Data-Driven Optimal Control of Unknown Nonlinear Dynamical Systems Using the Koopman Operator

Zhexuan Zeng XUANXUAN@HUST.EDU.CN
Department of Automatic Control, Huazhong University of Science and Technology, Wuhan, China.

Ruikun Zhou RUIKUN.ZHOU@UWATERLOO.CA
Department of Applied Mathematics, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada.

Yiming Meng YMMENG@ILLINOIS.EDU
Coordinated Science Laboratory, University of Illinois Urbana-Champaign, Urbana, IL 61801, USA.

Jun Liu J.LIU@UWATERLOO.CA
Department of Applied Mathematics, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada.

Editors: N. Ozay, L. Balzano, D. Panagou, A. Abate

Abstract

Nonlinear optimal control is vital for numerous applications but remains challenging for unknown systems due to the difficulties in accurately modelling dynamics and handling computational demands, particularly in high-dimensional settings. This work develops a theoretically certifiable framework that integrates a modified Koopman operator approach with model-based reinforcement learning to address these challenges. By relaxing the requirements on observable functions, our method incorporates nonlinear terms involving both states and control inputs, significantly enhancing system identification accuracy. Moreover, by leveraging the power of neural networks to solve partial differential equations (PDEs), our approach is able to achieve stabilizing control for high-dimensional dynamical systems, up to 9-dimensional. The learned value function and control laws are proven to converge to those of the true system at each iteration. Additionally, the accumulated cost of the learned controller closely approximates that of the true system, with errors ranging from 10^{-5} to 10^{-3}.

Keywords: Nonlinear optimal control, system identification, policy iteration, Koopman operator

1. Introduction

A central problem in control engineering is nonlinear optimal control, which has broad applications across various fields, including autonomous vehicle navigation, satellite and spacecraft control, and robotic manipulators.

A natural approach to pursue optimal control for continuous-time nonlinear dynamical systems is first linearizing the system at each state, representing the nonlinear dynamics as a state-dependent linear system. This allows the control law to be derived by solving state-dependent Riccati equation (Farsi et al., 2022). However, this approach typically yields only a sub-optimal controller. An alternative method for solving the optimal control problem involves addressing the Hamilton-Jacobi-Bellman (HJB) equation. Since the HJB equation is a nonlinear partial differential equation that is notoriously difficult to solve directly, most research focuses on obtaining approximate solutions indirectly through policy iteration techniques (Leake and Liu, 1967; Saridis and Lee, 1979; Beard, 1995; Jiang and Jiang, 2017). Originating from the optimal control of Markov decision processes (Bellman et al., 1957; Howard, 1960), policy iteration begins with an initial stabilizing

control and iteratively improves the closed-loop performance through two key steps: policy evaluation and policy improvement. Specifically, policy evaluation involves solving the Generalized Hamilton-Jacobi-Bellman (GHJB) equation, a linear partial differential equation that is generally more tractable than the HJB equation. For low-dimensional problems, Galerkin approximations have demonstrated their effectiveness in providing accurate solutions to the HJB equation with arbitrary precision (Beard et al., 1997, 1998). To overcome the curse of dimensionality in high-dimensional systems, neural networks are increasingly employed to approximate the solution to the GHJB equation. These networks ensure convergence to the true solution at each iteration, leveraging their ability to approximate complex functions and scale efficiently with problem size (Meng et al., 2024a; Zhou et al., 2024a).

However, solving the GHJB equation requires complete knowledge of system dynamics, which is often unavailable in practice. To address this challenge, various methods based on adaptive dynamic programming (ADP) have been developed to approximate the value function and control laws directly from online measurements, such as, (Jiang and Jiang, 2012; Vrabie and Lewis, 2009; Jiang and Jiang, 2014). These model-free methods are particularly advantageous when abundant data are available, as they enable the use of advanced techniques such as deep learning, to address high-dimensional state space problems. Despite their flexibility, model-free methods often suffer from a lack of theoretical guarantees and involve high implementation complexity. In contrast, model-based methods can use established control theories—such as stability analysis, performance optimization—to design control laws with rigorous guarantees. However, their effectiveness heavily depends on the accuracy of the identified model, which can be challenging to achieve in high-dimensional scenarios.

In recent years, the Koopman operator (Koopman, 1931) has gained significant attention due to its ability to provide a linear representation of nonlinear systems within a function space. Through numerical algorithms such as Dynamic Mode Decomposition (DMD) (Schmid, 2010) and Extended DMD (Williams et al., 2015; Korda and Mezić, 2018a), Koopman operator-based methods have shown strong efficacy in system identification (Mauroy and Gonçalves, 2019), in analyzing identifiability in relation to sampling frequency (Zeng et al., 2022, 2024b,a), and in stability analysis (Mauroy and Mezić, 2016; Deka et al., 2022; Meng et al., 2025; Zhou et al., 2024b) for autonomous systems. To address control problems, many studies have explored representing nonlinear systems with inputs using the Koopman operator (Korda and Mezić, 2018b; Mauroy et al., 2020). Although theoretically feasible by treating inputs as augmented states, practical implementation of this framework remains challenging, as it often necessitates neglecting certain terms to render the system fully linear in both the lifted state and the input. This limitation reduces both the accuracy of the identified system and the effectiveness of the resulting controllers, especially for high-dimensional systems.

To address the aforementioned limitations, the main contributions of this paper are as follows:

- We develop a theoretically certifiable framework, integrating a modified Koopman operator approach with model-based reinforcement learning, for control system identification and deriving optimal control policies for unknown nonlinear systems.
- We improve the Koopman operator-based identification accuracy by relaxing the requirements on observable functions, allowing accurate recovery of nonlinear control-affine terms.
- We enhance the scalability of computing optimal control directly from data by leveraging the power of neural networks to solve PDEs. We demonstrate the effectiveness of our method via 4 examples with state dimensions ranging from 2 to 9 and input dimensions up to 4.

