x,0) = f(x),\ x \in [0,1],\ v(0,t) = u(t),\ v(1,t) = 0,\ t \geqslant 0\ .$$

This PDE commonly occurs in the area of fluid dynamics where it is used for modeling gas dynamics and traffic flow. The solution $v(x,t)$ can be interpreted as a function describing the velocity at (x,t). In general, the viscosity coefficient (here denoted by $\nu(x,t)$) might depend on space and time as well.

In what follows, it is assumed that the viscosity coefficient ν is constant w.r.t. the time variable. Furthermore, consider a zero initial condition on the system, i.e., $f(x) = 0$. Finally, we assume that the left boundary is subject to a control.

A spatial discretization of Equation (58), is performed, by means of a classical first-order Euler approximation scheme with equidistant step size $h = \frac{1}{n+1}$, Here, n denotes the number of interior points of the interval $(0,1)$. The following spacial derivative approximations are used:

$$\frac{\partial v}{\partial x} = \frac{v(x+h) - v(x)}{h}, \ \frac{\partial^2 v}{\partial x^2} = \frac{v(x+h) - 2v(x) + v(x-h)}{h^2}.$$

By substituting these formulas into (58), we manage to obtain the following QBTI characterizing the original dynamics:

$$\dot{v}_k(t) = \begin{cases} -\frac{v_1(t)v_2(t)}{2h} + \frac{\nu}{h^2}(v_2(t) - 2v_1(t)) + \left(\frac{v_1(t)}{2h} + \frac{\nu}{h^2}\right)u(t), & k = 1, \\ -\frac{v_k(t)}{2h}(v_{k+1}(t) - v_{k-1}(t)) + \frac{\nu}{h^2}(v_{k+1}(t) - 2v_k(t) + v_{k-1}(t)), & 2 \leqslant k \leqslant n-1, \\ -\frac{v_n(t)v_{n-1}(t)}{2h} + \frac{\nu}{h^2}(-2v_n(t) + 2v_{n-1}(t)), & k = n. \end{cases} \quad (59)$$

The observed output is chosen to be the average o fall states, i.e., $y(t) = \frac{1}{n}\sum_{k=1}^{n} v_k(t)$. More precisely, for $n = 3$, we can write that

$$\begin{aligned} \dot{v}_1(t) &= -\frac{v_1(t)v_2(t)}{2h} + \frac{\nu}{h^2}(v_2(t) - 2v_1(t)) + \left(\frac{v_1(t)}{2h} + \frac{\nu}{h^2}\right)u(t), \\ \dot{v}_2(t) &= -\frac{v_2(t)}{2h}(v_3(t) - v_1(t)) + \frac{\nu}{h^2}(v_3(t) - 2v_2(t) + v_1(t)), \\ \dot{v}_3(t) &= -\frac{v_3(t)v_2(t)}{2h} + \frac{\nu}{h^2}(-2v_3(t) + 2v_2(t)), \text{ and } y(t) = \frac{v_1(t) + v_2(t) + v_3(t)}{3}. \end{aligned} \quad (60)$$

In what follows, for the numerical experiments, we use $n = 90$, and we put together a QBTI system in 90 state variables described by equations in (59). The time-domain full-order solution of this system is depicted in Figure 1 on the temporal-spatial grid $[0,1] \times [0,0.2]$.

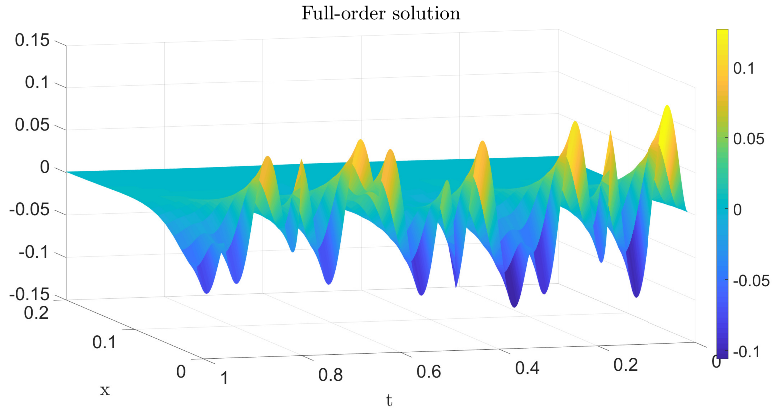

Figure 1. The solution of the original full-order QBTI system.

