공학박사 학위논문

Data-Driven Optimal Control for Linear Systems with Arbitrary Initial Policy and Application to Nonlinear Systems Using Koopman Operators

임의의 초기 정책에 대한 선형 시스템의 데이터 기반 최적 제어 및 쿠프만 연산자를 활용한 비선형 시스템에 대한 응용

2023 년 8 월

서울대학교 대학원

기계항공공학부

김 성 훈

Data-Driven Optimal Control for Linear Systems with Arbitrary Initial Policy and Application to Nonlinear Systems Using Koopman Operators

임의의 초기 정책에 대한 선형 시스템의 데이터 기반 최적 제어 및 쿠프만 연산자를 활용한 비선형 시스템에 대한 응용

지도교수 김 유 단

이 논문을 공학박사 학위논문으로 제출함

2023 년 6 월

서울대학교 대학원

기계항공공학부

김 성 훈

김성훈의 공학박사 학위논문을 인준함

2023 년 6 월

위 원 장	김 현 진	(인)
부위원장	김 유 단	(인)
위 원	박 찬 국	(인)
위 원	심 형 보	(인)
위 원	박 종 호	(인)

Data-Driven Optimal Control for Linear Systems with Arbitrary Initial Policy and Application to Nonlinear Systems Using Koopman Operators

by

Seong-hun Kim

Submitted to the Graduate School of
Seoul National University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Mechanical and Aerospace Engineering
Seoul National University

Advisor: Prof. Youdan Kim

August 2023

Data-Driven Optimal Control for Linear Systems with Arbitrary Initial Policy and Application to Nonlinear Systems Using Koopman Operators

by

Seong-hun Kim

Approved as to style and content by:

H. Jin Kim, Chair, Ph.D.

Youdan Kim, Vice-Chair, Ph.D.

Chan Gook Park, Member, Ph.D.

Hyungbo Shim, Member, Ph.D.

Jongho Park, Member, Ph.D.

Abstract

Data-Driven Optimal Control for Linear Systems with Arbitrary Initial Policy and Application to Nonlinear Systems Using Koopman Operators

Seong-hun Kim
Department of Mechanical and Aerospace Engineering
The Graduate School
Seoul National University

A model-free off-policy reinforcement learning algorithm is proposed for solving optimal control problems for dynamic systems. The algorithm is designed to converge to not only the optimal but also stabilizing policy, which is one of the most critical concerns in designing the controller for safety-critical systems such as unmanned aerial vehicles. Unlike typical approximate dynamic programming methods, an initial stabilizing policy is not required by the proposed algorithm, which is a key advantage.

In the first part of the dissertation, a data-driven surrogate Q-leaning algorithm is proposed for linear systems based on the extended Kleinman iteration that solves algebraic Riccati equation. To allow an initial unstable policy, the value function is redefined implicitly to evaluate the performance index of the unstable policy. Based on this implicit value function, an action-value function called the surrogate Q-function is proposed by augmenting virtual control dynamics in the state space to properly define values of state and control input pairs. An off-policy data-driven method called the surrogate Q-learning is then provided based on the surrogate Q-function, which enables the reuse of data

obtained from an arbitrary control sources, e.g., trained human experts or fine-tuned PID controllers. The convergence of the extended Kleinman iteration to the unique positive definite solution, which yields the optimal stabilizing policy, is proven based on matrix inertia theory since the surrogate Q-learning is equivalent to the extended Kleinman iteration.

The second part of the dissertation is devoted to an application of the proposed reinforcement learning algorithm to nonlinear systems. The Koopman operator theory is employed to linearize nonlinear systems in an infinite-dimensional space, called the Koopman lifting linearization. The controllability and observability of linearized systems are investigated by assuming that there exists a finite-dimensional invariant subspace of the Koopman operator spanned by a mapping called the lifting. The equivalence between two optimal control problems for the original nonlinear system and the linearized system is shown under several conditions on the lifting. To find the lifting satisfying all of the conditions, a diffeomorphic lifting approximation by coupling flow-based invertible deep neural network is employed. A meta-learning framework is proposed to train the network to approximate a common lifting for a group of systems, and therefore the mode-free surrogate Q-learning can be applied to uncertain systems.

Numerical simulations using illustrative nonlinear systems with known optimal controllers are used to demonstrate the feasibility of the proposed framework, along with practical considerations and implementation details.