2. Problem formulation

We consider a control-affine dynamical system:

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u}, \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ denotes the state vector; $\boldsymbol{u} \in \mathbb{R}^m$ denotes the input; $f : \mathbb{R}^n \to \mathbb{R}^n$ is a continuously differentiable vector field, and $g : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ is a smooth function. We also assume that $\mathbf{0}$ is an equilibrium point of (1), i.e., $f(\mathbf{0}) = \mathbf{0}$. Subject to the control $\boldsymbol{u}$, the unique solution starting from $\boldsymbol{x}_0$ is denoted by $S^t(\boldsymbol{x}_0, \boldsymbol{u})$.

We define the space of admissible controls of (1) as follows.

Definition 1 (Admissible control) *A feedback control $\boldsymbol{u} = \kappa(\boldsymbol{x})$ is admissible on $\Omega \subset \mathbb{R}^n$, where $\mathbf{0} \in \Omega$ and $\Omega \subset \mathbb{R}^n$ is a fixed compact set, if the following conditions are satisfied: (1) κ is Lipschitz continuous on Ω; (2) $\kappa(\mathbf{0}) = \mathbf{0}$; (3) $\boldsymbol{u} = \kappa(\boldsymbol{x})$ is a stabilizing control, i.e., $\lim_{t\to\infty} |S^t(\boldsymbol{x}_0, \boldsymbol{u})| = 0$ for $\forall \boldsymbol{x}_0 \in \Omega$. We denote the space of admissible feedback controls on Ω as $\mathcal{U}(\Omega)$.*

We are interested in finding the optimal control $\kappa^*(\boldsymbol{x}) \in \mathcal{U}(\Omega)$ from data, in the case of infinite interval $t \in [0, \infty)$. Specifically, we introduce the function $L(\boldsymbol{x}, \boldsymbol{u}) = Q(\boldsymbol{x}) + \|\boldsymbol{u}\|_R^2$, where $Q : \mathbb{R}^n \to \mathbb{R}$ is a positive definite function, and $\|\boldsymbol{u}\|_R^2 = \boldsymbol{u}^T R\boldsymbol{u}$ given some symmetric and positive definite $R : \mathbb{R}^n \to \mathbb{R}^{m\times m}$. The associated cost is commonly defined as follows:

$$J(\boldsymbol{x}, \boldsymbol{u}) = \int_0^\infty L(S^t(\boldsymbol{x}, \boldsymbol{u}), \boldsymbol{u}(t))\mathrm{d}t \tag{2}$$

The optimal control is denoted as $\boldsymbol{u}^* = \kappa^*(\boldsymbol{x})$ such that

$$J(\boldsymbol{x}, \boldsymbol{u}^*) = \inf_{\kappa\in\mathcal{U}} J(\boldsymbol{x}, \kappa(\boldsymbol{x})), \tag{3}$$

and the value function is defined as $V(\boldsymbol{x}) := J(\boldsymbol{x}, \boldsymbol{u}^*)$. Intuitively, the value function describes the system's infimum energy loss over the state space for all possible control inputs, thereby providing a foundation for deriving the optimal control $\boldsymbol{u}^* = \kappa^*(\boldsymbol{x})$. Let DV denote the gradient of V. The optimal control is derived by minimizing the Hamiltonian:

$$\kappa^*(\boldsymbol{x}) := \mathrm{argmin}_{\boldsymbol{u}\in\mathcal{U}}\{L(\boldsymbol{x}, \boldsymbol{u}) + DV(\boldsymbol{x}) \cdot f(\boldsymbol{x}, \boldsymbol{u})\} = -\frac{1}{2}R^{-1}g^T(\boldsymbol{x})(DV(\boldsymbol{x}))^T. \tag{4}$$

This paper aims to systematically explore a theoretically certifiable method for computing optimal control from data in unknown high-dimensional nonlinear systems.

3. Preliminaries of exact policy iteration and Koopman operator theory

3.1. Exact policy iteration

We begin by reviewing the policy iteration method for systems with known dynamics. The value function that we aim to find is generally a viscosity solution to the HJB equation, i.e.,

$$H(\boldsymbol{x}, DV(\boldsymbol{x})) = 0, \tag{5}$$

where $H(\boldsymbol{x}, p) := \sup_{\boldsymbol{u}\in\mathbb{R}^m} -G(\boldsymbol{x}, \boldsymbol{u}, p)$, and $G(\boldsymbol{x}, \boldsymbol{u}, p) = L(\boldsymbol{x}, \boldsymbol{u}) + p(f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u})$. However, solving and analyzing this equation is a complex task. The policy iteration method assumes that

$V \in C^1(\Omega)$ and seeks C^1 solutions V_i to the GHJB equation $G(\boldsymbol{x}, \boldsymbol{u}_i, DV_i(\boldsymbol{x})) = 0$ for each iteration $i \in \{0, 1, \cdots\}$, specifically, given an initial input $\boldsymbol{u}_0 = \kappa_0(\boldsymbol{x})$ that stabilizes the system, the policy iteration method performs policy evaluation and policy improvement iteratively:

1 The i-th policy evaluation: Given a policy $\boldsymbol{u}_i = \kappa_i(\boldsymbol{x})$, solving the GHJB equation below to compute the value function $V_i(\boldsymbol{x})$ at $x \in \Omega \setminus \{\mathbf{0}\}$, and we set $V_i(\mathbf{0}) = 0$.

$$G(\boldsymbol{x}, \kappa_i(\boldsymbol{x}), DV_i(\boldsymbol{x})) := L(\boldsymbol{x}, \kappa_i(\boldsymbol{x})) + DV_i(\boldsymbol{x})(f(\boldsymbol{x}) + g(\boldsymbol{x})\kappa_i(\boldsymbol{x})) = 0. \tag{6}$$

2 The i-th policy improvement: Given the value function $V_i(\boldsymbol{x})$, solving the GHJB to update the policy:

$$\kappa_i(\boldsymbol{x}) = \begin{cases} -\frac{1}{2}R^{-1}g^T(\boldsymbol{x})(DV_i(\boldsymbol{x}))^T, & \boldsymbol{x} \neq \mathbf{0}; \\ \mathbf{0}, & \boldsymbol{x} = \mathbf{0}. \end{cases} \tag{7}$$

Assuming that $V \in C^1(\Omega)$, the convergence value function V_∞ is expected to solve the HJB equation and $\boldsymbol{u}_i \to \boldsymbol{u}^*$ at least pointwise (Jiang and Jiang, 2017, Theorem 3.1.4).