Next, by making use of the Carleman linearization technique, we approximate the above nth QBTI system $\mathbf{\Sigma}_{\mathrm{QB}}$ described by Equation (59), with a bilinear system of order $\mathcal{N} = n^2 + n = 8190$. We follow the procedure discussed in the previous sections. We denote with $\mathbf{\Sigma}_{\mathrm{B}}^{\otimes}$ the 8190th order initial bilinear system obtained by means of the Carleman linearization. The model $\mathbf{\Sigma}_{\mathrm{B}}^{\otimes}$ is reduced using the bilinear Loewner framework to obtain $\hat{\mathbf{\Sigma}}_{\mathrm{B}}^{(1)}$ of dimension $r = 34$ by following the procedure in Section 4.

As discussed there, one needs to start with a grid of interpolation points; in this case, we choose 300 logarithmically spaced points in the interval $[10^{-2}, 10^{2}]\imath$ with $\imath = \sqrt{-1}$. Then, by appropriately partitioning the data (sample points and evaluations of the transfer functions at this points) and then extracting the dominant features, we compute a reduced-order BTI model of dimension $r = 34$.

Here, this value was chosen with respect to the singular value decay of the Loewner matrix $\mathbb{L}$, as shown in the left plot of Figure 2 (the tolerance value for applying truncation was chosen as $\tau = 10^{-14}$). Additionally, the poles of the original QBTI and of the reduced-order BTI are displayed in the right plot of Figure 2. The former (QBTI) are all located on the negative real axis, while the latter (BTI) tends to have the same behavior, with exceptions given by some complex conjugate poles.

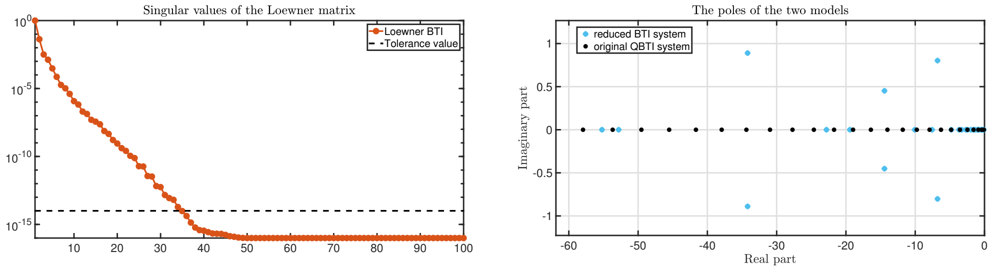

Figure 2. The singular value decay of the Loewner matrix + tolerance value (**left**) and the poles of the original QBTI system and of the reduced-order BTI system (**right**).

In what follows, we perform time-domain simulations in order to asses the quality of approximation attained by the reduced-order model. We choose $T = 10$ seconds as the end time of the simulation and excite the large-scale BTI system with a control input $u(t) = 1/5 \cdot \text{square}(20t) \cdot \sin(16\pi t)e^{-t/2}$, where square$(t)$ generates a square wave signal with period 2π for the elements of the time array t (using the MATLAB specifications).

We compute the observed output as the average value of all state entries (in the original setup) and compare the outputs of the two bilinear systems $\mathbf{\Sigma}_{\text{B}}^{\otimes}$ and of $\hat{\mathbf{\Sigma}}_{\text{B}}^{(1)}$. The left plot of Figure 3 contains the depiction of the two outputs, while the right plot depicts the magnitude of deviation (absolute error) between the two signals. As can be observed, a high approximation quality was attained.