Keywords: Reinforcement Learning, Data-Driven Control, Learning-Based Control, Automatic Control System, Optimal Control, Algebraic Riccati Equation
Student Number: 2015-20765

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Problem Statement

Data-driven control refers to a type of control that utilizes data acquired from various sources related to the system of interest. Unlike traditional feedback control methods, which utilizes the system dynamics to design a controller and the data obtained from sensors for feedback, the main difference of the data-driven control is that the data is accumulated over a period of time to achieve a specific control objective.

The data-driven control encompasses several promising approaches such as solving optimal control problems using methods like reinforcement learning (RL) or approximate dynamic programming (ADP). The RL algorithms are typically applied to the problems where the reward function for selecting a control input at each state is unknown. In contrast, data-driven optimal control methods concentrate on addressing model-free problems that are characterized by a lack of knowledge about the system dynamics. This is because the reward function in optimal control problems is typically specified by a given performance index but constructing the dynamic model of systems can be a time-consuming and expensive process.

There are two main categories of RL methods: on-policy and off-policy methods. On-policy methods require a dataset obtained by applying the control policy being learned, while off-policy methods do not. Although on-policy methods typically converge faster, off-policy methods are more data efficient because they can reuse the dataset. Furthermore, the data acquisition process of the on-policy methods can be dangerous for the safety critical systems such as the unmanned aerial vehicles (UAVs) because the control policy being learned cannot guarantee the stability of the closed-loop system. The off-policy methods, however, can utilize the datasets obtained using independent control sources such as human experts, proportional-integral-derivative (PID) controllers, adaptive controllers, etc. In this study, the model-free and off-policy optimal control problem is considered for both linear and nonlinear continuous-time systems, where the trained control policy is stable, and the dataset acquisition process is safe.

1.2 Background, Motivation, and Necessities

The optimal control problems can be formulated as dynamic programming problems, which is to find the optimal control policy that optimizes a given performance index subject to the system dynamics. Dynamic programming is a fundamental concept in optimal control theory, and its essence can be captured by the Bellman equation. The optimal value function of the state can then be defined by a function that satisfies the Bellman optimality equation for discrete-time systems or the Hamilton-Jacobi-Bellman (HJB) equation for continuous-time systems. The HJB equation is nonlinear partial differential equations, which is difficult to solve analytically especially for complex systems. Therefore, numerical techniques such as finite difference methods or approximation algorithms are often used to obtain approximate solutions to the optimal control problem, which are the ADP methods.

Solving the HJB equation with ADP methods is classified into two strategies. The first strategy is policy iteration-based ADP, which iteratively evaluates the current policy and updates the policy for the next iteration using the evaluated value function. However, the evaluation process requires the existence of the value function corresponding to the current policy, making the policy iteration method dependent on admissible policies. The second strategy is value iteration-based ADP, which directly improves the value function and does not require admissible policies. However, value iteration-based ADP methods typically involve additional assumptions on the problem and incorporate complex numerical integration processes.

In general, value iteration-based ADP methods require more computational resources compared to policy iteration-based ADP methods due to the direct

improvement of the value function using numerical integrations. Furthermore, the stability of the policy based on the value function being learned is not guaranteed, and additional techniques such as regularization or constraints are often necessary to ensure the policy's stability. On the other hand, policy iteration-based ADP methods can be considered more appropriate for UAVs because they update the policy and the value function in an iterative manner, ensuring that the policy is admissible at each iteration by assuming that the initial policy is admissible. However, this assumption may not be practical or effective for model-free problems where the system dynamics are unknown or complex. In such cases, the knowledge of the initial admissible policy can be a form of system knowledge, which may not be readily available. This poses a challenge for policy iteration-based ADP methods, because they require an initial admissible policy even for linear systems, which is a long-standing restriction that stems from the Kleinman iteration [1].

The Kleinman iteration is an algorithmic approach for solving the continuous-time algebraic Riccati equation (ARE), which corresponds to the HJB equation for linear systems. This algorithm serves as the basis for most ADP methods [2] due to the ability to converge in quadratic rates and the possibility of extension to data-driven model-free reinforcement learning methods [3, 4]. The requirement for admissibility of all policies, including the initial guess, in the Kleinman iteration, arises from the need for the value function of the policy to be positive semidefinite. This condition is essential in the evaluation process, where the iteration repeatedly assesses the policy using the Lyapunov equation. However, this assumption may be too restrictive for data-driven model-free optimal control problems, especially in situations where the system dynamics are unknown.

Therefore, alternative algorithms are needed that can bypass this limitation and allow for more flexible policy initialization.

