

Learning Koopman Bilinear Models with Multiplication-closed Observations for Linear Optimal Controller Design

Ketong Zheng¹, Peng Huang², Andrés Villamil^{1,3}, Jonathan Casas^{1,3} and Gerhard P. Fettweis^{1,2,3}

Abstract—The Koopman operator approximation is emerging as a leading approach for identifying and controlling nonlinear systems by transforming them into a bilinear form. However, designing reactive controllers for the Koopman bilinear system remains challenging. This paper proposes a purely data-driven method to learn the Koopman bilinear representation of control-affine systems using measurement data only and design a linear optimal controller for the learned system. Specifically, Deep Neural Networks (DNNs) are employed to learn a finite set of observables that approximately span a Koopman-invariant subspace and form a multiplication-closed set. This multiplication-closed property facilitates optimal controller design by enabling the conversion of the Koopman bilinear system into a closed-loop linear system. The linear control matrix is derived by iteratively solving the Koopman Riccati equation while minimizing an upper bound of the optimal cost. The proposed approach is validated on the Van der Pol oscillator, which outperforms the method that approximates the Koopman control system using a fixed function library in prediction accuracy and control performance.

I. INTRODUCTION

The accurate modeling of dynamical systems is crucial for the safe and efficient operation of agents, whether they function locally or remotely through networked environments [1]. However, in many real-world scenarios, the underlying system dynamics remain either unknown or only partially available due to the complex nonlinear characteristics inherent in robotic systems and their interactions with the environment. Fortunately, the abundance of data measurements in the digital world offers an alternative route for data-driven modeling and controller design, which enables tasks ranging from predicting fluid flows [2] to controlling robotic systems with high degrees of freedom (DoF) [3], [4]. A key approach in this context is using the Koopman operator, which allows for the identification of dynamical systems and facilitates the design of efficient model-based controllers.

*The authors acknowledge the financial support by the Federal Ministry of Education and Research of Germany in the programme of "Souverän. Digital. Vernetzt.". Joint project 6G-life, project identification number: 16KISK001K; and the German Research Foundation (DFG, Deutsche Forschungsgemeinschaft) as part of Germany's Excellence Strategy – EXC 2050/I – Project ID 390696704 – Cluster of Excellence "Centre for Tactile Internet with Human-in-the-Loop" (CeTI) of Technische Universität Dresden. This work is partially financed on the basis of the budget passed by the Saxon State Parliament.

¹Vodafone Chair Mobile Communications Systems, Technische Universität Dresden, 01069 Dresden, Germany. ²Barkhausen Institut, 01062 Dresden, Germany. ³Centre for Tactile Internet with Human-in-the-Loop (CeTI).

ketong.zheng@ifn.et.tu-dresden.de,
{peng.huang, andres.villamil, jonathan.casas,
gerhard.fettweis}@tu-dresden.de

The Koopman operator is a linear operator that acts on infinite-dimensional state observations within a Hilbert space, enabling the linear evolution of a nonlinear dynamical system [5]. However, the infinite degrees of freedom in the Koopman-lifted space pose a significant challenge when applying the Koopman operator to practical prediction and control tasks. To address this challenge, Dynamic Mode Decomposition (DMD) is first connected with the Koopman theory to find the best linear approximation of the Koopman operator using direct state measurements [6], [7], and later developed into extended DMD (eDMD), which uses a finite set of nonlinear state observations to better approximate the Koopman operator [8]. These nonlinear observations should approximate a Koopman-invariant subspace, which can be manually selected from a predefined kernel [8], [9], or learned via Deep Neural Networks (DNNs) [10]–[13], the latter often exhibiting reduced approximation error due to its flexibility to learn any observation set according to the universal approximation theorem.

While the data-driven Koopman approximation was primarily applied to encode unactuated systems, researchers have demonstrated the effectiveness of representing the actuated system into a lifted linear system, allowing for the design of controllers using Linear Model Predictive Control (LMPC) and Linear Quadratic Regulator (LQR) [12], [14]–[17]. For general control-affine systems, however, the Koopman representation takes the form of a bilinear system through the Koopman Canonical Transformation (KCT) [18], [19], assuming that the control-affine vector field lies inside the linear span of Koopman observations. Nevertheless, designing controllers for bilinear systems remains a challenging task, with common approaches like Nonlinear MPC (NMPC) [20]–[22] or Linear Parameter Varying MPC (LPV-MPC) [23] often incurring higher computational costs compared to reactive controllers. In [24], [25], Straesser *et al.* designed a robust data-driven reactive controller directly from the bilinear system with guaranteed stability. In [26], Ma *et al.* proposed a linear optimal control approach for Koopman bilinear systems, assuming that the product between the observations and their Lie derivative along the control vector field lies inside the linear span of Koopman observations. This assumption leads to a closed-loop linear system under linear feedback control, which simplifies the controller design. However, it relies on precise knowledge of the control vector field and is quite a strong assumption, which can introduce significant truncation errors when applied to manually selected Koopman observations.