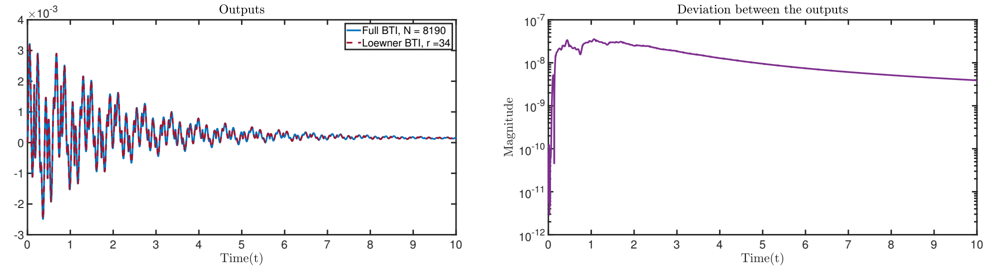

Figure 3. The output of the original large-scale and of the reduced-order BTI systems (**left**) and the absolute error, i.e., the magnitude of the deviation (**right**).

5.2. A Nonlinear RC Ladder Circuit

The next test-case is a nonlinear resistor-capacitor (RC) ladder electrical circuit first introduced by [41]. Various variants of this model were used in many other MOR works, such as [34] or [42]. The nonlinear first-order system under study models a RC network; the nonlinearity is, hence, explained by nonlinear resistors consisting of a parallel connection between classical resistors with a diode as depicted in the schematic in Figure 4.

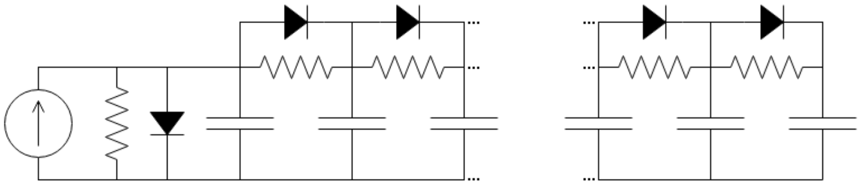

Figure 4. Schematic of the RC ladder circuit from [41].

As shown in the original contribution [41], the underlying model of ordinary differential equations forms a SISO nonlinear time-invariant system characterized by:

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \vdots \\ \dot{x}_k(t) \\ \vdots \\ \dot{x}_n(t) \end{bmatrix} = \begin{bmatrix} -g(x_1(t)) - g(x_1(t) - x_2(t)) \\ g(x_1(t) - x_2(t)) - g(x_2(t) - x_3(t)) \\ \vdots \\ g(x_{k-1}(t) - x_k(t)) - g(x_k(t) - x_{x+1}(t)) \\ \vdots \\ g(x_{n-1}(t) - x_n(t)) \end{bmatrix} + \begin{bmatrix} u(t) \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \tag{61}$$

where the mapping g is defined as $g : \mathbb{R} \to \mathbb{R}$ defined as $g(x_i) = g_D(x_i) + x_i$, which combines the effects of a diode and a resistor. The observed output is given by the first state variable $y(t) = x_1(t)$. The non-linearity g_D models a diode as a nonlinear resistor, based on the classical Shockley model:

$$g_D(x_i) = i_S(\exp(u_P x_i) - 1), \tag{62}$$

with material parameters $i_S > 0$ and $u_P > 0$. For this benchmark, the parameters were selected as follows: i_S = 1 and $u_P = 40$ as in [41]. By substituting these values into (62), we find that $g_D(x_i) = \exp(40x_i) - 1$, and hence it follows that $g(x_i) = \exp(40x_i) + x_i - 1$.

Analogous to the example in Section 2.2.1, the original nonlinear RC-ladder model can be lifted into an equivalent QBTI model. For more details of the exact procedure, we refer the reader to Section 5.5.1 in [42].

The lifted QBTI model from before, which is equivalent to the original nonlinear system, is then transformed into a bilinear system by means of Carleman linearization as originally shown in [41] and later in [42]. Consequently, the resulting BTI system has dimensions $n^2 + n$ with n denoting the number of circuit blocks of the original system. We follow here the procedures outlined in Sections 2 and 3.