In the context of nonlinear systems, employing Kleinman iteration requires function approximators that can effectively approximate the value functions and the policy functions in the state-feedback form. Most ADP methods utilize linear parameterized hand-crafted basis functions to approximate those unknown functions, although the hand-crafted basis functions can be regarded as a form of prior knowledge. Using deep neural networks as the function approximators offers greater flexibility due to their universal function approximation property. However, it should be noted that training neural networks typically demands a substantial amount of data and time, which may limit their suitability for model-free ADP methods in complicated applications including UAV control design.

1.3 Literature Review

1.3.1 Iterative Methods for Solving AREs

The algebraic Riccati equation, which arises in linear quadratic regulation (LQR) problems, has been extensively studied for several decades [5]. AREs can have multiple solutions that can be real, complex, symmetric, and non-symmetric. Among those, the positive semidefinite solution is of interest in general because it can be used to obtain a stable feedback gain and represent the optimal value function [6, 7]. However, the stabilizing solution is difficult to obtain analytically, and therefore many iterative algorithms have been developed to approximate the solution, for example, eigenvector-based method using Pontryagin's maximum principle [8], the Schur vector method, which is a numerically sustainable variant of the eigenvector-based method [9], and the matrix sign function-based methods [10, 11].

Kleinman proposed a Newton method to iteratively solve the AREs [1], which has received a great attention due to its quadratic convergence rate given a good initial guess making closed-loop system stable [12–15]. In the Kleinman iteration, a Lyapunov equation is solved at each iteration step, and the feedback gain matrix for the next iteration step is determined based on the solution to the Lyapunov equation. This procedure can be considered as a variation of the policy iteration method described by Howard [16], which involves performing policy evaluation steps (i.e., solving Lyapunov equations) and policy improvement steps (i.e., finding the gain matrix for the next step) iteratively. Although the convergence to the optimal stabilizing solution is theoretically guaranteed, it requires an initial stabilizing feedback gain matrix. Several automatic stabilizing

procedures have been developed to generate the initial stabilizing gain [17–19]. However, all of these methods, including the Kleinman iteration, require the complete knowledge of the system dynamics, which is not available for model-free problems.

1.3.2 Model-Free Policy Iteration Methods

To alleviate the requirements of the system dynamics, Murray et al. proposed the adaptive dynamic programming method for continuous-time nonlinear systems utilizing a dataset collected from the system [20]. Because this method iteratively approximates the solution to the Lyapunov equation corresponding to the stable policy using a set of data including the state, control input, and the time-derivative of the state, it can be considered as a data-driven approach of the Kleinman iteration. Vrabie et al. extended this method to avoid the state derivatives in the dataset using integration of the cost for a fixed time-step, called the integral reinforcement learning (IRL) [3]. For linear quadratic regulator problems, the authors demonstrated the equivalence between the IRL and the Kleinman iteration. However, the above methods still require the knowledge of control input matrices for linear systems. Jiang and Jiang proposed a data-driven method to completely remove the requirements on the knowledge of the system by solving the Lyapunov equation and updating the control input at once [4]. This method is also the off-policy method that allows for the addition of exploration noises to the control input while avoiding contamination of the true value function to be approximated. While the main goal of the aforementioned methods is to establish a model-free framework for solving AREs or HJB equations with the aid of a dataset, it should be noted

that all of these methods rely on having prior knowledge of an initial stable or admissible policy.

1.3.3 ADP Methods Without Initial Admissible Policies

Recent research has focused on developing ADP methods that do not require an initial admissible policy. This is motivated by the observation that it is impossible to evaluate an unstable policy using the Lyapunov equation in policy iteration. The fundamental idea behind these methods is to implement the value iteration developed by Bellman [21], which does not require any explicit policy to be evaluated and drops the requirement of an initial stabilizing policy. Bian and Jiang implemented the value iteration for continuous-time systems [22, 23]. This method iteratively approximates the finite-horizon value function backward in time, and the convergence to the optimal stabilizing solution is guaranteed. Lee et al. proposed a generalized policy iteration method for continuous-time linear systems by introducing the update horizon [24]. Both methods employ a positive (semi)definite matrix as an initial estimate of the value function and iteratively refine the approximation by incorporating additional integration steps. However, the inclusion of these integration steps may introduce numerical and algorithmic complexities that can affect the efficiency and accuracy of the method.