Therefore, inspired by the work from [26], this paper

introduces a purely data-driven approach to learn a Koopman bilinear representation of control-affine systems with multiplication-closed Koopman observations, such that the product of any two observables remains within the linear span of the observation set. We show that by imposing this multiplication-closed property, the Koopman bilinear system can be transformed into a closed-loop linear form under a linear controller, without requiring any prior knowledge of the original system, thereby facilitating optimal controller design. The key contributions of this paper are threefold: (1) We demonstrate that the learned Koopman bilinear system is closed-loop linear under a linear controller with multiplication-closed observations. (2) We employ DNNs to learn a set of observations that constructs a precise Koopman bilinear representation while maintaining the multiplication-closed property more effectively. (3) An iterative algorithm is developed to compute the linear optimal controller for the Koopman bilinear system by minimizing the upper bound of the optimal cost in the Koopman-lifted space. The proposed approach is validated on the Van der Pol oscillator and compared with the method that constructs Koopman observables from a fixed library. The results demonstrate the superior performance of our DNN-based approach in both prediction accuracy and control effectiveness.

The remainder of this paper is organized as follows: Section II recaps the Koopman operator theory and the Koopman bilinear representation for control-affine systems. Section III discusses the Koopman linear optimal controller design with multiplication-closed observables. The data-driven Koopman operator approximation and the construction of the DNN learning framework are presented in Section IV. Section V presents the training results and evaluates the control performance. Finally, Section VI concludes this paper.

II. PRELIMINARIES

Notations: In this paper, matrices are denoted by capital letters, while bold lowercase letters represent column vectors and non-bold lowercase letters are used to denote scalars. The m -dimensional Euclidean and complex spaces are indicated by \mathbb{R}^m and \mathbb{C}^m , respectively. We use $\mathcal{C}^m \subset \mathcal{H}$ to express the m^{th} order differentiable function space where \mathcal{H} refers to the Hilbert space and \mathcal{L}^m indicates the m^{th} order integrable Lebesgue measurement space. The set of integers within $[m, n]$ where $m \leq n$ is denoted as $[m, n]_{\mathbb{Z}}$. The Lie derivative on vector field \mathbf{f} is given as $L_{\mathbf{f}}$. The $\{\cdot\}$ stands for the set that contains the elements specified within the braces, and $\text{span}\{\cdot\}$ represents linear span of the set of elements. The matrix Frobenius norm is given as $\|\cdot\|_F$, and $(\cdot)^\dagger$ implies the Moore-Penrose inverse. Finally, the Kronecker product is marked as \otimes .

A. Koopman Control System Overview

Consider a nonlinear dynamical system with control-affine terms that evolves on the manifold $\mathcal{X} \subset \mathbb{R}^n$

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \sum_{i=1}^m \mathbf{b}_i(\mathbf{x})u_i, \quad (1)$$

where $\mathbf{f}, \mathbf{b}_i \in \mathcal{C}^2 : \mathcal{X} \mapsto \mathbb{R}^n$ for $i \in [1, m]_{\mathbb{Z}}$, and $\mathbf{u} \in \mathcal{U} := [u_1, \dots, u_m]^T$, with \mathcal{X} and \mathcal{U} indicating the state space and input space, respectively. To introduce the Koopman operator, we first consider the drift term $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ only, which corresponds to the case where $\mathbf{u} = \mathbf{0}$. The flow map of the drift term is given by $S_{\mathbf{f}}^t : [0, \infty) \times \mathcal{X} \mapsto \mathcal{X}$, such that $S_{\mathbf{f}}^t(\mathbf{x}_0) := \mathbf{x}_0 + \int_{t_0}^{t_0+t} \mathbf{f}(\mathbf{x}(\tau))d\tau$, with $\mathbf{x}(t_0) = \mathbf{x}_0$. According to the Koopman operator theory [5], there exists an infinite-dimensional linear operator $\mathcal{M}_{\mathbf{f}}^t : [0, \infty) \times \mathcal{L}^2(\mathcal{X}) \mapsto \mathcal{L}^2(\mathcal{X})$ acts on observations $\forall \varphi \in \mathcal{L}^2(\mathcal{X}) \subset \mathcal{H} : \mathcal{X} \mapsto \mathbb{C}$, such that

$$[\mathcal{M}_{\mathbf{f}}^t \varphi](\mathbf{x}) = \varphi(S_{\mathbf{f}}^t(\mathbf{x})). \quad (2)$$

The infinitesimal generator of $\mathcal{M}_{\mathbf{f}}^t$ acts on φ is given by $[\mathcal{K}_{\mathbf{f}} \varphi](\mathbf{x}) := \lim_{t \rightarrow 0^+} ([\mathcal{M}_{\mathbf{f}}^t \varphi](\mathbf{x}) - \varphi(\mathbf{x}))/t = L_{\mathbf{f}} \varphi(\mathbf{x})$. Meanwhile, if $[\mathcal{K}_{\mathbf{f}} \varphi](\mathbf{x}) = L_{\mathbf{f}} \varphi(\mathbf{x}) = \lambda \cdot \varphi(\mathbf{x})$, then φ is called the Koopman eigenfunction with eigenvalue $\lambda \in \mathbb{C}$. Given that any powered multiplicative combination of Koopman eigenfunctions (e.g. $\varphi_i^m \cdot \varphi_j^n$) is itself a Koopman eigenfunction, it follows that the Koopman eigenfunctions span an infinite-dimensional space.

