arXiv:2403.00646v1 [cs.LG] 1 Mar 2024

Stability-Certified Learning of Control Systems with Quadratic Nonlinearities

Igor Pontes Duff[†] Pawan Goyal[‡] Peter Benner[††]

[†] *Max Planck Institute of Dynamics of Complex Technical Systems*
Email: pontes@mpi-magdeburg.mpg.de, ORCID: 0000-0001-6433-6142
[‡] *Max Planck Institute of Dynamics of Complex Technical Systems*
Email: goyalp@mpi-magdeburg.mpg.de, ORCID: 0000-0003-3072-7780
[††] *Max Planck Institute of Dynamics of Complex Technical Systems, Otto-von-Guericke University Magdeburg*
Email: benner@mpi-magdeburg.mpg.de, ORCID: 0000-0003-3362-4103

Abstract: This work primarily focuses on an operator inference methodology aimed at constructing low-dimensional dynamical models based on a priori hypotheses about their structure, often informed by established physics or expert insights. Stability is a fundamental attribute of dynamical systems, yet it is not always assured in models derived through inference. Our main objective is to develop a method that facilitates the inference of quadratic control dynamical systems with inherent stability guarantees. To this aim, we investigate the stability characteristics of control systems with energy-preserving nonlinearities, thereby identifying conditions under which such systems are bounded-input bounded-state stable. These insights are subsequently applied to the learning process, yielding inferred models that are inherently stable by design. The efficacy of our proposed framework is demonstrated through a couple of numerical examples.

Keywords: Stability, control systems, scientific machine learning, operator inference, Lyapunov function, energy-preserving systems.

Novelty statement:

- Learning stability-certified control systems with quadratic nonlinearities.
- Utilizing a stable matrix parameterization for the certification.
- Bounded-input bounded-state stability for quadratic systems with control is shown under parametrization assumptions.
- Several numerical examples demonstrate the stability-certified results of the learned models.

1 Introduction

Significant developments have been made recently in the fields of learning dynamical systems from data, driven by the availability of large amount of data sets and the demand across various applications such as robotics, epidemiology, and climate science. First principles model construction often falls short due to the complexity of these applications, paving the way for data-driven methodologies. Among these, the *Dynamic Mode Decomposition* (DMD) together with Koopman operator theory is a common used approach, since it allows to handle complex dynamics by linearizing non-linear systems in a high-dimensional (or even infinite-dimensional) observable space, see [1,2]. Additionally, the DMD framework was extended to allow handling control inputs in [3].

Recent advancements have also seen the emergence of *Sparse Identification of Nonlinear Dynamics* (SINDy) for discovering the essential nonlinear terms from extensive libraries through sparse regression, offering an alternative perspective on nonlinear system identification ([4]). Moreover, the integration of prior knowledge, particularly physics-based, into learning frameworks has also attracted attentions, particularly through the operator inference (OpInf) technique ([5]), which aim at the construction of data-driven (reduced-order) models by focusing on quadratic (or polynomial) dynamics.

Many physical phenomena demonstrate stability behavior, i.e., their state variables evolve in a bounded region of the state space over extended time horizons. Consequently, the accurate representation of these phenomena necessitates stable differential equations, which is also essential for the long-term numerical integration. However, the critical aspect of stability is often not addressed while learning dynamical systems. In this work, our main focus is to learn quadratic control models of the form

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{H}\left(\mathbf{x}(t) \otimes \mathbf{x}(t)\right) + \mathbf{B}\mathbf{u}(t), \quad \mathbf{x}(0) = \mathbf{x}_0, \tag{1}$$

where $\mathbf{x}(t) \in \mathbb{R}^n$ is the state vector, $\mathbf{u}(t) \in \mathbb{R}^m$ represents the inputs, and $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{H} \in \mathbb{R}^{n\times n^2}$, and $\mathbf{B} \in \mathbb{R}^{n\times m}$ are the system matrices. It is worth mentioning that quadratic models emerge inherently within discretized models of, e.g., fluid mechanics. Furthermore, through a process known as lifting transformation, smooth nonlinear dynamical systems can be transformed into quadratic dynamical systems [6]. Also, this transformation was leveraged to operator inference in [7] to learn physics-based quadratic models for a given data set, and identifying such a lifted transformation using neural networks was investigated in [8].

Our prior work concentrated on imposing stability on learned (uncontrolled) linear and quadratic models, see [9, 10]. Therein, we introduced methodologies to ensure local and global stability, leveraging parametrizations for stable matrices and energy-preserving Hessians to enforce stability by design, mainly inspired by the results on energy-preserving nonlinearities proposed in [11]. The approach [9] not only addressed computational challenges but also bypassed the limitations of conventional stability constraints, thus providing a robust foundation for stable dynamical system modeling.

Building upon this foundation, our current work extends these concepts to quadratic systems with control inputs, enhancing their applicability in controlled environments and broadening the scope of data-driven dynamical system learning. To this aim, firstly in Section 2, we recapitulate the operator inference framework for learning quadratic control systems from (possibly) high dimensional data sets. Then, in Section 3, we revisit the parameterizations of stable matrices and energy-preserving Hessians used in [10] to impose stability on learned uncontrolled quadratic models. Then, we establish our main result, which consists of parametrizations of quadratic control systems that are guaranteed to be bounded-input bounded-state stable. Section 4 leverages the proposed parametrization to learn stable quadratic control systems. Subsequently, Section 5 extends the presented results to Lyapunov functions of generalized quadratic form. Finally, Section 6 illustrates the proposed methodology using a coupled of numerical examples, and Section 7 concludes the manuscript with summary and open avenues.

2 Model Inference of quadratic control systems

We now present a brief overview of the Operator Inference (OpInf) methodology ([5]), focusing specifically on systems with quadratic nonlinearities. Our discussion starts with learning quadratic models from high-dimensional data and considers dynamical system having a control input.