We choose $n = 50$, and hence the original QBTI model $\mathbf{\Sigma}_{\text{QB}}$ derived from (61) is of dimension $n_q = 100$ and the bilinear model $\mathbf{\Sigma}_{\text{B}}^{\otimes}$ derived from (61) is of dimension $n_b = 2550$. Next, we choose 100 logarithmically spaced points in the interval $[10^0, 10^2]\imath$. Then, by following the extended Loewner approaches, we compute a reduced-order BTI model of dimension $r_b = 39$ and a reduced-order QBTI model of dimension $r_q = 17$. As before, these values were chosen with respect to the singular value decay of Loewner matrices, as shown in the left plot of Figure 5 (the tolerance value was $\tau = 10^{-14}$). Additionally, the poles of the original QBTI and BTI systems are displayed in the right plot of Figure 5.

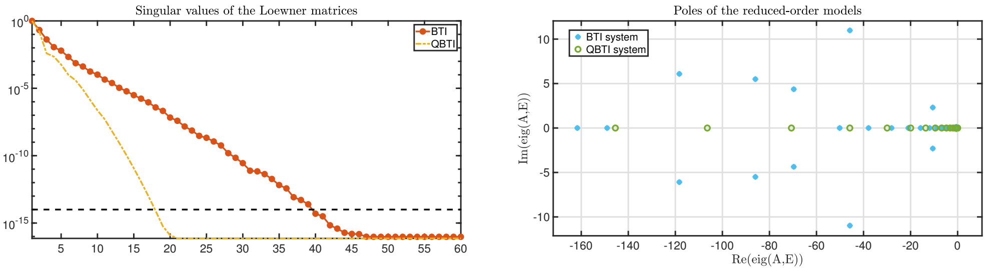

Figure 5. The singular value decay of the Loewner matrices of the BTI and QBTI reduced systems (**left**) and the poles of the reduced-order BTI and QBTI models (**right**).

Next, we perform time-domain simulations by using as the control input, the signal $u(t) = 0.2 \cdot \text{sawtooth}(60t) \cdot \sin(100t)$. Here, sawtooth$(t)$ generates a sawtooth wave signal with period 2π for the elements of the time array t (using the MATLAB specifications). We choose the end simulation time to be $T = 1$, and a discretization step of $\Delta t = 10^{-3}$. The three output signals are illustrated in the left plot of Figure 6, i.e., that of the original nonlinear system in (61), and the outputs of the reduced-order BTI and QBTI models computed via the Loewner approach. As can be seen, both of the latter two are in good accordance with the original output.

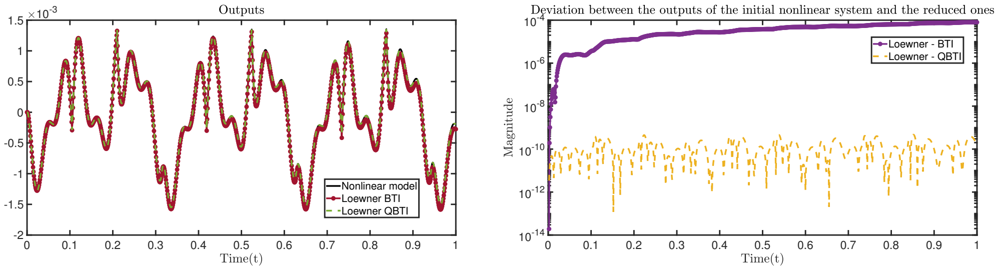

Figure 6. The output corresponding to the original nonlinear system, and to the Loewner BTI and QBTI systems (**left**) and the magnitude of the absolute errors between the outputs (**right**).

Additionally, in the right plot of Figure 6, we also depict the magnitude of the absolute approximation errors between the output of the original nonlinear model and the outputs of the two reduced-order systems. As can be clearly observed, the QBTI model provides a far better approximation quality that that of the BTI. This is indeed to be expected, since the QB reformulation of analytic nonlinear systems is exact, while Carleman's linearization introduces errors in the enforced truncation.