3.2. Koopman operator theory

To solve (6) for unknown systems, we first identify $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$. Koopman operator provides an alternative perspective to analyze and learn nonlinear dynamical systems. Below we briefly introduce the Koopman operator theory. Let us first consider $\boldsymbol{u} = \mathbf{0}$. Then the system (1) becomes:

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}). \tag{8}$$

The flow induced by this autonomous system is denoted as $S^t(\boldsymbol{x}, \mathbf{0}), t > 0$, i.e., $\boldsymbol{x}(t) = S^t(\boldsymbol{x}(0), \mathbf{0})$. The Koopman operator $U^t : \mathcal{F} \to \mathcal{F}$ is a linear operator acting on the observable functions of the states, i.e., $\varphi \in \mathcal{F} : \mathbb{R}^n \to \mathbb{C}$, which is defined as

$$U^t\varphi(\boldsymbol{x}) = \varphi(S^t(\boldsymbol{x}, \mathbf{0})). \tag{9}$$

The infinitesimal generator $\mathcal{L}$ of the Koopman operator is defined as

$$\mathcal{L}\varphi = \lim_{t \to 0^+} \frac{1}{t}(U^t - I)\varphi, \ \varphi \in \mathcal{D}(\mathcal{L}), \tag{10}$$

where $\mathcal{D}(\mathcal{L})$ denotes the domain of $\mathcal{L}$. The generator is also a linear operator. Assuming that observable functions $g \in \mathcal{F}$ are continuously differentiable with compact support, we have $\mathcal{L} = f \cdot \nabla$.

Due to its linearity and rich theoretical support, the Koopman operator theory enjoys wide popularity in nonlinear system identification and control. Despite its success, accurately representing a nonlinear system with input as a linear input-output system remains challenging, as it often requires disregarding certain terms that describe how control actions evolve in the observation space.

4. Description of the method

The main idea of the method is to first identify the nonlinear dynamical system with control using the generator. Then we solve the optimal control problems for it using the policy iteration method with its neural approximations.

4.1. Lifting of the dynamical system

The nonlinear system is equivalently described as an infinite-dimensional linear system driven by the Koopman operator. In practice, we lift and embed the original system into a high-dimensional function space, then approximate the Koopman operator and its generator using a linear, matrix-like operator defined on the same function space domain.

Theoretically, we treat the input $\boldsymbol{u}$ of (1) as an external state of the dynamical system and assume that it remains unchanged during the sampling time. Then we have the extended system:

$$\begin{aligned} \dot{\boldsymbol{x}} &= f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u}, \\ \dot{\boldsymbol{u}} &= 0, \end{aligned} \tag{11}$$

and the corresponding extended state $[\boldsymbol{x}, \boldsymbol{u}]^T$. This implies that the control input $\boldsymbol{u}$ is considered constant. For simplicity of the notation, we also denote the flow of the extended system as $S^t : \mathcal{M} \to \mathcal{M}$ with the invariant set $\mathcal{M}$ of the extended states, where $\mathcal{M} \subseteq \mathbb{R}^{n+m}$.

To accurately characterize the original system, we recover the generator within an observable space where the basis functions $\{\varphi_i(\boldsymbol{x}, \boldsymbol{u})\}$ include coupling terms between $\boldsymbol{x}, \boldsymbol{u}$. This approach avoids constructing a high-dimensional linear input-output system of the form $\dot{\boldsymbol{z}} = A\boldsymbol{z} + B\boldsymbol{u}$, where $\boldsymbol{z} = \varphi(\boldsymbol{x})$ and φ is a vector-valued function consisting of multiple scalar observation functions that depend solely on $\boldsymbol{x}$. In this work, we select the polynomial observable functions of $(\boldsymbol{x}, \boldsymbol{u})$. Furthermore, to recover $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$ in the system, we restrict the total degree of each control input $u_i, i = 1, \ldots, m$ to 1 within the observable functions, i.e., $\mathcal{F}_N = \operatorname{span}_{i=1,\ldots,N}\{\varphi_i(\boldsymbol{x}, \boldsymbol{u})\}$, where $\varphi_i(\boldsymbol{x}, \boldsymbol{u}) = \prod_{j=1}^{n} x_j^{p_j} \prod_{l=1}^{m} u_l^{q_l}$, $\sum_{l=1}^{m} q_l \leq 1$, with x_j being the j-th component of $\boldsymbol{x}$, u_l being the l-th component of $\boldsymbol{u}$, $q_l, p_j \in \mathbb{N}$.