We proceed to define the problem of deriving low-dimensional dynamical systems from high-dimensional data originated from a nonlinear (psossibly quadratic) control system

$$\dot{\mathbf{y}}(t) = \mathbf{f}(\mathbf{y}(t), \mathbf{u}(t)), \quad t \geq 0,$$

where $\mathbf{y}(t) \in \mathbb{R}^N$ is the state vector and $\mathbf{u}(t) \in \mathbb{R}^m$ is an control input. We assume the availability of the state snapshots $\mathbf{y}(t)$ at time $t \in \{t_0, t_1, \ldots, t_{\mathcal{N}}\}$. These snapshots are aggregated into the snapshot matrix:

$$\mathbf{Y} = \left[\mathbf{y}(t_0), \ldots, \mathbf{y}(t_{\mathcal{N}})\right] \in \mathbb{R}^{N\times\mathcal{N}}. \tag{2}$$

Additionally, we assume that the dynamics is actuated by an input function $\mathbf{u}(t) \in \mathbb{R}^m$ and that we have access to input snapshots at the same time steps, i.e., $\mathbf{u}(t_0), \ldots, \mathbf{u}(t_{\mathcal{N}})$ which can be aggregated into the matrix:

$$\mathbf{U} = \left[\mathbf{u}(t_0), \ldots, \mathbf{u}(t_{\mathcal{N}})\right] \in \mathbb{R}^{m \times \mathcal{N}}.$$

Although the dynamics of state $\mathbf{y}(t)$ are inherently N-dimensional, it can often be effectively represented in a low-dimensional subspace. As a result, we can aim at learning dynamics using the coordinate systems in the low-dimensional subspace. To that end, we identify a low-dimensional representation for $\mathbf{y}(t)$, which is done by determining the projection matrix $\mathbf{V} \in \mathbb{R}^{N \times n}$, derived from the singular value decomposition of $\mathbf{Y}$ and selecting the n most dominant left singular vectors. This leads to the computation of the reduced state trajectory as follows:

$$\mathbf{X} = \mathbf{V}^\top \mathbf{Y}, \tag{3}$$

where $\mathbf{X} := \left[\mathbf{x}(t_0), \ldots, \mathbf{x}(t_{\mathcal{N}})\right]$ with $\mathbf{x}(t_i) = \mathbf{V}^\top \mathbf{y}(t_i)$. With our quadratic model hypothesis, we then aim to learn the system operators from the available data. Precisely, our goal is to learn a quadratic control system of the form (1), where $\mathbf{A}$, $\mathbf{H}$, and $\mathbf{B}$ are the system operators.

Next, we cast the inference problem, which is as follows. Having the low-dimensional trajectories $\{\mathbf{x}(t_0), \ldots, \mathbf{x}(t_{\mathcal{N}})\}$ and the input snapshots $\{\mathbf{u}(t_0), \ldots, \mathbf{u}(t_{\mathcal{N}})\}$, we aim to learn operators $\mathbf{A}$, $\mathbf{H}$, and $\mathbf{B}$ in (1). At the moment, let us also assume to have (estimated) the derivative information of $\mathbf{x}$ at time $\{t_0, \ldots, t_{\mathcal{N}}\}$, which is denoted by $\dot{\mathbf{x}}(t_0), \ldots, \dot{\mathbf{x}}(t_{\mathcal{N}})$. Using this derivative information, we form the following matrix:

$$\dot{\mathbf{X}} = \left[\dot{\mathbf{x}}(t_0), \ldots, \dot{\mathbf{x}}(t_{\mathcal{N}})\right]. \tag{4}$$

Then, determining the operators boils down to solving a least-squares problem, which can be written as

$$\min_{\mathbf{A},\mathbf{H},\mathbf{B}} \left\| \dot{\mathbf{X}} - \left[\mathbf{A},\ \mathbf{H},\ \mathbf{B}\right] \mathcal{D} \right\|_F, \tag{5}$$

with $\mathcal{D} = \begin{bmatrix} \mathbf{X} \\ \mathbf{x} \tilde{\otimes} \mathbf{x} \\ \mathbf{U} \end{bmatrix}$, where the product $\tilde{\otimes}$ is defined as $\mathbf{G} \tilde{\otimes} \mathbf{G} = \left[\mathbf{g}_1 \otimes \mathbf{g}_1, \ldots, \mathbf{g}_{\mathcal{N}} \otimes \mathbf{g}_{\mathcal{N}}\right]$ with $\mathbf{g}_i$ being the i-th column of the matrix $\mathbf{G} \in \mathbb{R}^{n \times \mathcal{N}}$, and $\otimes$ denotes the Kronecker product. Additionally, the reader should notice that whenever the provided data in (2) is already low-dimensional and further compression is not possible, then the projection onto the POD coordinates in equation (3) is not required.

Although the optimization problem (5) appears straightforward, it poses a couple of major challenges. Firstly, the matrix $\mathcal{D}$ can be ill-conditioned, thus making the optimization problem (5) challenging. A way to circumvent this problem is to make use of suitable regularization schemes, and many proposals are made in this direction in the literature, see, e.g., [12–14].

An important challenge—one of the most crucial ones—is related to the stability of the inferred models. When the optimization problem (5) is solved, then the inferred operators only aim to minimize the specific design objective. However, solving (5) does not guarantee that the resulting dynamical system will be stable; The problem of imposing boundedness for quadratic system without control was studied in [15] using soft constraints. More recently, the authors in [10] proposed a parametrization for the system operators allowing to impose different types of stability properties on the learned uncontrolled dynamical models, such as local stability, global stability and the existence of trapping regions. In this paper, we build upon the concepts established in [10] for uncontrolled quadratic systems and extend those results to quadratic control system of the form (1).

3 Stability for quadratic control systems

In this section, our main goal is to parametrize quadratic control systems that are stable. To this aim, we start by short reviewing the parametrization of stable matrices proposed in [16]). Thereafter, we discuss the results presented in [10] allowing to parametrize energy preserving quadratic nonlinearities. Based on these, we subsequently characterize boundness for quadratic control systems with energy-preserving nonlinearities. These results will then be used to impose stability while learning of quadratic control systems.