Definition 1: An observation space $\text{span}\{g_j\}_{j=1}^p \subset \mathcal{L}^2(\mathcal{X})$ with $g_j : \mathcal{X} \mapsto \mathbb{C}$ is called a Koopman-invariant subspace if

$$[\mathcal{K}_{\mathbf{f}} \mathbf{g}](\mathbf{x}) := \mathcal{K}_{\mathbf{f}} \begin{bmatrix} g_1(\mathbf{x}) \\ \vdots \\ g_p(\mathbf{x}) \end{bmatrix} = L_{\mathbf{f}} \mathbf{g}(\mathbf{x}) = A_c \cdot \mathbf{g}(\mathbf{x}), \quad (3)$$

where A_c denotes a $p \times p$ matrix [15].

It is clear that any subset of Koopman eigenfunctions forms a Koopman-invariant subspace. For an arbitrary observation $g \in \{g_j\}_{j=1}^p$, it can be represented by its decomposition into the Koopman eigenspace such that

$$\begin{aligned} L_{\mathbf{f}} g(\mathbf{x}) &= [\mathcal{K}_{\mathbf{f}} g](\mathbf{x}) = \mathcal{K}_{\mathbf{f}} \sum_{i=1}^{\infty} \langle g(\mathbf{x}), \overline{\varphi}_i(\mathbf{x}) \rangle \varphi_i(\mathbf{x}) \\ &= \sum_{i=1}^{\infty} \lambda_i \varphi_i(\mathbf{x}) \int_{\mathcal{X}} g(\mathbf{x}) \cdot \overline{\varphi}_i(\mathbf{x}) d\mathbf{x}. \end{aligned} \quad (4)$$

Since Koopman eigenfunctions $\{\varphi_j\}_{j=1}^{\infty}$ are infinite-dimensional, for an observation set $\{g_j\}_{j=1}^p$ with $p \gg 0$, there exists a finite set of Koopman eigenfunctions $\{\varphi_j\}_{j=1}^p$ such that each observation can be approximated by its decomposition into this finite eigenspace using (4). In this case, $\text{span}\{g_j\}_{j=1}^p$ can be approximated by a Koopman-invariant subspace, such that (3) holds. Now, let us further consider system (1) with control-affine term, where the Koopman-lifted dynamics can be expressed as

$$\dot{\mathbf{g}}(\mathbf{x}) = A_c \cdot \mathbf{g}(\mathbf{x}) + \sum_{i=1}^m L_{\mathbf{b}_i} \mathbf{g}(\mathbf{x}) \cdot u_i. \quad (5)$$

Assumption 1: For a Koopman-invariant subspace spanned by $\{g_j\}_{j=1}^p$ with Koopman-lifted dynamics described in (5), the Lie derivative of the observations

along the control-affine vector field and the state vector $\{L_{b_i} \mathbf{g}\}_{i=1}^m \cup \{\mathbf{x}\}$ lies inside $\text{span}\{\mathbf{g}_j\}_{j=1}^p$. It is reasonable to conclude Assumption 1 holds if the Koopman observations are sufficiently rich [18], [19]. By defining the Koopman-lifted space as $\mathbf{z} := \mathbf{g}(\mathbf{x}) \in \mathcal{Z} \subset \mathbb{R}^p$ and separate the state-independent control part, the lifted dynamics takes the form of a bilinear control system, described as follows:

$$\begin{aligned} \dot{\mathbf{z}} &= A_c \cdot \mathbf{z} + B_{c_0} \cdot \mathbf{u} + \sum_{i=1}^m B_{c_i} \cdot \mathbf{z} \cdot u_i \\ \mathbf{x} &= C \cdot \mathbf{z}. \end{aligned} \quad (6)$$

This bilinear transform is also referred to as KCT, as detailed in [18]. For convenience, we choose the first n Koopman observations as the original state \mathbf{x} , so that the C matrix in (6) can be written as $C = [I_n | 0_{n \times (p-n)}]$.

III. KOOPMAN OPTIMAL CONTROL

This section discusses the design of a linear optimal controller for the Koopman bilinear representation. From this point forward, we consider the controller design in discrete time, with the cost function defined as:

$$\underset{\mathbf{u}_k}{\text{minimize}} J := \frac{1}{2} \sum_{k=0}^{\infty} \mathbf{z}_k^T \cdot Q \cdot \mathbf{z}_k + \mathbf{u}_k^T \cdot R \cdot \mathbf{u}_k \quad (7a)$$

$$\text{s.t. } \mathbf{z}_{k+1} = A \cdot \mathbf{z}_k + B_0 \cdot \mathbf{u}_k + \sum_{i=1}^m B_i \cdot \mathbf{z}_k \cdot u_{k,i} \quad (7b)$$

$$\mathbf{u}_k = -\Gamma \cdot \mathbf{z}_k \quad (7c)$$

$$\mathbf{z}_0 = \mathbf{g}(\mathbf{x}_0), \quad (7d)$$

where $Q \in \mathbb{R}^{p \times p}$ is defined as $Q = \text{diag}(\tilde{Q}_n, 0_{p-n})$ with $\tilde{Q}_n \in \mathbb{R}^{n \times n}$ such that $\tilde{Q}_n = \tilde{Q}_n^T \succeq 0$ denotes the state weight matrix and $0_{p-n} \in \mathbb{R}^{(p-n) \times (p-n)}$ represents the zero matrix. The $\text{diag}(\cdot)$ stands for the block diagonal, and $R = R^T \succ 0$ is the $m \times m$ input weight matrix. The discrete analog of (6) is given as (7b), and $\Gamma \in \mathbb{R}^{m \times p}$ in (7c) represents the linear control matrix.