Remark 2 *Though it is common to learn the system using dictionary functions that are independent in u and x, and linear in u, given the control-affine form of the dynamics, the dictionary of basis functions may still provide a sparse subset of the lifted function space that fails to sufficiently span the generator domain. We offer a novel perspective by considering the coupling effect between x and u when selecting the basis functions for the lifted function space. Numerical results in the next section demonstrate that this approach improves learning accuracy compared to the aforementioned overly decoupled basis functions. This also highlights a promising direction for future theoretical research on control-involved dynamics and the properties of the associated lifted space.*

4.2. Identification of the generator

To avoid the approximation errors introduced by the indirect approach, we identify the generator using the Yosida approximation, which is based on the resolvent operator of the Koopman operator. The Yosida approximation is particularly suitable because it provides a well-defined approximation of the generator without requiring the observable space to be invariant under the Koopman operator. Moreover, it ensures better numerical stability and theoretical convergence properties compared to methods that rely on computing the logarithm of the Koopman operator.

The Yosida approximation L_λ, defined as

$$L_\lambda \varphi_i(\boldsymbol{x}) = \lambda^2 \int_0^\infty e^{-\lambda t} U^t \varphi_i(\boldsymbol{x}) \mathrm{d}t - \lambda \varphi_i(\boldsymbol{x}), \tag{12}$$

converges to the generator $\mathcal{L}$ on $C^1(\mathcal{M})$ as $\lambda \to \infty$ in a strong sense (Meng et al., 2024b, Theorem 3.3), i.e., $L_\lambda h \to \mathcal{L}h$ for any observable function $h \in C^1(\mathcal{M})$. In practice, the integral in (12) is approximated using a discrete set of trajectory data. Given a set of initial conditions $\{\boldsymbol{x}_k\}_{k=1}^N$, we collect the observable values along trajectories sampled at discrete time points t_j with a fixed time step Δt. Then $L_\lambda \varphi_i$ is approximated using empirical averages or numerical quadrature techniques. To ensure numerical tractability, we further approximate (12) for each observable function φ_i using a truncated integral over a fixed finite-time horizon $[0, T_{\max}]$, as follows:

$$L_{\lambda,T_{\max}}\varphi_i(\boldsymbol{x}) = \lambda^2 \int_0^{T_{\max}} e^{-\lambda t} U^t \varphi_i(\boldsymbol{x}) \mathrm{d}t - \lambda \varphi_i(\boldsymbol{x}). \tag{13}$$

Given the choice of polynomial observables $\{\varphi_i\}$, the overall approximation error of $\mathcal{L}\varphi_i$ (for each i) using $L_{\lambda,T_{\max}}\varphi_i$ is of the order $\mathcal{O}(e^{-\lambda T_{\max}})$, as stated in (Meng et al., 2024b, Theorem 4.2). Notably, for large λ, truncating the integral at any $T_{\max}$ has a non-dominant impact on the approximation accuracy. This error bound allows us to directly obtain the value of $\mathcal{L}\varphi_i$ using the evaluation of (13), and further adapt it with sampled data using numerical quadrature techniques. This allows us to construct two matrices $[X]_{i,j} = \varphi_j(\boldsymbol{x}_i), [Y]_{i,j} = L_{\lambda,T_{max}}\varphi_j(\boldsymbol{x}_i)$ where $\{\boldsymbol{x}_i\}_{i=1}^M$ denote M samples. Then we compute the matrix representation of the generator as $\hat{L}_N = (X^T X)^{-1} X^T Y$.

4.3. Recovery of nonlinear dynamical system

To recover $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$ from the identified generator, we denote the i as the index of the observable function such that $\varphi_i(\boldsymbol{x}) = x_j$. Then we have

$$\hat{f}_j(\boldsymbol{x}) + \hat{g}_j(\boldsymbol{x})\boldsymbol{u} = \sum_{k=1}^{N} \varphi_k(\boldsymbol{x}, \boldsymbol{u})[\hat{L}_N]_{k,i}, \tag{14}$$

where $[\hat{L}_N]_{k,j}$ represents the element in the k-th row and j-th column of $\hat{L}_N$. We can approximate $\hat{f}(\boldsymbol{x})$ and $\hat{g}(\boldsymbol{x})$ corresponding to observable functions $\varphi_k(\boldsymbol{x}, \boldsymbol{u}) = \prod_{j=1}^n x_j^{p_j} \prod_{l=1}^m u_l^{q_l}$ with $\sum_{l=1}^m q_l = 0$ and $\sum_{l=1}^m q_l = 1$, respectively.

4.4. Policy iteration via linear least squares

Based on the identified $\hat{f}(\boldsymbol{x})$ and $\hat{g}(\boldsymbol{x})$, we continue to solve the optimal control problem by employing policy evaluation (6) and policy improvement (7). Specifically, in the i-th iteration, we solve the following equation:

$$L(\boldsymbol{x}, \kappa_i(\boldsymbol{x})) + DV_i(\boldsymbol{x})(\hat{f}(\boldsymbol{x}) + \hat{g}(\boldsymbol{x})\kappa_i(\boldsymbol{x})) = 0, \tag{15}$$

$$\kappa_i(\boldsymbol{x}) = \begin{cases} -\frac{1}{2} R^{-1} \hat{g}^T(\boldsymbol{x})(DV_i(\boldsymbol{x}))^T, & \boldsymbol{x} \neq \mathbf{0}; \\ \mathbf{0}, \boldsymbol{x} = \mathbf{0}. \end{cases} \tag{16}$$

To solve (15) and approximate the value function, we use random feature neural network functions, resulting in a neural solution of the form $\hat{V}(\boldsymbol{x}) = \boldsymbol{\beta}^T \sigma(W\boldsymbol{x} + \boldsymbol{b})$, where $\boldsymbol{\beta} \in \mathbb{R}^s, W \in \mathbb{R}^{s \times n}, \boldsymbol{b} \in \mathbb{R}^s$, and $\sigma : \mathbb{R} \to \mathbb{R}$ is an activation function applied element-wise. It follows that $D\hat{V}(\boldsymbol{x}) = \boldsymbol{\beta}^T \operatorname{diag}(\sigma'(W\boldsymbol{x} + \boldsymbol{b}))W$. Note that W and $\boldsymbol{b}$ can be randomly chosen, which does not require training. Then the problem of solving (15) transforms to the problem of finding the parameter $\boldsymbol{\beta}$. Due to the linear dependence of $D\hat{V}(\boldsymbol{x})$ on $\boldsymbol{\beta}$ and the linearity of (15), it results in a linear least squares optimization problem that can be solved efficiently and accurately.