3.1 Parametrization of a stable matrix

A linear dynamical system

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t), \tag{6}$$

where $\mathbf{A} \in \mathbb{R}^{n\times n}$ is said to be asymptotically stable when all eigenvalues of the matrix $\mathbf{A}$ lie strictly in the left half complex plane. In this case, the matrix $\mathbf{A}$ is called Hurwitz.

In the light of this, an important characterization of stable matrices is provided in [16, Lemma 1], which states that any Hurwitz matrix $\mathbf{A}$ can be expressed as:

$$\mathbf{A} = (\mathbf{J} - \mathbf{R})\mathbf{Q}, \tag{7}$$

where $\mathbf{J} = -\mathbf{J}^\top$ is a skew-symmetric matrix, $\mathbf{R} = \mathbf{R}^\top \succ 0$, and $\mathbf{Q} = \mathbf{Q}^\top \succ 0$ are symmetric positive definite matrices. Moreover, if $\mathbf{A}$ can be expressed as in (7), then $\mathbf{V}(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top\mathbf{Q}\mathbf{x}$ is a Lyapunov function for the linear dynamical system (6). In particular, the system (6) has $\mathbf{E}(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top\mathbf{x}$ as a (strict) Lyapunov function if and only if the matrix $\mathbf{A}$ can be decomposed as

$$\mathbf{A} = \mathbf{J} - \mathbf{R}, \tag{8}$$

for $\mathbf{J} = -\mathbf{J}^\top$ and $\mathbf{R} = \mathbf{R}^\top \succ 0$. It this case, we say that the system (6) is *monotonically stable*, because the 2-norm of the state, i.e., $\|\mathbf{x}(t)\|_2$, always decreases with time.

It is worth mentioning that the authors in [9] proposed a framework to learn linear stable dynamical system by levering the parametrization (7). Additionally, in [10], this parametrization of $\mathbf{A}$ together with the notion of energy preserving quadratic nonlinearities plays a crucial role in learning (uncontrolled) quadratic models from data. In what follows, we recall the notion of energy preserving quadratic nonlinearities and their parametrization.

3.2 Energy preserving quadratic nonlinearity

Here, we examine quadratic dynamical systems as described in (1) for which the quadratic nonlinearity satisfies some algebraic constraints, as proposed e.g., in [11,17]. We refer the Hessian matrix, or quadratic matrix $\mathbf{H}$ in (1) to as energy-preserving when it satisfies the following criteria:

$$\mathbf{H}_{ijk} + \mathbf{H}_{ikj} + \mathbf{H}_{jik} + \mathbf{H}_{jki} + \mathbf{H}_{kij} + \mathbf{H}_{kji} = 0, \tag{9}$$

for each $i, j, k \in \{1, \dots, n\}$, with $\mathbf{H}_{ijk} := e_i^\top \mathbf{H}(e_j \otimes e_k)$, where $e_i \in \mathbb{R}^n$ is the ith canonical basis vector. The condition (9) can be expressed using Kronecker product notation, see [10], as follows:

$$\mathbf{z}^\top \mathbf{H}(\mathbf{z} \otimes \mathbf{z}) = 0, \quad \forall \mathbf{z} \in \mathbb{R}^n. \tag{10}$$

Energy preserving nonlinearities typically appear in finite element discretization of fluid mechanical models with certain boundary conditions, see, e.g., [18,19], as well as, in magneto-hydrodynamics applications, see [20]. For such uncontrolled quadratic systems (the system (1) with $\mathbf{u} \equiv 0$) with energy-preserving Hessian, it is possible to establish conditions that ensure the system's energy, defined by $\mathbf{E}(\mathbf{x}(t)) := \frac{1}{2}\mathbf{x}^\top(t)\mathbf{x}(t) = \frac{1}{2}\|\mathbf{x}(t)\|_2^2$, decreases in a strictly monotonic fashion for all trajectories, see [10,11]. Additionally, the authors in [10] (Lemma 2) showed that a Hessian matrix $\mathbf{H} \in \mathbb{R}^{n\times n^2}$ satisfying (10) can be parametrized without loss of generality as

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_1 & \dots & \mathbf{H}_n \end{bmatrix}, \tag{11}$$

with $\mathbf{H}_i \in \mathbb{R}^{n\times n}$ being skew-symmetric, i.e., $\mathbf{H}_i = -\mathbf{H}_i^\top$. As a consequence, this parametrization is leveraged in the learning process in [10], leading to the inference of stable quadratic (uncontrolled) models. Furthermore, the authors in [10] generalized these results to the case of more general quadratic Lyapunov functions.

3.3 Bounded-input bounded-state stability result

Based on the results presented so far, we now proceed further to establish our main results of this paper on the stability for quadratic control systems.

Theorem 1 *Consider a quadratic control system as in (1). Assume that the matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ is monotonically stable and can be decomposed as $\mathbf{A} = \mathbf{J} - \mathbf{R}$, where $\mathbf{J} = -\mathbf{J}^\top$ and $\mathbf{R} = \mathbf{R}^\top \succ 0$, and $\mathbf{H} \in \mathbb{R}^{n\times n^2}$ is an energy-preserving Hessian. Then, if the input function $\mathbf{u} \in L_\infty$, then the state vector $\mathbf{x}(t)$ monotonically converges to the interior of the ball $\mathcal{B}_r(0)$, where*

$$r = \frac{\|\mathbf{B}\|_2 \|\mathbf{u}\|_{L_\infty}}{\sigma_{\min}(\mathbf{R})},$$

$\sigma_{\min}(\cdot)$ is the minimum singular value of a matrix and $\|\mathbf{u}\|_{L_\infty} = \texttt{ess sup}_{t\geq 0}\|\mathbf{u}(t)\|_2$. Furthermore, for every input $\mathbf{u} \in L_\infty$, the state vector is bounded

$$\|\mathbf{x}(t)\|_2 \leq \max\{\mathbf{x}_0, r\}; \tag{12}$$

thus, the quadratic control system is bounded-input bounded-state stable.