Proposition 1: If $\text{span}\{\mathbf{g}_j\}_{j=1}^p$ forms a Koopman-invariant subspace that is multiplication-closed, such that $\mathbf{g}(\mathbf{x}) \otimes \mathbf{g}(\mathbf{x}) = [\tilde{B}_1^T, \dots, \tilde{B}_p^T]^T \cdot \mathbf{g}(\mathbf{x})$, then the optimal solution of (7a) satisfies

$$A_\Gamma^T P A_\Gamma + \Gamma^T R \Gamma + Q - P = 0, \quad (8)$$

where $P = P^T \succ 0$ is the cost-to-go matrix, and A_Γ is a Schur stable matrix that linearly dependent on controller Γ .

Proof: With the linear controller $\Gamma = [\gamma_1^T, \dots, \gamma_m^T]^T$, the bilinear control system (7b) can be reformulated as

$$\begin{aligned} \mathbf{z}_{k+1} &= A \cdot \mathbf{z}_k - B_0 \Gamma \cdot \mathbf{z}_k - \sum_{i=1}^m B_i \cdot \mathbf{z}_k \cdot \gamma_i \cdot \mathbf{z}_k \\ &= A \cdot \mathbf{z}_k - B_0 \Gamma \cdot \mathbf{z}_k - \sum_{i=1}^m \sum_{j=1}^p B_i \cdot \mathbf{z}_k \cdot z_{k,j} \cdot \gamma_{ij}. \end{aligned} \quad (9)$$

According to the multiplication-closed property, where $\mathbf{z}_k \otimes \mathbf{z}_k = [\tilde{B}_1^T, \dots, \tilde{B}_p^T]^T \cdot \mathbf{z}_k$, we can further express $\mathbf{z}_k \cdot \mathbf{z}_{k,j}$ as

$\tilde{B}_j \cdot \mathbf{z}_k$, so that (9) can be written as

$$\mathbf{z}_{k+1} = \underbrace{(A - B_0 \Gamma - \sum_{i=1}^m \sum_{j=1}^p \gamma_{ij} \cdot \tilde{B}_{ij})}_{A_\Gamma} \cdot \mathbf{z}_k, \quad (10)$$

where $\tilde{B}_{ij} := B_i \tilde{B}_j$. Therefore, the problem (7a) can be reformulated as

$$\underset{\Gamma}{\text{minimize}} J := \frac{1}{2} \sum_{k=0}^{\infty} \mathbf{z}_k^T \cdot [Q + \Gamma^T R \Gamma] \cdot \mathbf{z}_k \quad (11a)$$

$$\text{s.t. } \mathbf{z}_{k+1} = A_\Gamma \cdot \mathbf{z}_k \quad (11b)$$

$$\mathbf{z}_0 = \mathbf{g}(\mathbf{x}_0). \quad (11c)$$

To find the optimality condition, we further construct the Hamiltonian of problem (11a) as

$$H_k := \frac{1}{2} \mathbf{z}_k^T \cdot [Q + \Gamma^T R \Gamma] \cdot \mathbf{z}_k + \mathbf{q}_{k+1}^T \cdot A_\Gamma \cdot \mathbf{z}_k, \quad (12)$$

where $\mathbf{q}_k \in \mathbb{R}^p$ represents the costate, which can be interpreted as the Lagrange multiplier associated with the constraint (11b). Given the linear closed-loop dynamics, the costate can be proved to have a linear relation *w.r.t* the Koopman-lifted state, which can be written as $\mathbf{q}_k = P \cdot \mathbf{z}_k$ with $P = P^T \succ 0$. Since the optimal controller needs to fulfill the costate equation

$$\begin{aligned} \mathbf{q}_k = P \cdot \mathbf{z}_k &= \frac{\partial H_k}{\partial \mathbf{z}_k} = [Q + \Gamma^T R \Gamma] \cdot \mathbf{z}_k + A_\Gamma^T \cdot \mathbf{q}_{k+1} \\ &= [Q + \Gamma^T R \Gamma] \cdot \mathbf{z}_k + A_\Gamma^T P A_\Gamma \cdot \mathbf{z}_k, \end{aligned} \quad (13)$$

(8) should be guaranteed. \blacksquare

According to the Hamilton-Jacobi-Bellman (HJB) equation, the P matrix derived from (8) is also the optimal cost-to-go matrix, where the induced cost for problem (11a) is given by $J = \mathbf{z}_0^T \cdot P \cdot \mathbf{z}_0$. Let \mathcal{B}_z be defined as the smallest Euclidean ball that contains the Koopman lifted space \mathcal{Z} , such that $\mathcal{B}_z := \{\mathbf{z} \in \mathbb{R}^p \mid \|\mathbf{z}\|_2 \leq r\}$, where $r := \max_{\mathbf{z} \in \mathcal{Z}} \|\mathbf{z}\|_2$, the cost is then upper bounded by $r^2 \cdot \lambda_{\max}(P)$ with $\lambda_{\max}(P)$ denoting the maximum eigenvalue of P . Therefore, problem (11a) is transferred into

$$\begin{aligned} \underset{\Gamma, P}{\text{minimize}} \lambda_{\max}(P) \\ \text{s.t. } P = P^T \succ 0 \text{ and (8),} \end{aligned} \quad (14)$$

where the objective is to minimize the upper bound of the optimal cost. Unlike the optimal control of a linear system, where P is derived as the unique solution of the algebraic Riccati equation with an analytical solution for Γ , problem (14) remains non-convex due to the bilinear product of Γ and P . To address this issue, a similar iterative optimization method, as introduced in [26], is adopted here. When Γ is fixed, the problem (14) reduces to a Semidefinite Programming (SDP) problem which is convex. When P is fixed, however, the constraint is still non-convex *w.r.t* Γ . To tackle this, constraint (8) is further relaxed into

$$A_\Gamma^T P A_\Gamma + \Gamma^T R \Gamma + Q - P \preceq \epsilon \cdot I, \quad \epsilon \geq 0, \quad (15)$$