Remark 3 (Random bases) *It is well established in the theoretical machine learning community that, in high-order Sobolev norms, neural networks with the hyperbolic tangent activation function yield relatively small approximation errors for Sobolev-regular and analytic functions (De Ryck et al., 2021). Many applications, including the use of one-layer tanh-activated neural networks (or equivalently, linear combinations of tanh-activated random bases) to solve Hamilton–Jacobi-type PDEs, have shown strong performance under mild conditions (Zhou et al., 2024a). In the context of Koopman generator operator learning, where the ultimate goal is to approximate an image function expressible as a linear combination of basis functions, it is natural to adopt the set of tanh-activated basis functions as dictionary functions to achieve high precision. In particular, for unknown systems governed by unknown physical laws, where specialized approximation bases such as polynomials may not be suitable, a random feature approach using tanh-activated bases helps reduce bias in the selection of dictionary functions. Given both the theoretical and practical advantages, we adopt tanh-activated random bases in our framework.*

5. Theoretical convergence

We begin by proving the convergence of the identified system. In the following, we use $L_{\lambda,T_{\max},N}$ to denote N-dimensional approximation of $L_{\lambda,T_{\max}}$, $F_{\lambda,T_{\max},N}(\boldsymbol{x},\boldsymbol{u})$ to denote the vector field recovered from $L_{\lambda,T_{\max},N}$, and $F(\boldsymbol{x},\boldsymbol{u})$ to denote the original vector field.

Theorem 4 (Convergence of vector field) *As $\lambda \to \infty, T_{max} \to \infty, N \to \infty$ simultaneously, we have $F_{\lambda,T_{max},N} \to F$ uniformly on $\mathcal{M}$, where $\mathcal{M}$ is a compact set in $\mathbb{R}^{n+m}$.*

Proof For the i-th component of vector field, where $i = 1, \ldots, n$, we have $F_i(\boldsymbol{x},\boldsymbol{u}) = \mathcal{L}\varphi_q(\boldsymbol{x},\boldsymbol{u})$, $F_{i;\lambda,T_{max},N}(\boldsymbol{x},\boldsymbol{u}) = L_{\lambda,T_{max},N}\varphi_q(\boldsymbol{x},\boldsymbol{u})$, where q is the index of the observable function such that $\varphi_q(\boldsymbol{x},\boldsymbol{u}) = x_i$. It follows that

$$\|F_i - F_{i;\lambda,T_{max},N}\|_\infty \le \|(L - L_\lambda)\varphi_q\|_\infty + \|(L_\lambda - L_{\lambda,T_{max}})\varphi_q\|_\infty + \|(L_{\lambda,T_{max}} - L_{\lambda,T_{max},N})\varphi_q\|_\infty,$$

where $\|\cdot\|_\infty$ denotes $\sup_{(\boldsymbol{x},\boldsymbol{u})\in\mathcal{M}} \|\cdot\|$ with $\|\cdot\|$ being the 2-norm. Based on Theorem 3.3, Theorem 4.2 and Corollary 4.6 in Meng et al. (2024c), we have $\|F_i - F_{i;\lambda,T_{max},N}\|_\infty \to 0$ as $\lambda \to \infty, T_{max} \to \infty, N \to \infty$ simultaneously. ■

While Theorem 4 guarantees theoretical convergence of $F_{\lambda,T_{max},N}$ to F, as $\lambda \to \infty, T_{max} \to \infty, N \to \infty$, the following assumption states that, with sufficiently many samples, the identified system $\hat{F}(\boldsymbol{x},\boldsymbol{u}) = \hat{f}(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u}$ should be close to $F_{\lambda,T_{max},N}$.

Assumption 1 *For $\forall\theta > 0$, the initial conditions for (11) can be sampled sufficiently densely in $\mathcal{M}$ such that $\hat{F}(\boldsymbol{x},\boldsymbol{u}) = \hat{f}(\boldsymbol{x}) + \hat{g}(\boldsymbol{x})\boldsymbol{u}$ identified from (14) satisfies $\|F_{\lambda,T_{\max},N} - \hat{F}\|_\infty \le \theta$.*

Remark 5 *The approximation scheme consists of an analytical approximation of the generator using a finite-horizon, finite-dimensional modification of the Yosida approximation framework, followed by data fitting using least squares for each dictionary function. The first stage has been proven to converge with a tunable error, while the second stage follows the data-fitting framework presented in (Williams et al., 2015), where densely populated initial conditions are generally accepted to ensure convergence. Due to space limitations, we omit the proof, as it directly follows from combining the two aforementioned stages of approximation. While data efficiency, such as achieving quantitative error guarantees using minimal set of spatio-temporal data, is beyond the scope of this paper, it is an important direction the authors intend to pursue in future work.*

Under Assumption 1 and based on Theorem 4, we can conclude that, for every $\theta > 0$, there exist sufficiently large $\lambda, T_{\max}, N$ and sufficiently dense initial conditions such that

$$\|f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u} - \hat{f}(\boldsymbol{x}) - \hat{g}(\boldsymbol{x})\boldsymbol{u}\| \leq \theta, \quad \forall(\boldsymbol{x}, \boldsymbol{u}) \in \mathcal{M}. \tag{17}$$

Without loss of generality, we assume that (17) holds for $\boldsymbol{u} \in B = \{\|\boldsymbol{u}\| \leq 1\}$. Letting $\boldsymbol{u} = \mathbf{0}$, we have $\|f - \hat{f}\|_\infty \leq \theta$. It follows from the triangle inequality that $\|g(\boldsymbol{x})\boldsymbol{u} - \hat{g}(\boldsymbol{x})\boldsymbol{u}\| \leq 2\theta$. With $\boldsymbol{u} = \frac{g(\boldsymbol{x})-\hat{g}(\boldsymbol{x})}{\|g(\boldsymbol{x})-\hat{g}(\boldsymbol{x})\|} \in B$, this implies $\|g - \hat{g}\|_\infty \leq 2\theta$. In other words, for every $\theta > 0$, there exist sufficiently large $\lambda, T_{\max}, N$ and sufficiently dense initial conditions such that $\|f - \hat{f}\|_\infty \leq \theta, \|g - \hat{g}\|_\infty \leq 2\theta$, which are essential requirements for the following analysis.