Proof: Let us consider the energy function $\mathbf{E}(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top\mathbf{x}$. We then show that this function is a Lyapunov function outside of $\mathcal{B}_r(0)$, which proves the result. We utilize the parameterization of the matrix $\mathbf{A} = \mathbf{J} - \mathbf{R}$, where $\mathbf{J} = -\mathbf{J}^\top$ and $\mathbf{R} = \mathbf{R}^\top \succ 0$ and the matrix $\mathbf{H}$ to be energy-preserving . To this aim, the derivative of $\mathbf{E}(\mathbf{x}(t))$ along the trajectory $\mathbf{x}(t)$ is given by

$$\begin{aligned}
\dot{\mathbf{E}}(\mathbf{x}(t)) &= \mathbf{x}(t)^\top \dot{\mathbf{x}}(t) \\
&= \mathbf{x}(t)^\top \left(\mathbf{A}\mathbf{x}(t) + \mathbf{H}\left(\mathbf{x}(t) \otimes \mathbf{x}(t)\right) + \mathbf{B}\mathbf{u}(t)\right) \\
&= \mathbf{x}(t)^\top \left(\left(\mathbf{J} - \mathbf{R}\right)\mathbf{x}(t) + \mathbf{H}\left(\mathbf{x}(t) \otimes \mathbf{x}(t)\right) + \mathbf{B}\mathbf{u}(t)\right) \\
&= -\mathbf{x}(t)^\top \mathbf{R}\mathbf{x}(t) + \cancelto{0}{\mathbf{x}(t)^\top \mathbf{H}\left(\mathbf{x}(t) \otimes \mathbf{x}(t)\right)} \\
&\quad + \mathbf{x}(t)^\top \mathbf{B}\mathbf{u}(t) \\
&\leq -\sigma_{\min}(\mathbf{R})\|\mathbf{x}(t)\|_2^2 + \|\mathbf{B}\|_2\|\mathbf{u}\|_{L_\infty}\|\mathbf{x}(t)\|_2.
\end{aligned}$$

Define $r = \dfrac{\|\mathbf{B}\|_2\|\mathbf{u}\|_{L_\infty}}{\sigma_{\min}(\mathbf{R})}$. Then, $\dot{\mathbf{E}}(\mathbf{x}(t)) < 0$ for $\|\mathbf{x}(t)\|_2 > r$. Hence, $\mathbf{E}(\mathbf{x}(t))$ is a Lyapunov function for $\|\mathbf{x}(t)\| > r$, and hence, it is monotonically decreasing outside of the ball $\mathcal{B}_r(0)$. As a consequence, the state norm $\|\mathbf{x}(t)\|_2$ is also a monotonically decreasing function outside of $\mathcal{B}_r(0)$. This implies that $\|\mathbf{x}_0\|_2 \geq \|\mathbf{x}(t)\|$ when $\|\mathbf{x}_0\|_2 \geq r$, which proves the result in (12). As a consequence, every input $\mathbf{u} \in L_\infty$ will lead to bounded trajectories and the system is bounded-input bounded-state stable. □

Theorem 1 shows that if $\mathbf{A}$ is monotonically stable and $\mathbf{H}$ is energy preserving, the quadratic control system of the form (1) is bounded-input bounded-state stable, i.e., every input $\mathbf{u} \in L_\infty$ will lead to bounded trajectories $\mathbf{x}(t)$ satisfying (12). Additionally, to prove this result we use the state energy as a Lyapunov function.

Theorem 1 together with the parametrization of monotonically stable matrices in (8) and energy-preserving Hessians in (11) will allow us to learn quadratic control systems with a stable behavior.

4 Learning bounded control systems

Based on Theorem 1, we establish an inference framework to learn bounded quadratic control models of the form (1), via the designated $\mathbf{X}$, $\mathbf{U}$ and $\dot{\mathbf{X}}$ dataset. The inference problem is formulated as:

$$\begin{aligned}
&\underset{\widehat{\mathbf{J}},\widehat{\mathbf{R}},\widehat{\mathbf{H}}_1,\ldots,\widehat{\mathbf{H}}_n,\widehat{\mathbf{B}}}{\arg\min} \left\|\dot{\mathbf{X}} - \widehat{\mathbf{A}}\mathbf{X} - \widehat{\mathbf{H}}\mathbf{X}^{\otimes} - \widehat{\mathbf{B}}\mathbf{U}\right\|_F, \\
&\text{where } \widehat{\mathbf{A}} = (\widehat{\mathbf{J}} - \widehat{\mathbf{R}}) \quad \text{and} \quad \widehat{\mathbf{H}} = \left[\widehat{\mathbf{H}}_1, \ldots, \widehat{\mathbf{H}}_n\right], \\
&\text{subject to } \widehat{\mathbf{J}} = -\widehat{\mathbf{J}}^\top, \widehat{\mathbf{R}} = \widehat{\mathbf{R}}^\top \succ 0, \text{ and} \\
&\qquad \widehat{\mathbf{H}}_i = -\widehat{\mathbf{H}}_i^\top, \ i \in \{1, \ldots, n\}.
\end{aligned} \tag{13}$$

Upon determining the optimal set $(\mathbf{J}, \mathbf{R}, \mathbf{H}_1, \ldots, \mathbf{H}_n, \mathbf{B})$ solving (13), the matrices $\mathbf{A}$ and $\mathbf{H}$ are constructed as:

$$\mathbf{A} = (\mathbf{J} - \mathbf{R}), \quad \mathbf{H} = [\mathbf{H}_1, \ldots, \mathbf{H}_n], \tag{14}$$

yielding a quadratic control system in the form of (1), which is guaranteed to be stable in the view of Theorem 1. Notice that the problem formulation in (13) imposes certain constraints on the matrices. To circumvent these restrictions, as used in [9, 10], skew-symmetric matrices $\widehat{\mathbf{J}}$ (or $\widehat{\mathbf{H}}_k$) and symmetric positive (semi)definite matrices $\tilde{\mathbf{R}}$ can be parameterized as

$$\widehat{\mathbf{J}} = \bar{\mathbf{J}} - \bar{\mathbf{J}}^\top, \quad \text{and} \quad \tilde{\mathbf{R}} = \bar{\mathbf{R}}\bar{\mathbf{R}}^\top, \tag{15}$$

where $\bar{\mathbf{J}}, \bar{\mathbf{R}} \in \mathbb{R}^{n\times n}$ are square matrices with any constraints. With this parametrization, (13) becomes an unconstrained optimization problem. However, due to the lack of analytical solution to (13), we solve the problem using a gradient-based approach.