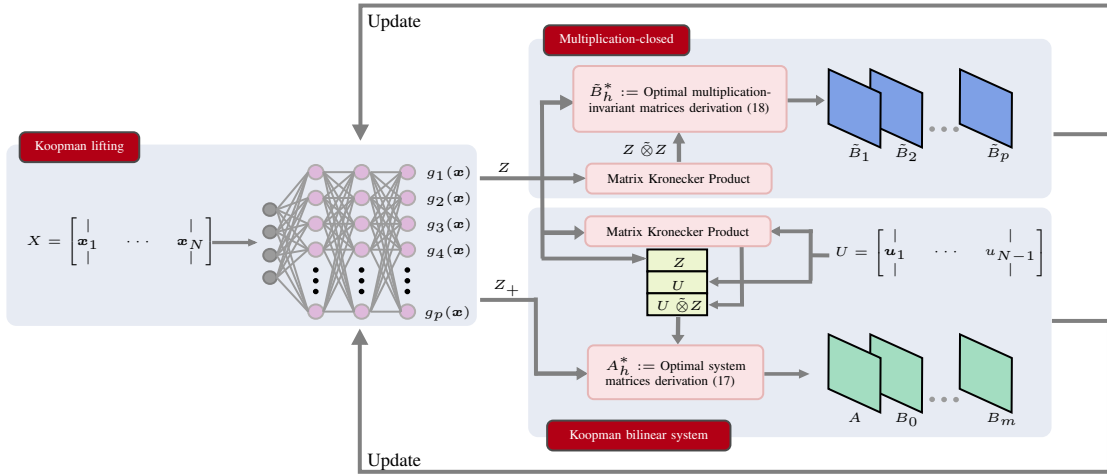


Fig. 1: DNN architecture for joint learning of Koopman bilinear systems and the multiplication-closed observables. Each DNN forward path denotes a Koopman observations. The matrices $A_h^*(\theta)$ and $\tilde{B}_h^*(\theta)$ are calculated using the DNN-lifted states and input measurements, which are then used to formulate the loss function (21).

which can be reformulated as a Linear Matrix Inequality (LMI) using Schur complement, where (14) becomes

$$\begin{aligned} & \text{minimize } \epsilon \\ & \Gamma, \epsilon \geq 0 \\ & \text{s.t. } \begin{bmatrix} -P^{-1} & \mathbf{0} \\ \mathbf{0} & -R^{-1} \\ A_\Gamma^T & \Gamma^T \end{bmatrix} \begin{bmatrix} A_\Gamma \\ \Gamma \end{bmatrix} \begin{bmatrix} Q - P - \epsilon I \end{bmatrix} \preceq 0, \end{aligned} \quad (16)$$

which is again an SDP problem.

A good initial point is crucial for ensuring the effective convergence of the proposed iterative algorithm. Many studies have demonstrated that, in certain cases, a controller designed based on a Koopman linear system $z_{k+1} = A_L \cdot z_k + B_L \cdot u_k$ can stabilize the original nonlinear system, particularly when operating within a small neighborhood around the equilibrium. As a result, utilizing the controller Γ_{lqr} derived from the Linear Quadratic Regulator of a Koopman linear system, serves as an effective initial point. The implementation details are provided in Algorithm 1. Note that here the Koopman observations and the Koopman-lifted dynamics are assumed to have been derived using a data-driven approach, which will be discussed in detail in the subsequent section.

Algorithm 1 Iterative Koopman Optimal Controller Search

- 1: **Set:** $Q = Q^T \succeq 0$, $R = R^T \succ 0$, $\zeta > 0$
- 2: **Derive:** $\Gamma_{lqr} \leftarrow dlqr(A_L, B_L, Q, R)$
- 3: **Initialize:** $\Gamma_0 \leftarrow \Gamma_{lqr}$, $P_0 \leftarrow \text{solve (14) with } \Gamma = \Gamma_0$.
 $\lambda_0 \leftarrow \lambda_{max}(P_0)$
- 4: **for** $k = 0, 1, 2, \dots$ **do**
- 5: $\Gamma_{k+1} \leftarrow \text{solve (16) with } P = P_k$
- 6: $P_{k+1} \leftarrow \text{solve (14) with } \Gamma = \Gamma_{k+1}$
- 7: $\lambda_{k+1} \leftarrow \lambda_{max}(P_{k+1})$
- 8: **if** $|\lambda_{k+1} - \lambda_k| < \zeta$ **then**
- 9: $\Gamma \leftarrow \Gamma_{k+1}$
- 10: **break**
- 11: **Return** Γ