1.3.4 The Koopman Operator for Control

Recently, the Koopman operator has gained a lot of attention as an effective tool for predicting the behavior of complex nonlinear dynamic systems. This technique interprets the evolution of state variables by employing infinite-

dimensional linear operators on transformed state variables through a mapping called the *observable*, instead of relying on the trajectories of ordinary differential equations. It has been studied in various fields ranging from fluid dynamics to power systems and UAV path-following problems [25]. Developing a method for interpreting stable and unstable subspaces using zero level sets of Koopman eigenfunctions has provided a foundation for the stability analysis of linearized systems and the operator-theoretic optimal control theory [26].

Williams et al. developed data-driven extended dynamics mode decomposition (EDMD) algorithms to efficiently approximate the infinite-dimensional linear Koopman operator for high-dimensional systems [27] and systems with exogenous control inputs [28]. The performance of approximating nonlinear systems with finite-dimensional linear systems using EDMD and designing controllers based on this has been experimentally demonstrated for robotic systems [29]. Brunton et al. proposed a linear optimal control approach by including the state variables in the observables, although only a restricted class of nonlinear systems with a single isolated fixed point can be considered and the nonlinear optimality of the controller was not proven [30].

Generalizations of the Koopman operator with control inputs were proposed by several studies, where the Koopman operator is also applied to the control inputs [31, 32]. In [31], it was demonstrated that the output space of the generalized Koopman operator can be restricted to a subspace of the observable space. Rather than using bilinear predictors [33, 34], Korda and Mezić [35] emphasized the feasibility of linear predictors, where the linear control techniques such as model predictive control (MPC) can be exploited. The approach of the Koopman operator-based MPC [36–38] was validated through numerical simu-

lations [39] and hardware experiments [40]. Comprehensive reviews of Koopman operators and their applications can be found in [41, 42].

1.3.5 Learning-Based Koopman Operator Applications

As machine learning techniques continue to advance, there has been increasing interest in using deep neural networks to represent the observables and learn them together with the Koopman operator. Yeung et al. successfully simulated the responses of high-dimensional complex nonlinear systems using the learned linear systems by applying the deep neural network-based learning technique to EDMD [43]. Folkestad et al. proposed a similar idea of using deep neural networks, but instead of directly learning the observables, the authors trained the Koopman eigenfunctions and identified important observables using their spectral information, which were then used in EDMD [44]. This approach addressed the issue of non-linear dynamics being approximated by excessively high-dimensional linear systems, resulting in increased computational efficiency for calculating optimal control inputs through methods such as MPC. Krolicki et al. demonstrated that the optimal value function and optimal control inputs can be represented by the Koopman operator [45]. They showed that the optimal solutions can be obtained using Kleinman iteration based on EDMD. In addition to the studies mentioned above, there are active research efforts to apply deep neural networks and learning techniques to Koopman operator for control system design [46, 47].

1.4 Objectives and Contributions

1.4.1 Objectives

The objective of this study is to establish a theoretical foundation for the utilization of data-driven reinforcement learning techniques in optimal control problems, and to develop effective and practical algorithms for this purpose. The overarching goals of this study can be stated as follows:

- Development of a model-free reinforcement learning method for optimal control problems that does not require any knowledge of the system dynamics including an initial admissible policy
- Mathematically rigorous analysis of the convergence to stable optimal solutions and associated characteristics of the proposed reinforcement learning algorithm
- Safe acquisition and minimization of data required for optimal controller learning

In addition, the aim of this study is to propose a control design framework that can utilize data from other (similar) systems or virtual simulation data, instead of relying on actual data of the target system, which is typically required in the conventional control system design process.

1.4.2 Contributions

Model-Free Policy Iteration Without Initial Admissible Policies

To overcome the limitations of existing policy iteration methods, which cannot perform policy evaluation for unstable control inputs due to the ill-defined

value function, this study defined a value function implicitly, where the existence and uniqueness conditions are provided for linear systems. The proposed implicit value function reveals that the matrix inertia preservation property of the Lyapunov equation is the reason for the lack of convergence of the Kleinman iteration for unstable initial polices. A virtual control input dynamics is introduced to circumvent this problem, and an implicit value function for state-input pairs augmenting this virtual dynamics is defined. The surrogate Q-learning algorithm is proposed, where the control policy is evaluated using the implicit value function, and the policy improvement step is constructed based on a necessary condition for optimal control inputs. The surrogate Q-learning is inherently an off-policy method, and therefore it can exploit data from various control resources including human experts or fine-tuned experimental controllers, which renders data acquisition processes safe.