5 Extension to more general quadratic Lyapunov functions

Theorem 1 shows that a quadratic control system with $\mathbf{A}$ monotonically stable and $\mathbf{H}$ energy preserving is guaranteed to be bounded-input bounded-state stable. We have proved the result using the state energy $\mathbf{E}(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top\mathbf{x}$ as the Lyapunov function. In this section, we sketch an extension of this result to the case for which more general quadratic Lyapunov functions are considered, i.e., functions of the form $\mathbf{V}(\mathbf{x}) = \mathbf{x}^\top\mathbf{Q}\mathbf{x}$, where $\mathbf{Q} = \mathbf{Q}^\top \succ 0$ is a symmetric positive definite matrix.

Let us assume that the matrices $\mathbf{A}$ and $\mathbf{H}$ of a quadratic control system of the form (1) can be written as

$$\mathbf{A} = (\mathbf{J} - \mathbf{R})\mathbf{Q} \quad \text{and} \quad \mathbf{H} = \begin{bmatrix}\mathbf{H}_1\mathbf{Q} & \dots & \mathbf{H}_n\mathbf{Q}\end{bmatrix}. \tag{16}$$

In this case, $\mathbf{A}$ is a Hurwitz matrix and $\mathbf{H}$ is a generalized energy-preserving Hessian (see [10]). With similar arguments as those in the proof of Theorem 1, one can show that for bounded inputs $\mathbf{u} \in L_\infty$, the state $\mathbf{x}(t)$ also has a bounded behavior. Indeed, to this aim, one needs to use $\mathbf{V}(\mathbf{x})$ as a Lyapunov function, and the result follows straightforwardly. As a consequence, the parametrization in (16) can be leveraged within the learning process, i.e., the optimization problem (13) can incorporate this more general parametrization, thus yielding inferred quadratic control systems to be stable by construction.

6 Numerical results

In this section, we assess the efficacy of the methodology outlined in (13), referred to herein as `stable-OpInfc`, through a coupled of numerical examples. We compare our approach with operator inference [5], which we denote as (`OpInfc`). All experiments are carried out using `PyTorch`, with 12,000 updates with the Adam optimizer ([21]) and a triangular cyclic learning rate ranging from 10^{-6} to 10^{-2}. Additionally, we regularize the matrix $\mathbf{H}$ in quadratic systems by adding $10^{-4} \cdot \|\mathbf{H}\|_{l_1}$ in the loss function, where $\|\cdot\|_{l_1}$ denotes l_1-norm. The initial values for the matrix coefficients are randomly generated from a Gaussian distribution with a mean of 0 and standard deviation of 0.1.

6.1 Low-dimensional example I

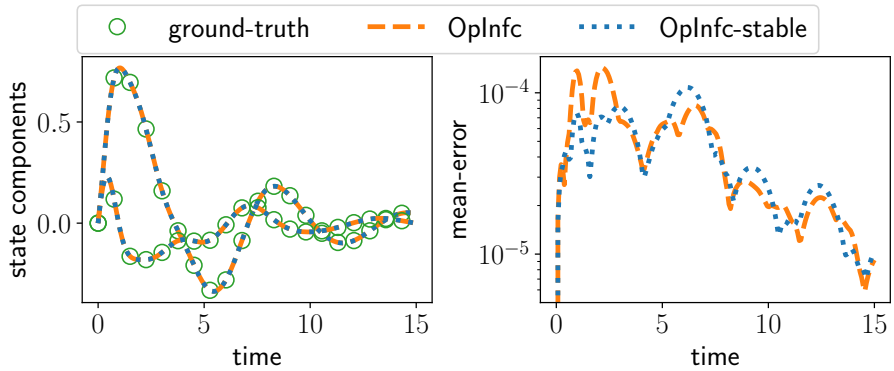
Our first numerical example consists of a low-dimensional quadratic control system of the form (1), where

$$\mathbf{A} = \begin{pmatrix} -1 & 1 \\ -1 & -2 \end{pmatrix}, \quad \mathbf{H} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{pmatrix}, \quad \text{and} \quad \mathbf{B} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}. \tag{17}$$

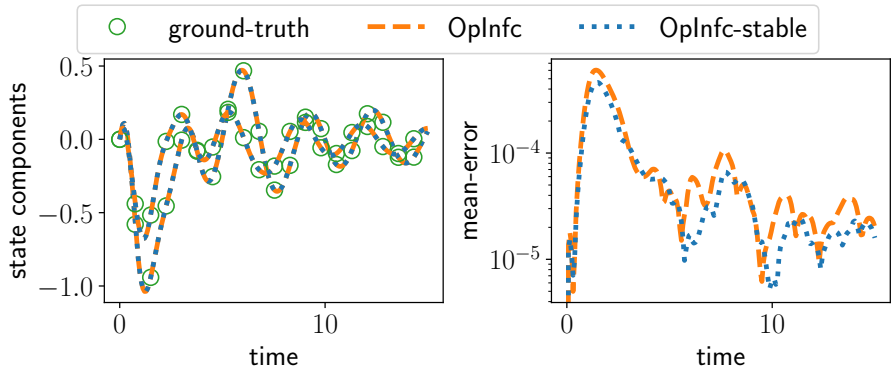
We collect the data with zero initial condition and two different training input functions of the form

$$\mathbf{u}(t) = \sin(f_1 t)e^{-f_2 t} + \sin(g_1 t)e^{-g_2 t}, \tag{18}$$

where $f_i \in \mathbb{Z}$, $i = \{1, 2\}$ are randomly chosen integers between 0 and 5, and $g_i \in \mathbb{R}$, $i = \{1, 2\}$ are randomly chosen real numbers between 0 and 0.5. We collect 200 points for each training input in the time-span of $[0, 10]$. Then, we learn quadratic control models using `stable-OpInfc` and



(a) For the testing input $\mathbf{u}_1$.