IV. DATA-DRIVEN KOOPMAN APPROXIMATION USING DNN

Let $X = [x_1, \dots, x_N]$ and $U = [u_1, \dots, u_{N-1}]$ represent the N consecutive measurements of system (1). If the Koopman observables $g(x)$ are fixed, the state measurements can be mapped into the Koopman-lifted space, where the first $N-1$ lifted measurements is denoted by $Z = [z_1, \dots, z_{N-1}]$, while $Z_+ = [z_2, \dots, z_N]$ refers to the left-shifted lifted measurements. Define $A_h := [A, B_0, \dots, B_m]$ to be the concatenated system matrices in (7b), which can be obtained by solving the following least-squares problem:

$$\begin{aligned} A_h^* & := \arg \min_{A_h \in \mathbb{R}^{p \times (m+p+m)}} \left\| Z_+ - A_h \cdot \begin{bmatrix} Z \\ U \\ U \otimes Z \end{bmatrix} \right\|_F \\ & = Z_+ \cdot \begin{bmatrix} Z \\ U \\ U \otimes Z \end{bmatrix}^\dagger, \end{aligned} \quad (17)$$

where $U \otimes Z := [u_1 \otimes z_1, \dots, u_{N-1} \otimes z_{N-1}]$. The same process can be applied to calculate the concatenated multiplication-closed matrices $\tilde{B}_h := [\tilde{B}_1^T, \dots, \tilde{B}_p^T]^T$, resulting in

$$\tilde{B}_h^* = (Z \otimes Z) \cdot Z^\dagger. \quad (18)$$

While the system identification process is straightforward when the Koopman observables are fixed, simultaneously determining the Koopman observables and identifying the system remains a challenging task. Generating observables from a predefined function library is simple but can result in significant approximation errors if the chosen functions do not sufficiently approximate a Koopman-invariant subspace. The approximation error will further increase by enforcing the multiplication-closed property when designing the optimal controller, as outlined in Section III. Therefore, a DNN-based learning architecture is introduced to jointly learn the Koopman observables and identify the system matrices, which better satisfy the properties described in previous

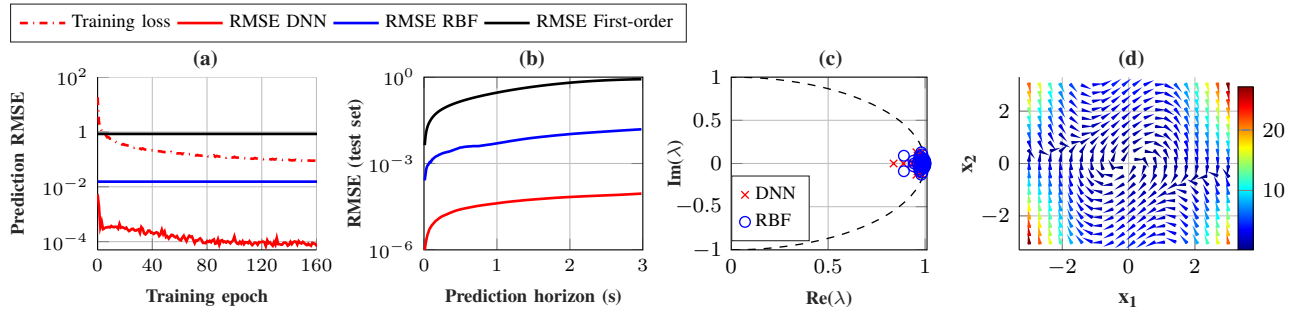


Fig. 2: Training process and model accuracy. (a) The training process, showing the average RMSE after 3s of consecutive prediction on the test set, evaluated at each training epoch. (b) Step-wise prediction RMSE comparison on the test set after training. (c) Eigenvalues distribution of the drifted system matrix A for the learned model. (d) Vector field of the DNN-trained model.

sections.

The DNN parameters are denoted by θ . As illustrated in Fig. 1, each forward pass through the DNN represents a Koopman observable, such that the lifted state is given by $z_k = g(x_k; \theta)$. Both $A_h^*(\theta)$ derived in (17) and the $\tilde{B}_h^*(\theta)$ derived in (18) depend on the DNN parameters. Consequently, the bilinear prediction loss therefore can be formulated as

$$L_{bi} := \frac{1}{(N-1)p} \sum_{k=1}^{N-1} \left\| z_{k+1}(\theta) - A_h^*(\theta) \cdot \begin{bmatrix} z_k(\theta) \\ \mathbf{u}_k \\ \mathbf{u}_k \otimes z_k(\theta) \end{bmatrix} \right\|_F^2, \quad (19)$$

and the loss associated with the multiplication-closed property is given by

$$L_{mc} := \frac{1}{(N-1)p} \sum_{k=1}^{N-1} \left\| z_k(\theta) \otimes z_k(\theta) - \tilde{B}_h^*(\theta) \cdot z_k(\theta) \right\|_F^2. \quad (20)$$

The overall training loss is written as

$$L := w_1 \cdot L_{bi} + w_2 \cdot L_{mc} + w_3 \cdot \|A_h^*(\theta)\|_{L_1} + w_4 \cdot \|\theta\|_{L_2}, \quad (21)$$

where $\|A_h^*(\theta)\|_{L_1}$ represents the L_1 regularization applied to the system matrices, indicating the sparsity penalty as shown in [2]. Additionally, $\|\theta\|_{L_2}$ denotes the L_2 regularization for weight decay, and $w_i, i \in [1, 4]_{\mathbb{Z}}$, are the weights assigned to each loss term.