6. Conclusions and Outlook

In this work, we addressed the problem of approximating generic nonlinear system by means of enforcing a specific structure of the reduced-order models. Bilinear and quadratic-bilinear systems accomplish precisely this goal, either by approximation or by exact reformulation. Additionally, we dealt with complexity reduction of large-scale lifted models of nonlinear systems.

The method used—the Loewner framework—is a data-driven approach that requires samples of generalized transfer functions, which are appropriately defined for both bilinear and quadratic-bilinear systems. We showed explicit relations between these mappings, as well as between the matrices and the poles of the reduced-order models. The theoretical considerations were illustrated by two numerical examples. These included a classical application for computational fluid dynamics (the first), and a nonlinear circuit from the field of electronic engineering. Both numerical test-cases demonstrated connections between the two types of reformulations as indicated by the theory and technical results.

Finally, it was shown that formulating general nonlinear systems as QBTI systems is indeed the approach to be preferred (for applying MOR methods, such as the LF). However, using the reformulation as BTI systems also has its advantages: the simpler structure allows many connections to classical LTI systems theory.

Future research endeavors may potentially include studying the connections between the original QBTI and the BTI lifted model (by Carleman's approach) in the time domain (by explicitly comparing the Markov parameters of both models), analyzing lifting techniques for recasting nonlinear systems with polynomial nonlinearities as QBTIs, and imposing stability preservation for the reduced-order models.