(b) For the testing input $\mathbf{u}_2$.

Figure 1: Low-dimensional example I: A performance test for testing control inputs of the inferred models.

`OpInfc`. Since the data is low-dimensional, the proper orthogonal decomposition step in (3) is not performed in this example.

For comparison, we consider two test control inputs of the form

$$\mathbf{u}_1(t) = \sin(t)e^{(-0.2\cdot t)} + \sin(2t)e^{(-0.6\cdot t)} + \cos(3t)e^{(-t)}$$
$$\mathbf{u}_2(t) = -\sin(2t)e^{(-0.1\cdot t)} - \sin(t)e^{(-0.3\cdot t)} + \cos(4t)e^{(-0.5t)}.$$

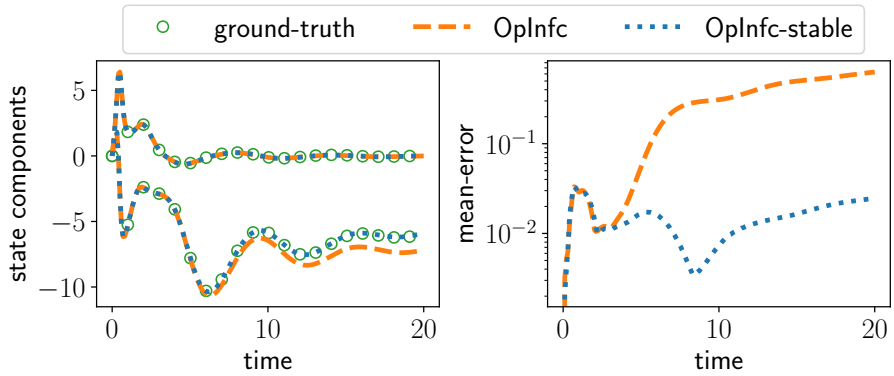
Note that the testing inputs are very different than from the training ones (see (18)). Next, we compare the time-domain simulations of the learned models with the ground truth and the results are depicted in Figure 1. We notice a faithful learning of the underlying models using both approaches; however, the proposed methods `stable-OpInfc` ensures stability by construction for any other selected input.

6.2 Low-dimensional example II

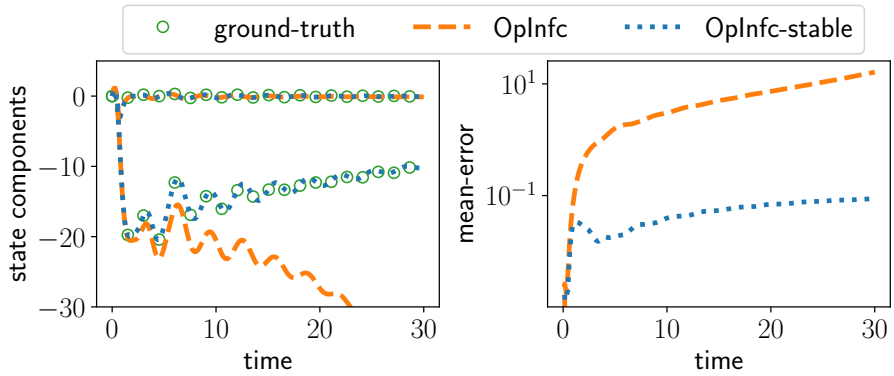
Our second numerical example, we consider a slightly different quadratic control system than the previous example whose the matrix $\mathbf{A}$ is as follows:

$$\mathbf{A} = 0.01 \begin{pmatrix} -1 & 1 \\ -1 & -2 \end{pmatrix},$$

and the matrices $\mathbf{H}$ and $\mathbf{B}$ are the same as (17). We collect training data using zero initial conditions and two controlled inputs with the same setting as the previous example. Moreover, for this example, we add Gaussian noise of zero mean and 0.02 standard derivation in the training



(a) For the testing input $\mathbf{w}_1$.



(b) For the testing input $\mathbf{w}_2$.

Figure 2: Low-dimensional example II: A performance test for testing control inputs of the inferred models.

data. Then, we learn quadratic controlled inputs using `OpInfc` and `stable-OpInfc`. We compare the qualities of these two models using testing control inputs, similar to the previous example. Particularly, to test the stability of both models, we use high-magnitude test control inputs as follows:

$$\mathbf{w}_1(t) = 10 \cdot \Big(\sin(t)e^{(-0.2\cdot t)} + \sin(2t)e^{(-0.6\cdot t)} + \cos(3t)e^{(-t)} \Big)$$
$$\mathbf{w}_2(t) = 10 \cdot \Big(-\sin(2t)e^{(-0.1\cdot t)} - \sin(t)e^{(-0.3\cdot t)} + \cos(4t)e^{(-0.5t)} \Big).$$

The time-domain simulations using the learned models for these two test control inputs are shown in Figure 2. We notice that `OpInfc` yields unstable behaviors, particularly for the control $\mathbf{w}_2$, whereas `stable-OpInfc` results into the models which are stable by construction and this phenomena is also numerically observed.