V. LEARNING AND CONTROL RESULTS

The proposed training and control methods are validated on a Van del Pol oscillator, which has dynamics described as

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= (1 - x_1^2) \cdot x_2 - x_1 + u. \end{aligned} \quad (22)$$

To ensure sufficient data coverage across the state-input space $\mathcal{X} \times \mathcal{U}$, 400 trajectories are generated with initial states uniformly distributed within $[-3, 3] \times [-3, 3]$. Each trajectory lasts 10s sampled at $t_s = 0.01s$, and evolves either autonomously or under an LQR controller designed using the

linearized model around the origin $\mathbf{x} = [0, 0]^T$. Gaussian noise is introduced to the control signal ($u_k = -\gamma_{lqr} \cdot \mathbf{x}_k + \mathcal{N}(0, \sigma^2)$) to explore the system dynamics more thoroughly. The Koopman observation space has $p = 50$ dimensions, including the original states. The DNN-based approach is compared with the observations selected from RBF kernel, where $g_i(\mathbf{x}) := \|\mathbf{x} - \mathbf{p}_i\|^2 \cdot \log(\|\mathbf{x} - \mathbf{p}_i\|)$, $i \in [1, p]_{\mathbb{Z}}$, and each point \mathbf{p}_i are randomly sampled from $[-3, 3] \times [-3, 3]$. The DNN contains 4 fully connected layers with 48 output dimensions (excluding the original states) and employs *Tanh* activation. The model is trained in Pytorch using *Adam* optimizer with a learning rate of $1e-3$, with 85% of the measurements used for training and 15% for testing. Simulations were performed on a workstation with RTX4090 24GB GPU and Intel Core i7-14700KF CPU.

Fig. 2 presents the training process and model accuracy evaluation. As shown in Fig. 2(a), the prediction error significantly decreases during training, with the consecutive prediction RMSE on the test set reduced by approximately two orders of magnitude, indicating notable improvements in model accuracy. The final model outperforms both the first-order linearization around the origin and the RBF-based method, as illustrated in Fig. 2(b). By analyzing the autonomous part of the system through the eigendecomposition of the A matrix, Fig. 2(c) reveals that most eigenvalues are located on the unit circle, reflecting the non-decaying nature of the unactuated Van del Pol oscillator. Fig. 2(d) shows the vector field evaluated on the DNN-trained system, which corresponds closely to the actual behavior of the system that converges to a limit cycle.

To evaluate the multiplication-closed property and the optimal control performance of the proposed approach, the controller was first derived using Algorithm 1 over several iterations. Fig. 3 shows the decrease in $\lambda_{max}(P)$, indicating that Algorithm 1 is effectively identifying suitable (Γ, P) pairs that satisfy (8) while reducing the upper bound of the optimal cost for the lifted system. By comparing the control performance in Fig. 4, it is clear that both the DNN- and RBF-based controllers successfully stabilize the system at the origin, with the DNN-trained model exhibiting faster convergence. The dashed lines indicate the predicted trajectories using the closed-loop matrix A_{Γ} in (10) from the given initial

state. The RBF-based model predictions diverge beyond a certain time horizon. This divergence is due to its inability to accurately satisfy the multiplication-closed property, which in turn compromises the accuracy of the derived closed-loop matrix for controller design. In contrast, the predictions from the DNN-trained model closely align with the actual trajectory, which demonstrates that the multiplication-closed property is hold with minimal truncation error.

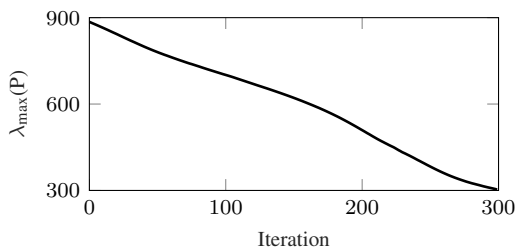


Fig. 3: Changes of $\lambda_{max}(P)$ when iterating Algorithm 1 using MOSEK with CVXOPT.

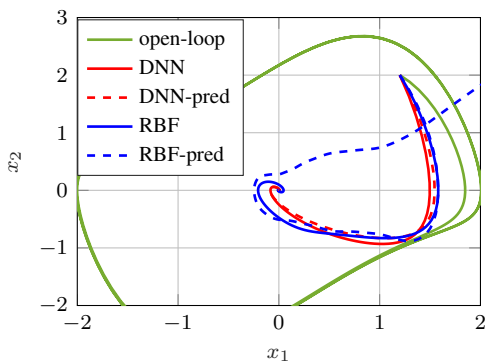


Fig. 4: Control performance evaluation. The solid lines represent the unactuated system behavior, as well as the control performance under the controllers designed using the DNN and RBF-based approaches. The dashed lines illustrate the predicted trajectories using the corresponding closed-loop linear matrices A_Γ , demonstrating the satisfaction of the multiplication-closed property.

VI. CONCLUSIONS

This paper presents a purely data-driven method for learning Koopman bilinear representations of control-affine systems and designing linear optimal controllers to stabilize the learned system. By employing DNNs, the proposed method identifies a finite set of Koopman observables that satisfy the multiplication-closed property and approximate a Koopman-invariant subspace with low error, enabling precise conversion of the Koopman bilinear system into a linear closed-loop system under a linear controller. After deriving the optimality condition for the optimal control problem in the Koopman-lifted space, an iterative algorithm is employed to compute the linear optimal controller by minimizing the upper bound of the optimal cost. The performance is validated on the Van der Pol oscillator and compared with the RBF-based method, where our proposed approach outperforms in both prediction accuracy and control effectiveness. The DNN-trained model not only exhibited faster convergence when stabilizing the

system to the origin, but also preserved the multiplication-closed property with minimal truncation error.