Funding: The author gratefully acknowledges the support of the Max Planck Society, and in particular, that of the Max Planck Digital Library which has covered the article processing charges.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Antoulas, A.C. *Approximation of Large-Scale Dynamical Systems*; SIAM: Philadelphia, PA, USA, 2005.
2. Benner, P.; Ohlberger, M.; Cohen, A.; Willcox, K. *Model Reduction and Approximation*; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2017.
3. Antoulas, A.C.; Beattie, C.A.; Gugercin, S. *Interpolatory Methods for Model Reduction*; SIAM: Philadelphia, PA, USA, 2020.
4. Baur, U.; Benner, P.; Feng, L. Model order reduction for linear and nonlinear systems: A system-theoretic perspective. *Arch. Comput. Methods Eng.* **2014**, *21*, 331–358. [CrossRef]
5. Benner, P.; Breiten, T. Two-sided projection methods for nonlinear model order reduction. *SIAM J. Sci. Comput.* **2015**, *37*, B239–B260. [CrossRef]
6. Ahmad, M.I.; Benner, P.; Jaimoukha, I. Krylov Subspace Projection Methods for Model Reduction of Quadratic-Bilinear Systems. *IET Control Theory Appl.* **2016**, *10*, 2010–2018. [CrossRef]
7. Benner, P.; Goyal, P. Balanced Truncation Model Order Reduction for Quadratic-Bilinear Control Systems. *arXiv* **2017**, arXiv:1705.00160.
8. Benner, P.; Goyal, P.; Gugercin, S. $\mathcal{H}_2$-Quasi-Optimal Model Order Reduction for Quadratic-Bilinear Control Systems. *SIAM J. Matrix Anal. Appl.* **2018**, *39*, 983–1032. [CrossRef]
9. Ruberti, A.; Isidori, A.; D'Alessandro, P. *Theory of Bilinear Dynamical Systems*; Springer: Berlin/Heidelberg, Germany, 1972.
10. Freund, R. Krylov-subspace methods for reduced-order modeling in circuit simulation. *J. Comput. Appl. Math.* **2000**, *123*, 395–421. [CrossRef]
11. Zhang, L.; Lam, J. On $\mathcal{H}_2$ model reduction of bilinear systems. *Automatica* **2002**, *38*, 205–216. [CrossRef]
12. Phillips, J.R. Projection-Based Approaches for Model Reduction of Weakly Nonlinear, Time-Varying Systems. *IEEE Trans. Comput.-Aided Des. Integr. Circuits* **2003**, *22*, 171–187. [CrossRef]
13. Bai, Z.; Skoogh, D. A projection method for model reduction of bilinear dynamical systems. *Linear Algebra Appl.* **2006**, *415*, 406–425. [CrossRef]
14. Condon, M.; Ivanov, R. Krylov subspaces from bilinear representations of nonlinear systems. *COMPEL-Int. J. Comp. Math. Electr. Electron. Eng.* **2007**, *26*, 399–406. [CrossRef]
15. Breiten, T.; Damm, T. Krylov subspace methods for model order reduction of bilinear control systems. *Syst. Control Lett.* **2010**, *59*, 443–450. [CrossRef]
16. Benner, P.; Damm, T. Lyapunov equations, energy functionals, and model order reduction of bilinear and stochastic systems. *SIAM J. Control Optim.* **2011**, *49*, 686–711. [CrossRef]
17. Benner, P.; Breiten, T. Interpolation-based $\mathcal{H}_2$-model reduction of bilinear control systems. *SIAM J. Matrix Anal. Appl.* **2012**, *33*, 859–885. [CrossRef]
18. Hartmann, C.; Schäfer-Bung, B.; Thöns-Zueva, A. Balanced averaging of bilinear systems with applications to stochastic control. *SIAM J. Matrix Anal. Appl.* **2013**, *51*, 2356–2378. [CrossRef]
19. Flagg, G.; Gugercin, S. Multipoint Volterra Series Interpolation and $\mathcal{H}_2$ Optimal Model Reduction of Bilinear Systems. *SIAM J. Matrix Anal. Appl.* **2015**, *36*, 549–579. [CrossRef]
20. Antoulas, A.C.; Gosea, I.V.; Ionita, A.C. Model reduction of bilinear systems in the Loewner framework. *SIAM J. Sci. Comput.* **2016**, *38*, B889–B916. [CrossRef]
21. 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; Birkhäuser: Cham, Switzerland, 2021; Volume 171, pp. 3–30. [CrossRef]
22. Peherstorfer, B.; Willcox, K. Data-driven operator inference for nonintrusive projection-based model reduction. *Comput. Methods Appl. Mech. Eng.* **2016**, *306*, 196–215. [CrossRef]
23. Gosea, I.V.; Antoulas, A.C. Data-driven model order reduction of quadratic-bilinear systems. *Numer. Linear Algebra Appl.* **2018**, *25*, e2200. [CrossRef]
24. Benner, P.; Goyal, P.; Kramer, B.; Peherstorfer, B.; Willcox, K. Operator Inference for Non-Intrusive Model Reduction of Systems with Non-Polynomial Nonlinear Terms. *Comput. Methods Appl. Mech. Eng.* **2020**, *372*, 113433. [CrossRef]
25. Qian, E.; Krämer, B.; Peherstorfer, B.; Willcox, K. Lift & Learn: Physics-informed machine learning for large-scale nonlinear dynamical systems. *Phys. D Nonlinear Phenom.* **2020**, *406*, 132401. [CrossRef]
26. Gosea, I.V.; Karachalios, D.S.; Antoulas, A.C. Learning reduced-order models of quadratic dynamical systems from input–output data. In Proceedings of the 2021 IEEE European Control Conference (ECC), Delft, The Netherlands, 29 June–2 July 2021; pp. 1426–1431.
27. Kutz, J.N.; Brunton, S.L.; Brunton, B.W.; Proctor, J.L. *Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems*; Society of Industrial and Applied Mathematics: Philadelphia, PA, USA, 2016. [CrossRef]
28. Alla, A.; Kutz, J.N. Nonlinear Model Order Reduction via Dynamic Mode Decomposition. *SIAM J. Sci. Comput.* **2017**, *39*, B778–B796. [CrossRef]
29. Goldschmidt, A.; Kaiser, E.; Dubois, J.L.; Brunton, S.L.; Kutz, J.N. Bilinear dynamic mode decomposition for quantum control. *New J. Phys.* **2021**, *23*, 033035. [CrossRef]