Mathematical Convergence Analysis

Using the matrix inertia theory and monotone convergence theory, it is proven that the surrogate Q-learning always converges to a stable optimal control policy, even when an unstable initial policy is used. First, the extended Kleinman iteration based on the matrix equation, which is equivalent to the surrogate Q-learning, is formalized. It is shown that if a solution to the initial Lyapunov equation exists uniquely, regardless of the stability of the initial feedback gain, it is easy to find the design parameters that make solutions to Lyapunov equations unique in all subsequent policy evaluation steps. Next, using the matrix inertia theory, it is proven that the number of eigenvalues with the positive real part of the closed-loop system matrix monotonically decreases

as the iteration progresses. This implies that the feedback control system is monotonically stabilized as the iteration progresses. Based on this observation, local stability analysis around the solution of the algebraic Riccati equation and monotone convergence theory completes the convergence proof that the feedback gain becomes completely stable within a finite number of steps and consequently converges to the optimal stable solution.

Meta-Learning Framework for Koopman Operator

A meta-learning framework is proposed that combines Koopman operator theory with the proposed surrogate Q-learning to minimize the amount of required data and reduce the learning time for deep reinforcement learning for nonlinear systems. It is shown that the nonlinear optimal control is equivalent to the linear optimal control for the Koopman lifting linearization, if it exists. For the existence of the linear optimal control, the controllability and observability of the linearized system are investigated. Based on these analyses, six sufficient conditions for the lifting have been established for the optimal control obtained from the Koopman lifting linearization to be the same as the nonlinear optimal control of the original nonlinear system. A meta-learning problem is formulated for a specific group of nonlinear systems to find a common lifting that satisfies all of the above conditions, so that any nonlinear system within the group can be represented solely by linear system matrices. Once the meta-learning process is completed offline, the proposed framework allows obtaining nonlinear optimal control using the common lifting and surrogate Q-learning with actual data, for any uncertain nonlinear system in the group.

In summary, the proposed reinforcement learning achieves theoretically guar-

anteed convergence to the optimal stable solution for completely unknown linear systems, exploiting safely acquired real data. For nonlinear systems, the common lifting obtained from the proposed meta-learning enables rapid training of the nonlinear optimal control with a minimal data.

1.5 Dissertation Outline

The organization of this dissertation is as follows:

In Chapter 1, the backgrounds, motivations, and necessitates of this study are described, and a comprehensive review of existing literatures on data-driven optimal control methods is presented. Based on these, the objectives of this study is clearly stated, and the contributions are presented.

In Chapter 2, mathematical preliminaries on the matrix inertia theory, the Fréchet derivative, and the Koopman operator are summarized. Brief introductions to linear system theory, the Kleinman iteration, and the meta-learning framework are presented.

The main algorithms and analytical contributions of this study are described in Chapters 3 and 4. In Chapter 3, the off-policy model-free surrogate Q-learning is proposed based on a virtual control input dynamics and an implicit value function of a state-action pair. The extended Kleinman iteration, which is equivalent to the surrogate Q-learning for linear systems, is formulated And, the rigorous convergence analysis along with monotonic stabilizing property is provided.

In Chapter. 4, the meta-learning framework to obtain the nonlinear optimal control is developed based on the Koopman operator theory. The equivalence between the nonlinear optimal control and the linear optimal control is revealed, and the controllability and observability are investigated. The meta-learning problem is formulated based on these observations.

In Chapter 5, the detailed implementation of the proposed meta-learning framework is presented. The feasibility and efficacy of the proposed reinforcement learning framework are demonstrated by numerical simulations for a group

of illustrative nonlinear systems that possess a common invariant subspace of the Koopman operator.

In Chapter 6, the summary of the main results of this dissertation and suggestions for future work are provided.

Chapter 2

Theoretical Backgrounds

This section introduces notations used in this study. The mathematical preliminaries to understand the theoretical proofs and the formal definitions of the problems are introduced.

2.1 Notation

Suppose that all matrices considered in this study have real entries, except where explicitly noted. The set of real matrices of dimensions $n \times m$ is denoted by $\mathbb{R}^{n\times m}$. The identity matrix in $\mathrm{R}^{n\times n}$ is denoted by I_n. Let $\mathbb{S}^n \subset \mathbb{R}^{n\times n}$, $\mathbb{S}^n_+ \subset \mathbb{S}$, and $\mathbb{S}^n_{++} \subset \mathbb{S}_+$ denote the set of real symmetric matrices, real symmetric positive semidefinite matrices, and real symmetric positive definite matrices, respectively. Let $F \succ 0$ ($F \succeq 0$) denote that F is symmetric positive (semi)definite, and $F \succ G$ ($F \succeq G$) means that $F - G \succ 0$ ($F - G \succeq 0$). Given a square matrix A, the set of eigenvalues of A is denoted by $\sigma(A)$, and the spectral radius of A is denoted by $\rho(A)$. The Frobenius norm of a matrix A is denoted by $\|A\|_F$. If a symmetric matrix F is bounded, it means $\|F\|_F \leq c$ for some $c \geq 0$. Given a vector $v \in \mathbb{R}^n$, $\|v\|$ denotes the Euclidean norm.