REFERENCES

- [1] G. P. Fettweis and H. Boche, "6g: The personal tactile internet—and open questions for information theory," *IEEE BITS the Information Theory Magazine*, vol. 1, no. 1, pp. 71–82, 2021.
- [2] S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," *Proceedings of the national academy of sciences*, vol. 113, no. 15, pp. 3932–3937, 2016.
- [3] D. Bruder, B. Gillespie, C. D. Remy, and R. Vasudevan, "Modeling and control of soft robots using the koopman operator and model predictive control," *arXiv preprint arXiv:1902.02827*, 2019.
- [4] A. Carron, E. Arcari, M. Wermelinger, L. Hewing, M. Hutter, and M. N. Zeilinger, "Data-driven model predictive control for trajectory tracking with a robotic arm," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3758–3765, 2019.
- [5] B. O. Koopman, "Hamiltonian systems and transformation in hilbert space," *Proceedings of the National Academy of Sciences*, vol. 17, no. 5, pp. 315–318, 1931.
- [6] C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. S. Henningson, "Spectral analysis of nonlinear flows," *Journal of fluid mechanics*, vol. 641, pp. 115–127, 2009.
- [7] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor, *Dynamic mode decomposition: data-driven modeling of complex systems*. SIAM, 2016.
- [8] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, "A data-driven approximation of the koopman operator: Extending dynamic mode decomposition," *Journal of Nonlinear Science*, vol. 25, pp. 1307–1346, 2015.
- [9] E. Kaiser, J. N. Kutz, and S. L. Brunton, "Data-driven discovery of koopman eigenfunctions for control," *Machine Learning: Science and Technology*, vol. 2, no. 3, p. 035023, 2021.
- [10] B. Lusch, J. N. Kutz, and S. L. Brunton, "Deep learning for universal linear embeddings of nonlinear dynamics," *Nature communications*, vol. 9, no. 1, p. 4950, 2018.
- [11] E. Yeung, S. Kundu, and N. Hodas, "Learning deep neural network representations for koopman operators of nonlinear dynamical systems," in *2019 American Control Conference (ACC)*. IEEE, 2019, pp. 4832–4839.
- [12] Y. Han, W. Hao, and U. Vaidya, "Deep learning of koopman representation for control," in *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE, 2020, pp. 1890–1895.
- [13] P. Huang, K. Zheng, and G. Fettweis, "Data-driven koopman operator-based error-state kalman filter for enhanced state estimation of quadrotors in agile flight," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024.
- [14] M. Korda and I. Mezić, "Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control," *Automatica*, vol. 93, pp. 149–160, 2018.
- [15] S. L. Brunton, B. W. Brunton, J. L. Proctor, and J. N. Kutz, "Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control," *PloS one*, vol. 11, no. 2, p. e0150171, 2016.
- [16] K. Zheng, P. Huang, and G. P. Fettweis, "Optimal control of quadrotor attitude system using data-driven approximation of koopman operator," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 834–840, 2023.
- [17] C. Folkestad, D. Pastor, I. Mezić, R. Mohr, M. Fonoberova, and J. Burdick, "Extended dynamic mode decomposition with learned koopman eigenfunctions for prediction and control," in *2020 american control conference (acc)*. IEEE, 2020, pp. 3906–3913.
- [18] A. Surana, "Koopman operator based observer synthesis for control-affine nonlinear systems," in *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, 2016, pp. 6492–6499.
- [19] D. Bruder, X. Fu, and R. Vasudevan, "Advantages of bilinear koopman realizations for the modeling and control of systems with unknown dynamics," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4369–4376, 2021.
- [20] C. Folkestad and J. W. Burdick, "Koopman nmpc: Koopman-based learning and nonlinear model predictive control of control-affine systems," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 7350–7356.

- [21] C. Folkestad, S. X. Wei, and J. W. Burdick, "Koopnet: Joint learning of koopman bilinear models and function dictionaries with application to quadrotor trajectory tracking," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 1344–1350.
- [22] D. Bruder, X. Fu, R. B. Gillespie, C. D. Remy, and R. Vasudevan, "Data-driven control of soft robots using koopman operator theory," *IEEE Transactions on Robotics*, vol. 37, no. 3, pp. 948–961, 2020.
- [23] K. Zheng, P. Huang, A. Villamil, J. Casas, and G. P. Fettweis, "System-oriented learning: An efficient dnn learning approach for koopman bilinear representation with control," in *2024 10th International Conference on Control, Decision and Information Technologies (CoDIT)*. IEEE, 2024.
- [24] R. Strässer, J. Berberich, and F. Allgöwer, "Robust data-driven control for nonlinear systems using the koopman operator," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 2257–2262, 2023.
- [25] R. Strässer, M. Schaller, K. Worthmann, J. Berberich, and F. Allgöwer, "Koopman-based feedback design with stability guarantees," *IEEE Transactions on Automatic Control*, 2024.
- [26] X. Ma, B. Huang, and U. Vaidya, "Optimal quadratic regulation of nonlinear system using koopman operator," in *2019 American Control Conference (ACC)*. IEEE, 2019, pp. 4911–4916.