We expect each policy evaluation of the identified system to closely approximate that of the true system, ensuring that the algorithm ultimately produces a meaningful result. The following theorem establishes that, with each iteration, the value function and control derived from the identified system converge to those of the true system. The proof is provided in the preprint version (Zeng et al., 2024c). For brevity, we directly use the index $h = (\lambda, T_{max}, N)$, and $f_h(\boldsymbol{x}) + g_h(\boldsymbol{x})\boldsymbol{u}$ to denote the identified vector field from data.

Theorem 6 *Let $\Omega \subset \mathcal{M}$ a compact invariant set for each $\dot{\boldsymbol{x}} = F_h^{(i)}(\boldsymbol{x}, \kappa_h^{(i-1)}(\boldsymbol{x})) = f_h(\boldsymbol{x}) + g_h(\boldsymbol{x})\kappa_h^{(i-1)}(\boldsymbol{x})$ and $\dot{\boldsymbol{x}} = F^{(i)}(\boldsymbol{x}, \kappa^{(i-1)}(\boldsymbol{x})) = f(\boldsymbol{x}) + g(\boldsymbol{x})\kappa^{(i-1)}(\boldsymbol{x})$. Assume $F_h^{(i)}$, $F^{(i)} \in C^1(\Omega)$, $L_h^{(i)}, L^{(i)} \in C^1(\Omega)$. Then, for every $\theta > 0$ and every $i \geq 0$, there exists $\delta > 0$ such that if $\|\kappa_h^{(0)} - \kappa^{(0)}\|_\infty < \delta, \|f_h - f\|_\infty < \delta, \|g_h - g\|_\infty < \delta$, we have $\|V_h^{(i)} - V^{(i)}\|_\infty < \theta, \|\kappa_h^{(i)} - \kappa^{(i)}\|_\infty < \theta$ for all $\boldsymbol{x} \in \Omega$.*

6. Numerical experiments

6.1. Experimental setup

To demonstrate the performance of the proposed method, we learn the optimal control for the following systems: 2-dimensional (inverted pendulum), 4-dimensional (cartpole), 6-dimensional (2D quadrotor), and 9-dimensional systems (3D quadrotor).

In the identification step: For 2-dimensional and 4-dimensional systems, we select the space of polynomials $\varphi_i(\boldsymbol{x}, \boldsymbol{u}) = \prod_{j=1}^n x_j^{p_j} \prod_{l=1}^m u_l^{q_l}$ where $p_j \leq p_{max}$ for $j = 1, \ldots, n$. For 6-dimensional and 9-dimensional systems, we select the space of polynomials constrained by $\sum_{j=1}^n p_j \leq p_{sum}$. To ensure that the identified system has an equilibrium at $\mathbf{0}$, we exclude constant functions from the set of observable functions, i.e., $\sum_{j=1}^n p_j + \sum_{l=1}^m q_l > 0$. For each system, we collect data with the time horizon $[0, 1]$ the sampling frequency 100Hz. The specific parameters and data details for identification and policy iteration steps are provided in Tables 1 and 2 below.

Table 1: The detailed information of data and parameters for identification

Dynamical system	Domain	Polynomial order	Initial samples
a) Inverted pendulum (2d)	$(\boldsymbol{x}, \boldsymbol{u}) \in [-1, 1]^3$	$p_{max} = 5$	1000
b) Cartpole (4d)	$(\boldsymbol{x}, \boldsymbol{u}) \in [-0.2, 0.2]^5$	$p_{max} = 3$	3125
c) 2D quatroter (6d)	$(\boldsymbol{x}, \boldsymbol{u}) \in [-0.2, 0.2]^8$	$p_{sum} = 3$	5000
d) 3D quadrotor (9d)	$(\boldsymbol{x}, \boldsymbol{u}) \in [-0.2, 0.2]^{13}$	$p_{sum} = 3$	10000

Table 2: The detailed information of data and parameters for policy iteration

Dynamical system	Domain	Hidden units (s)	Samples
a) Inverted pendulum	$\boldsymbol{x} \in [-1, 1]^2$	200	3000
b) Cartpole	$\boldsymbol{x} \in [-0.1, 0.1]^4$	3200	6000
c) 2D quadrotor	$\boldsymbol{x} \in [-0.1, 0.1]^6$	3200	9000
d) 3D quadrotor	$\boldsymbol{x} \in [-0.1, 0.1]^9$	3200	12000

6.2. Numerical results

1) Identification performance. To effectively solve the HJB equation, it is crucial to accurately estimate $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$ over the considered set. To demonstrate the advantages of the proposed method, we compare the evaluation error of our approach (the resolvent-based model) with the logarithm-based model proposed by Mauroy and Gonçalves (2019) and the widely used lifted linear model in control frameworks (Korda and Mezić, 2018b). The evaluation errors are calculated as $E_f = \sum_{i=1}^{M} \|f(\boldsymbol{x}_i) - \hat{f}(\boldsymbol{x}_i)\|_1/M, E_g = \sum_{i=1}^{M} \|g(\boldsymbol{x}_i) - \hat{g}(\boldsymbol{x}_i)\|_1/M$, where $\|\cdot\|_1$ denotes L^1-norm. To ensure a fair comparison, it is worth emphasizing that the basis functions used in the logarithm-based method are equivalent to those in our approach. Additionally, the resolvent-based identification method is employed to identify the lifted linear model, where the observable functions exclude cross-terms involving $\boldsymbol{x}$ and $\boldsymbol{u}$. To minimize the influence of the number of basis functions on the comparison, we further increase the polynomial order in the lifted linear model, ensuring its basis functions are at least equal to or more comprehensive than those used in our method.