30. Gosea, I.V.; Pontes Duff, I. Toward fitting structured nonlinear systems by means of dynamic mode decomposition. In *Model Reduction of Complex Dynamical Systems*; International Series of Numerical Mathematics; Birkhäuser: Cham, Switzerland, 2021; Volume 171, pp. 53–74. [CrossRef]
31. Mayo, A.J.; Antoulas, A.C. A framework for the generalized realization problem. *Linear Algebra Its Appl.* **2007**, *426*, 634–662. [CrossRef]
32. Carleman, T. Application de la théories des équations intégrales linéaires aux systèmes d'équations différentielles non linéaires. *Acta. Math.* **1932**, *59*, 63–87. [CrossRef]
33. McCormick, G.P. Computability of global solutions to factorable nonconvex programs: Part I—Convex underestimating problems. *Math. Program.* **1976**, *10*, 147–175. [CrossRef]
34. Gu, C. A projection-based nonlinear model order reduction approach using quadratic-linear representation of nonlinear systems. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2011**, *30*, 1307–1320. [CrossRef]
35. Kramer, B.; Willcox, K.E. Nonlinear model order reduction via lifting transformations and proper orthogonal decomposition. *AIAA J.* **2019**, *57*, 2297–2307. [CrossRef]
36. Gu, C. Model Order Reduction of Nonlinear Dynamical Systems. Ph.D. Thesis, EECS Department, University of California, Berkeley, CA, USA, 2012.
37. Gilbert, E. Functional expansions for the response of nonlinear differential systems. *IEEE Trans. Autom. Control* **1977**, *22*, 909–921. [CrossRef]
38. Rugh, W.J. *Nonlinear System Theory—The Volterra/Wiener Approach*; The Johns Hopkins University Press: Baltimore, MD, USA, 1981.
39. Gosea, I.V. Model Order Reduction of Linear and Nonlinear Systems in the Loewner Framework. Ph.D. Thesis, Department of Electrical Engineering, Jacobs University Bremen, Bremen, Germany, 2017.
40. Asif, M.M.A.; Ahmad, M.I.; Benner, P.; Feng, L.; Stykel, T. Implicit Higher-Order Moment Matching Technique for Model Reduction of Quadratic-bilinear Systems. *J. Frankl. Inst.* **2021**, *358*, 2015–2038. [CrossRef]
41. Chen, Y. Model Reduction for Nonlinear Systems. Master's Thesis, M.I.T, Boston, MA, USA, 1999.
42. Breiten, T. Interpolatory Methods for Model Reduction of Large-Scale Dynamical Systems. Ph.D. Dissertation, Department of Mathematics, Otto-von-Guericke University, Magdeburg, Germany, 2013. [CrossRef]
43. Antoulas, A.C.; Lefteriu, S.; Ionita, A.C. A tutorial introduction to the Loewner framework for model reduction. In *Model Reduction and Approximation*; SIAM: Philadelphia, PA, USA, 2017; Chapter 8, pp. 335–376.
44. Karachalios, D.S.; Gosea, I.V.; Antoulas, A.C. The Loewner framework for system identification and reduction. In *Model Reduction Handbook: Volume I: System- and Data-Driven Methods and Algorithms*; Handbook on Model Reduction; Benner, P., Grivet-Talocia, S., Quarteroni, A., Rozza, G., Schilders, W.H.A., Silveira, L.M., Eds.; De Gruyter: Berlin, Germany, 2020; Volume 1.
45. Gosea, I.V.; Poussot-Vassal, C.; Antoulas, A.C. Data-driven modeling and control of large-scale dynamical systems in the Loewner framework: methodology and applications. In *Numerical Control: Part A*; Handbook on Numerical Analysis; Zuazua, E., Trelat, E., Eds.; Elsevier: Amsterdam, The Netherlands, 2022.
46. Antoulas, A.C.; Gosea, I.V.; Heinkenschloss, M. On the Loewner Framework for Model Reduction of Burgers' Equation. In *Active Flow and Combustion Control*; King, R., Ed.; Springer: Cham, Switzerland, 2019; pp. 255–270.
47. Burgers, J.M. A mathematical model illustrating the theory of turbulence. In *Advances in Applied Mechanics*; Elsevier: Amsterdam, The Netherlands, 1948; Volume 1, pp. 171–199.