2.2 Mathematical Preliminaries

2.2.1 The Matrix Inertia Theorem

Definition 2.1 (Matrix inertia)**.** The *inertia* of a square matrix A, denoted by $\mathrm{In}(A)$, is defined by

$$\mathrm{In}(A) := (\pi(A), \nu(A), \delta(A)), \tag{2.1}$$

where the elements, $\pi(A)$, $\nu(A)$, and $\delta(A)$, are the number of eigenvalues of A with positive, negative, and zero real parts, respectively.

Given two square matrices A and B of the same dimensions, let the equality $\mathrm{In}(A) = \mathrm{In}(B)$ imply $\pi(A) = \pi(B)$, $\nu(A) = \nu(B)$, and $\delta(A) = \delta(B)$.

Definition 2.2 (Matrix congruence)**.** Given real symmetric matrices A and B of the same dimensions, if there exists a nonsingular matrix S such that $A = SBS^T$, then A and B are said to be *congruent.*

Theorem 2.3 (Sylvester's law of inertia)**.** *If real symmetric matrices A and B are congruent, then* $\mathrm{In}(A) = \mathrm{In}(B)$.

2.2.2 Fréchet Derivatives

The Fréchet derivative of a matrix function $f : \mathbb{R}^{m\times n} \to \mathbb{R}^{p\times q}$ at a point $K \in \mathbb{R}^{m\times n}$ in the direction $E \in \mathbb{R}^{m\times n}$ is a linear mapping, denoted by $L_f(K, E)$, that satisfies [48]

$$\lim_{\|E\|_F \to 0} \frac{\|f(K+E) - f(K) - L_f(K,E)\|_F}{\|E\|_F} = 0 \tag{2.2}$$

for all $E \in \mathbb{R}^{m\times n}$.

Given a matrix function f, if the Fréchet derivative exists at a point K, then the function f is said to be Fréchet differentiable. The following properties of the Fréchet derivative are borrowed from [48]:

Theorem 2.4 (Sum rule [48, Theorem 3.2])**.** *If $g : \mathbb{R}^{m\times n} \to \mathbb{R}^{p\times q}$ and $h : \mathbb{R}^{m\times n} \to \mathbb{R}^{p\times q}$ are Fréchet differentiable at $K \in \mathbb{R}^{m\times n}$, then so is $f = \alpha g + \beta h$ and $L_f(K, E) = \alpha L_g(K, E) + \beta L_h(K, E)$ for any scalars $\alpha \in \mathbb{R}$ and $\beta \in \mathbb{R}$.*

Theorem 2.5 (Product rule [48, Theorem 3.3])**.** *If $g : \mathbb{R}^{m\times n} \to \mathbb{R}^{p\times q}$ and $h : \mathbb{R}^{m\times n} \to \mathbb{R}^{q\times r}$ are Fréchet differentiable at $K \in \mathbb{R}^{m\times n}$, then so is $f = gh$ and $L_f(K, E) = L_g(K, E)h(K) + g(K)L_h(K, E)$.*

Theorem 2.6 (Chain rule [48, Theorem 3.4])**.** *If $g : \mathbb{R}^{m\times n} \to \mathbb{R}^{r\times s}$ and $h : \mathbb{R}^{r\times s} \to \mathbb{R}^{p\times q}$ are Fréchet differentiable at $K \in \mathbb{R}^{m\times n}$ and $g(K) \in \mathbb{R}^{r\times s}$, respectively, then so is $f = h \circ g$ and $L_f(K, E) = L_h(g(K), L_g(K, E))$.*

2.2.3 The Koopman Operator

Consider a class of nonlinear autonomous systems given by

$$\dot{x} = f(x), \tag{2.3}$$

where $f : X \to \mathbb{R}^n$ is a continuously differentiable function on a compact set $X \subset \mathbb{R}^n$. Because $f \in C^1(X)$, there exists a unique solution $x(t_0 + t)$ for any initial state $x(t_0) \in X$ and $t \geq 0$. Therefore, an operator $\xi^t : X \to X$ can be defined, which maps any initial state to the state for time t following the dynamics in (2.3), i.e., $\xi^t(x(t_0)) = x(t_0 + t)$. Note that the family $\{\xi^t\}$ associated with the system (2.3) is a one-parameter semigroup [49, Definition 13.34], because $\xi^0 = I$, where I denotes the identity operator on X, and $\xi^{t+s} = \xi^t\xi^s = \xi^s\xi^t$ for all $t, s \geq 0$.