The comparison results are detailed in Table 3. These results demonstrate that our method achieves evaluation errors for f and g that are reduced by one to two orders of magnitude compared to the other two approaches, which is particularly pronounced in high-dimensional systems. This improvement in accuracy is crucial for solving the HJB equation, as our attempts with the other two methods failed to learn a control law to stabilize the cartpole, 2D quadrotor, and 3D quadrotor. In contrast, the control law and value function learned using our method are presented below.

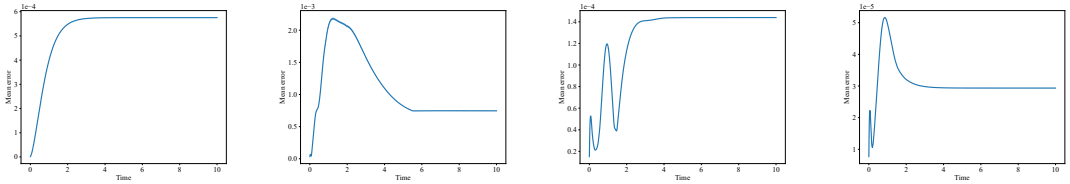
Table 3: Comparison of evaluation error for ours (resolvent-based control-affine model), LAM (logarithm-based control-affine model), and RLM (resolvent-based lifted linear model).

	Inverted pendulum		Cartpole		2D quadrotor		3D quadrotor	
	E_f	E_g	E_f	E_g	E_f	E_g	E_f	E_g
Ours	3.7E-3	**9.9E-3**	**3.5E-3**	**2.0E-3**	**8.6E-4**	**2.2E-3**	**1.7E-3**	**8.5E-3**
LAM	2.9E-1	6.4E-1	7.8E-2	1.4E-2	1.4E-2	2.1	2.4E-2	4.3E-2
RLM	2.8E-3	1.2E-2	1.8E-2	3.4E-2	9.5E-3	1.1E-1	1.8E-2	2.2E-1

2) Control performance. We randomly choose 50 initial conditions in the associated domain of Table 1, and we simulate these trajectories using the true system and the control learned from the identified system. To illustrate the performance of the learned control, we compute the average $\hat{C}(t)$ of the accumulated costs $\hat{C}_i(t), t \in [0, 10]$ for these 50 trajectories. We also perform the simulation and compute the average of the accumulated cost $C(t)$ using the learned control from the true system. The error of the mean accumulated cost $|\hat{C}(t) - C(t)|$ and the trajectories are depicted

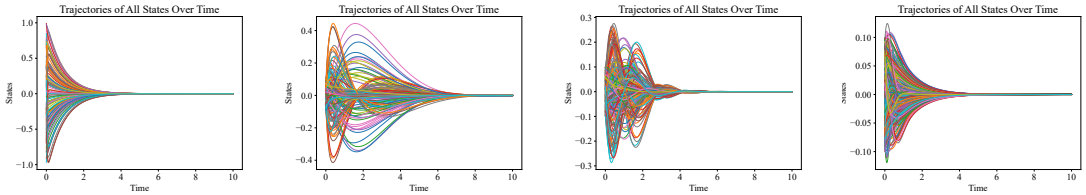
in Fig. 1 and Fig. 2 respectively. These results demonstrate that, when the dynamical system is unknown, the accumulated cost of the optimal control input obtained by this method closely aligns with that of the optimal control learned from the true system, with errors ranging from 10^{-5} to 10^{-3}. The optimal control input learned from data of this unknown system effectively stabilizes the trajectories of this true system.

For comparison, ADP (Jiang and Jiang, 2017, Chapter 3) performs well in low-dimensional cases, such as the 2D pendulum. However, it faces significant challenges in learning stable controllers for relatively higher-dimensional systems such as cartpole. Consequently, our method demonstrates strong potential for addressing optimal control problems in high-dimensional systems.



(*a*) Inverted pendulum (*b*) Cartpole (*c*) 2D quadrotor (*d*) 3D quadrotor

Figure 1: Error between accumulated costs computed by the controller learned from the identified system and the true system for the four examples.



(*a*) Inverted pendulum (*b*) Cartpole (*c*) 2D quadrotor (*d*) 3D quadrotor

Figure 2: Trajectories of n states with the learned controller for the four examples.

7. Conclusion

We proposed a novel approach for solving optimal control problems in high-dimensional nonlinear systems. The results demonstrated the effectiveness of our method in achieving stabilizing control and accurately approximating value functions, even for systems with state dimensions up to 9 and input dimensions up to 4. However, we acknowledge that the region of consideration for high-dimensional systems in this study is relatively limited. This reflects the inherent challenges in solving high-dimensional HJB equations, including computational complexity and sensitivity to identification errors. Addressing these limitations will be a key focus of our future research, aiming to broaden the stabilized region of the learned controls.

References

Randal W Beard, George N Saridis, and John T Wen. Galerkin approximations of the generalized Hamilton-Jacobi-Bellman equation. *Automatica*, 33(12):2159–2177, 1997.

Randal W Beard, George N Saridis, and John T Wen. Approximate solutions to the time-invariant Hamilton–Jacobi–Bellman equation. *Journal of Optimization theory and Applications*, 96:589–626, 1998.