6.3 High-dimensional Burgers' example

In our next example, we consider viscus Burgers' example, whose governing equation is as follows:

$$\begin{aligned} &\frac{\partial v}{\partial t} + v\frac{\partial v}{\partial \xi} = \mu \frac{\partial^2}{\partial \xi^2} + f(\xi, t), \\ &v(0,t) = 0, \quad v(L,t) = 0, \\ &v(\xi, 0) = 0, \end{aligned} \tag{21}$$

where $\xi \in [0, L]$ and t denote space and time, respectively, and $v(\xi, t)$ denotes the state variable at the spatial location ξ and at time t. We set $\mu = 0.05$ and $L = 2$. Moreover, $f(\xi, t)$ denotes a source term, and in this example, we assume that the source term $f(\xi, t)$ is separable, i.e., $b(\xi)u(t)$. Additionally, we consider

$$b(\xi) = \cos\left(\left(\frac{\xi}{L} - 1\right)\frac{\pi}{2}\right).$$

Note that the consider Burgers' example has Dirichlet boundary conditions at both boundary ends. Hence, the quadratic term is energy-preserving.

We discretize the governing equation using a finite difference scheme by considering 251 points in the space. For generating training data, we consider the control input $\mathbf{u}(t)$ of the form

$$\mathbf{u}(t) = \sin(f_1 t)e^{-g_1 t} + \sin(f_2 t)e^{-g_2 t}, \tag{22}$$

where f_1 and f_2 are randomly drawn from a Gaussian distribution of $\mathcal{N}(0, 2)$ and g_1 and g_2 are randomly drawn from a uniform distribution of $\mathcal{U}(0.1, 1.1)$. We consider 20 different training inputs and for each train input, we take 1001 points at equidistant in the time $[0, 10]$.

Towards learning quadratic control models, we first aim at determining a suitable low-dimensional representation of the high dimensional data. It is done by means of singular value deposition of the training data. We project the high-dimensional data onto a lower dimensional subspace using the most dominant left singular vectors. We take nine most dominant ones which captures more than 99.90% energy present in the training data. Furthermore, to learn quadratic models, we require derivative information, which is estimate using 5-order stencils.

Next, we learn quadratic models using `OpInfc` and `stable-OpInfc` using the projected low-dimensional data. To capture the qualities of both these learned models, we consider testing control inputs, which takes the following form:

$$\mathbf{u}(t) = \sin(f_1 t)e^{-g_1 t} + \sin(f_2 t)e^{-g_2 t} + \cos(f_3 t)e^{-g_3 t}, \tag{23}$$

where $f_i \in \mathcal{N}(0, 2)$ and $g_i \in \mathcal{U}(0.1, 1.1)$, $i \in \{1, 2, 3\}$. We run testing for 10 different testing control inputs which are quite different than the training ones. Among 10 test cases, for one case, we present time-domain solutions obtained using the learned models and compare with the ground truth in Figure 3. For a comparison for all 10 test cases, we compute the following measure:

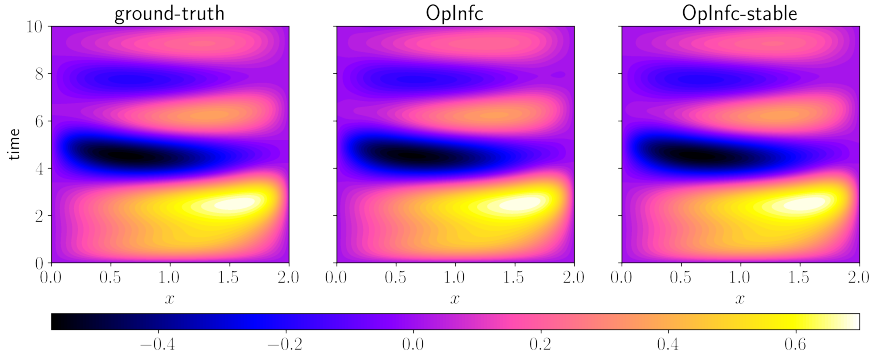
$$\texttt{err} = \texttt{mean}\left(\mathbf{X}^{\texttt{ground-truth}} - \mathbf{X}^{\texttt{learned}}\right), \tag{24}$$

where $\mathbf{X}^{\texttt{ground-truth}}$ and $\mathbf{X}^{\texttt{learned}}$ contain the solutions vector at all time t for the ground truth and learned quadratic models, respectively. Based on the measure (24), we compute the errors based on the solutions obtained using `OpInfc` and `stable-OpInfc` and plot in Figure 4. We notice that a slightly better performance for `stable-OpInfc` despite enforcing stability parameterization.

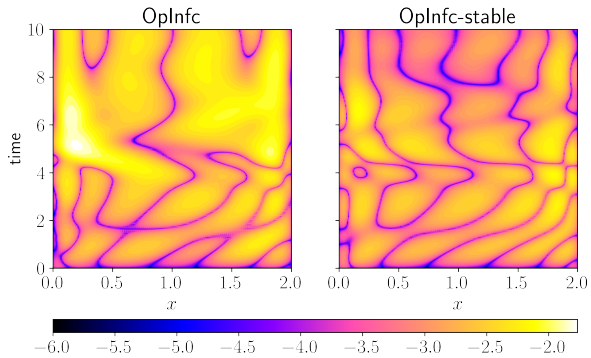
7 Conclusions

In this paper, we introduced a data-driven methodology designed to ensure a bounded stability of the learned quadratic control systems. Firstly, under the assumption that the linear operator is stable and the quadratic operator is energy-preserving, we have showed that quadratic control systems is bounded-input bounded-state stable. Leveraging our previous work [9], we have parameterized the matrices of a quadratic system, satisfying stability and energy-preserving hypotheses by construction. And we have utilized the matrix parameterizations in a data-driven setting to obtain stable quadratic control systems. We have discussed the effectiveness of our proposed methodology using two numerical examples and have compared the results when stability is not enforced. The results highlight the robust performance ad stability-certificates of the proposed approach, affirming its potential to significantly advance the field of data-driven learning of dynamical systems.