Consider a Banach space $\mathcal{F}$ of functions $\phi\colon X \to \mathbb{R}$, and further assume that $\mathcal{F} \subset C^1(X, \mathbb{R})$. Any function $\phi \in \mathcal{F}$ is referred to as an *observable* because it represents the measurement obtained from a sensor [25]. It is worth noting that the definition of observables in Koopman operator theory should be distinguished from the observability in control theory. The Koopman operator associated with (2.3) is defined on $\mathcal{F}$ as follows.

Definition 2.7 (Koopman operators [50])**.** The family of Koopman operators $\mathcal{K}^t : \mathcal{F} \to \mathcal{F}$ associated with the family of maps ξ^t, $t \geq 0$, is defined by

$$\mathcal{K}^t \phi = \phi \circ \xi^t, \quad \forall \phi \in \mathcal{F}. \tag{2.4}$$

Note that $\left\{\mathcal{K}^t\right\}_{t \geq 0}$ is also a one-parameter semigroup, called the *Koopman semigroup of operators*, because $\mathcal{K}^0 \phi = \phi \circ \xi^0 = \phi$ by construction and

$$\mathcal{K}^{t+s} \phi = \phi \circ \xi^{t+s} = \phi \circ \left(\xi^t \circ \xi^s\right) = \mathcal{K}^t \phi \circ \xi^s = \mathcal{K}^t \mathcal{K}^s \phi, \tag{2.5}$$

for all $t, s \geq 0$. The Koopman operator is linear, i.e., $\mathcal{K}^t(\alpha_1 g_1 + \alpha_2 g_2) = \alpha_1 \mathcal{K}^t g_1 + \alpha_2 \mathcal{K}^t g_2$ for all $g_1, g_2 \in \mathcal{F}$, $\alpha_1, \alpha_2 \in \mathbb{R}$, and $t \geq 0$.

Due to the continuity of the solution of (2.3) and of the observables ϕ, the Koopman semigroup of operators has an additional property of strong continuity, which can be stated as

$$\lim_{t \to 0} \left\| \mathcal{K}^t \phi - \phi \right\| = 0, \quad \forall \phi \in \mathcal{F}. \tag{2.6}$$

Then, associate $\left\{\mathcal{K}^t\right\}_{t \geq 0}$ with the operator $\mathcal{A}_\varepsilon$ by

$$\mathcal{A}^\varepsilon \phi = \frac{1}{\varepsilon}(\mathcal{K}^\varepsilon \phi - \phi), \quad \forall \phi \in \mathcal{F}, \forall \varepsilon > 0, \tag{2.7}$$

and define an operator $\mathcal{A}$ by

$$\mathcal{A}\phi = \lim_{\varepsilon \to 0} \mathcal{A}^\varepsilon \phi \tag{2.8}$$

for all $\phi \in \mathcal{D}(\mathcal{A})$, i.e., for all ϕ where the limit in (2.8) exists in the norm topology of $\mathcal{F}$. The operator $\mathcal{A}$, which is essentially $\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{K}^0$, is called the *infinitesimal generator* of the semigroup $\left\{\mathcal{K}^t\right\}_{t\geq 0}$ [49, Theorem 13.35]. Moreover, from $\mathcal{K}^t\phi(x_0) = \phi(\xi^t(x_0)) = \phi(x(t))$, where $x(0) = x_0$, and from the assumption that $\phi \in C^1(X, \mathbb{R})$, Theorem 13.35 in [49] implies that

$$\frac{\mathrm{d}}{\mathrm{d}t}\phi(x(t)) = \mathcal{A}\phi(x(t)) = \boldsymbol{\nabla}\phi(x(t))^T f(x(t)) \tag{2.9}$$

for all $t \geq 0$. It is clear that $\mathcal{D}(\mathcal{A})$ is a subspace of $\mathcal{F}$ and that $\mathcal{A}$ is thus a linear operator in $\mathcal{F}$.