Randal Winston Beard. *Improving the Closed-Loop Performance of Nonlinear Systems*. Rensselaer Polytechnic Institute, 1995.

R. Bellman, R.E. Bellman, and Rand Corporation. *Dynamic Programming*. Rand Corporation research study. Princeton University Press, 1957.

Tim De Ryck, Samuel Lanthaler, and Siddhartha Mishra. On the approximation of functions by tanh neural networks. *Neural Networks*, 143:732–750, 2021.

Shankar A Deka, Alonso M Valle, and Claire J Tomlin. Koopman-based neural lyapunov functions for general attractors. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 5123–5128. IEEE, 2022.

Milad Farsi, Yinan Li, Ye Yuan, and Jun Liu. A piecewise learning framework for control of unknown nonlinear systems with stability guarantees. In *Learning for Dynamics and Control Conference*, pages 830–843. PMLR, 2022.

R.A. Howard. *Dynamic Programming and Markov Processes*. Technology Press of Massachusetts Institute of Technology, 1960.

Yu Jiang and Zhong-Ping Jiang. Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics. *Automatica*, 48(10):2699–2704, 2012.

Yu Jiang and Zhong-Ping Jiang. Robust adaptive dynamic programming and feedback stabilization of nonlinear systems. *IEEE Transactions on Neural Networks and Learning Systems*, 25(5): 882–893, 2014.

Yu Jiang and Zhong-Ping Jiang. *Robust Adaptive Dynamic Programming*. John Wiley & Sons, 2017.

Bernard O Koopman. Hamiltonian systems and transformation in Hilbert space. *Proceedings of the National Academy of Sciences of the United States of America*, 17(5):315, 1931.

Milan Korda and Igor Mezić. On convergence of extended dynamic mode decomposition to the Koopman operator. *Journal of Nonlinear Science*, 28:687–710, 2018a.

Milan Korda and Igor Mezić. Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control. *Automatica*, 93:149–160, 2018b.

RJ Leake and Ruey-Wen Liu. Construction of suboptimal control sequences. *SIAM Journal on Control*, 5(1):54–63, 1967.

Alexandre Mauroy and Jorge Gonçalves. Koopman-based lifting techniques for nonlinear systems identification. *IEEE Transactions on Automatic Control*, 65(6):2550–2565, 2019.

Alexandre Mauroy and Igor Mezić. Global stability analysis using the eigenfunctions of the koopman operator. *IEEE Transactions on Automatic Control*, 61(11):3356–3369, 2016.

Alexandre Mauroy, Igor Mezić, and Yoshihiko Susuki. *The Koopman Operator in Systems and Control: Concepts, Methodologies, and Applications*, volume 484. Springer Nature, 2020.

Yiming Meng, Ruikun Zhou, Amartya Mukherjee, Maxwell Fitzsimmons, Christopher Song, and Jun Liu. Physics-informed neural network policy iteration: Algorithms, convergence, and verification. In *Forty-first International Conference on Machine Learning (ICML)*, pages 35378–35403. PMLR, 2024a.

Yiming Meng, Ruikun Zhou, Melkior Ornik, and Jun Liu. Koopman-based learning of infinitesimal generators without operator logarithm. *arXiv preprint arXiv:2403.15688*, 2024b.

Yiming Meng, Ruikun Zhou, Melkior Ornik, and Jun Liu. Resolvent-type data-driven learning of generators for unknown continuous-time dynamical systems. *arXiv preprint arXiv:2411.00923*, 2024c.

Yiming Meng, Ruikun Zhou, and Jun Liu. Learning regions of attraction in unknown dynamical systems via zubov-koopman lifting: Regularities and convergence. *IEEE Transactions on Automatic Control*, pages 1–16, 2025. doi: 10.1109/TAC.2025.3560653.

George N Saridis and Chun-Sing G Lee. An approximation theory of optimal control for trainable manipulators. *IEEE Transactions on systems, Man, and Cybernetics*, 9(3):152–159, 1979.

Peter J Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of fluid mechanics*, 656:5–28, 2010.

Draguna Vrabie and Frank Lewis. Neural network approach to continuous-time direct adaptive optimal control for partially unknown nonlinear systems. *Neural Networks*, 22(3):237–246, 2009.

Matthew O Williams, Ioannis G Kevrekidis, and Clarence W Rowley. A data–driven approximation of the Koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25:1307–1346, 2015.

Zhexuan Zeng, Zuogong Yue, Alexandre Mauroy, Jorge Gonçalves, and Ye Yuan. A sampling theorem for exact identification of continuous-time nonlinear dynamical systems. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 6686–6692. IEEE, 2022.

Zhexuan Zeng, Jun Liu, and Ye Yuan. A generalized Nyquist-Shannon sampling theorem using the Koopman operator. *IEEE Transactions on Signal Processing*, 2024a.

Zhexuan Zeng, Zuogong Yue, Alexandre Mauroy, Jorge Gonçalves, and Ye Yuan. A sampling theorem for exact identification of continuous-time nonlinear dynamical systems. *IEEE Transactions on Automatic Control*, 2024b.

Zhexuan Zeng, Ruikun Zhou, Yiming Meng, and Jun Liu. Data-driven optimal control of unknown nonlinear dynamical systems using the Koopman operator. *arXiv preprint*, 2024c.

Ruikun Zhou, Maxwell Fitzsimmons, Yiming Meng, and Jun Liu. Physics-informed extreme learning machine Lyapunov functions. *IEEE Control Systems Letters*, 2024a.

Ruikun Zhou, Yiming Meng, Zhexuan Zeng, and Jun Liu. Learning koopman-based stability certificates for unknown nonlinear systems. *arXiv preprint arXiv:2412.02807*, 2024b.