In our methodology, we require accurate derivative information, which can be difficult to estimate if data are noisy and sparse. To avoid this requirement, we can incorporate integrating scheme or the concept of neural ODEs [22]. With this spirit, methodologies to learn uncontrolled dynamical systems are discussed, e.g., in [23,24] which will be adaopted to controlled cases in our future work.



(a) Time-domain response.



(b) Absolute error in a log-scale.

Figure 3: Burgers' example: A performance test for a testing control input of the inferred models.

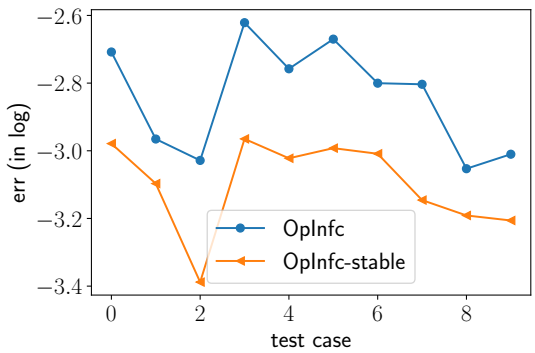


Figure 4: Burgers' example: A comparison of `OpInfc` and `stable-OpInfc` for 10 test cases.

References

[1] I. Mezić, Analysis of fluid flows via spectral properties of the Koopman operator, Annu. Rev. Fluid Mech. 45 (2013) 357–378.

[2] P. J. Schmid, Dynamic mode decomposition of numerical and experimental data, J. Fluid Mech. 656 (2010) 5–28. doi:10.1017/S0022112010001217.

[3] J. L. Proctor, S. L. Brunton, J. N. Kutz, Dynamic mode decomposition with control, SIAM J. Appl. Dyn. Syst. 15 (1) (2016) 142–161.

[4] S. L. Brunton, J. L. Proctor, J. N. Kutz, Discovering governing equations from data by sparse

identification of nonlinear dynamical systems, Proc. Nat. Acad. Sci. U.S.A. 113 (15) (2016) 3932–3937.

[5] B. Peherstorfer, K. Willcox, Data-driven operator inference for nonintrusive projection-based model reduction, Comp. Meth. Appl. Mech. Eng. 306 (2016) 196–215.

[6] C. Gu, QLMOR: A projection-based nonlinear model order reduction approach using quadratic-linear representation of nonlinear systems, IEEE Trans. Comput. Aided Des. Integr. Circuits. Syst. 30 (9) (2011) 1307–1320.

[7] E. Qian, B. Kramer, B. Peherstorfer, K. Willcox, Lift & learn: Physics-informed machine learning for large-scale nonlinear dynamical systems., Physica D: Nonlinear Phenomena 406 (2020) 132401. doi:10.1016/j.physd.2020.132401.

[8] P. Goyal, P. Benner, Generalized quadratic embeddings for nonlinear dynamics using deep learning, arXiv preprint arXiv:2211.00357 (2022).
URL https://arxiv.org/abs/2211.00357v2

[9] P. Goyal, I. Pontes Duff, P. Benner, Stability-guaranteed learning of linear models, arXiv preprint arXiv:2301.10060 (2023).

[10] P. Goyal, I. Pontes Duff, P. Benner, Guaranteed stable quadratic models and their applications in SINDy and operator inference, arXiv preprint arXiv:2308.13819 (2023).

[11] M. Schlegel, B. R. Noack, On long-term boundedness of Galerkin models, J. Fluid Mech. 765 (2015) 325–352.

[12] S. Yıldız, P. Goyal, P. Benner, B. Karasözen, Learning reduced-order dynamics for parametrized shallow water equations from data, Internat. J. Numer. Methods in Fluids 93 (8) (2021) 2803–2821.

[13] S. A. McQuarrie, C. Huang, K. E. Willcox, Data-driven reduced-order models via regularised operator inference for a single-injector combustion process, J. Royal Society of New Zealand 51 (2) (2021) 194–211.

[14] P. Benner, P. Goyal, J. Heiland, I. Pontes Duff, Operator inference and physics-informed learning of low-dimensional models for incompressible flows, Electron. Trans. Numer. Anal. 56 (2022) 28–51.

[15] A. A. Kaptanoglu, J. L. Callaham, A. Aravkin, C. J. Hansen, S. L. Brunton, Promoting global stability in data-driven models of quadratic nonlinear dynamics, Physical Review Fluids 6 (9) (2021) 094401.

[16] N. Gillis, P. Sharma, On computing the distance to stability for matrices using linear dissipative Hamiltonian systems, Automatica 85 (2017) 113–121.

[17] E. N. Lorenz, Deterministic nonperiodic flow, J. Atmospheric Sciences 20 (2) (1963) 130–141.

[18] P. Holmes, J. L. Lumley, G. Berkooz, C. W. Rowley, Turbulence, Coherent Structures, Dynamical Systems and Symmetry, Cambridge University Press, 2012.

[19] H. Schlichting, K. Gersten, Boundary-layer Theory, Springer, 2016.

[20] A. A. Kaptanoglu, K. D. Morgan, C. J. Hansen, S. L. Brunton, The structure of global conservation laws in Galerkin plasma models, arXiv preprint arXiv:2101.03436 (2021).

[21] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).

[22] R. T. Chen, Y. Rubanova, J. Bettencourt, D. K. Duvenaud, Neural ordinary differential equations, Adv. Neural Inform. Processing Sys. 31 (2018).

[23] P. Goyal, P. Benner, Discovery of nonlinear dynamical systems using a Runge-Kutta inspired dictionary-based sparse regression approach, Proc. Royal Society A: Mathematical, Physical and Engineering Sciences 478 (2262) (2022) 20210883.

[24] W. I. T. Uy, D. Hartmann, B. Peherstorfer, Operator inference with roll outs for learning reduced models from scarce and low-quality data, Comp. & Math. Appl. 145 (2023) 224–239.