2.3 Linear System Theory

Consider a linear system given by

$$\dot{x}(t) = Ax(t) + Bu(t), \tag{2.10}$$

where $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$ denote the system matrices, $x(t) \in \mathbb{R}^n$ is the state vector, and $u(t) \in \mathbb{R}^m$ is the control input vector.

2.3.1 Controllability and Observability

Definition 2.8 (Controllability)**.** The linear system (2.10) or the matrix pair (A, B) is called *controllable* if an input function $u : [0, \infty) \to \mathbb{R}^m$ exists to transfer the initial state $x(0) = x_0 \in \mathbb{R}^n$ to any final state $x_1 \in \mathbb{R}^n$ within a finite time. Conversely, if there does not exist such an input function, the system (2.10) or the pair (A, B) is called *uncontrollable*.

Consider an output equation given by

$$y(t) = Cx(t), \tag{2.11}$$

where $C \in \mathbb{R}^{q\times n}$ denotes the observer matrix, and $y(t) \in \mathbb{R}^q$ is the output vector.

Definition 2.9 (Observability)**.** The linear system (2.10) or the matrix pair (A, C) is called *observable* with respect to the output $y(t)$ in (2.11) if there exists a finite time interval $[0, t_1]$ for any unknown initial state $x(0) \in \mathbb{R}^n$, such that the knowledge of input $u(t)$ and output $y(t)$ during this interval is enough to uniquely determine $x(0)$. If such a time interval does not exist, the system (2.10) or the pair (A, C) is said to be *unobservable* with respect to the output.

The following theorem states the duality between the controllability and the observability.

Theorem 2.10 ([51, Theorem 6.5])**.** *The pair (A, B) is controllable if and only if the pair (A^T, B^T) is observable.*

In the following, useful properties of preserving controllability and observability are presented.

Proposition 2.11 ([52, Corollary 4.1.3])**.** *If $C \succ 0$, then (A, B) is controllable if and only if (A, BCB^T) is controllable.*

Lemma 2.12 ([53, Lemma 2.1])**.** *For any K, if (A, B) is controllable, then $(A - BK, B)$ is controllable.*

Theorem 2.13 ([53, Theorem 3.6])**.** *If $Q \succeq 0$ and (A, Q) is observable, then for all $R \succ 0$ and all B, K, the pair $(A - BK, Q + K^T RK)$ is observable.*

Proof. From Proposition 2.11 and the duality from Theorem 2.10, the pair (A, Q) is observable if and only if $(A, \sqrt{Q})$ is observable. Then, from [53, Theorem 3.6. ii.], the pair $(A - BK, \sqrt{Q + K^T RK})$ is observable, which completes the proof by applying Proposition 2.11 and Theorem 2.10 once again. □

2.3.2 Algebraic Riccati Equations

The ARE arises in the infinite-horizon optimal control problem for linear systems in (2.10) with a performance index given by

$$J(x_0; u) = \int_0^\infty \left(x(t)^T Q x(t) + u(t)^T R u(t) \right) \mathrm{d}t \,, \tag{2.12}$$

where $Q \in \mathbb{S}^n_+$ and $R \in \mathbb{S}^m_{++}$ denote the weighting matrices, and the $x(t)$ is the state trajectory of the system in (2.10) with $x(0) = x_0$ and the control

input function u. The infinite-horizon optimal control problem minimizing the performance index J in (2.12) is given by

$$\inf_u J(x;u), \quad \forall x \in \mathbb{R}^n, \tag{2.13}$$

and it corresponds to the ARE, which is a matrix equation of a variable $P \in \mathbb{R}^{n\times n}$, given by

$$\mathcal{R}(P) := PA + A^T P + Q - PBR^{-1}B^T P = 0, \tag{2.14}$$

where $\mathcal{R}$ denotes the Riccati operator. The ARE possesses a unique positive definite solution in the class of positive semidefinite matrices under controllability and observability conditions which are typically assumed in control problems.

Theorem 2.14 ([53, Theorem 4.1])**.** *If (A, B) is controllable and (A, Q) is observable, then the ARE in* (2.14) *has a unique solution $P^* \in \mathbb{S}^n_{++}$ in the class of $\mathbb{S}^n_+$, and $A - BR^{-1}B^T P^*$ is Hurwitz.*

Under the hypotheses of Theorem 2.14, the optimal control input solving (2.13) can be represented by [54, Theorem 6.1]

$$u^* = -R^{-1}B^T P^* x =: -K^* x \tag{2.15}$$

for $K^* \in \mathbb{R}^{m\times n}$, and the corresponding closed-loop-loop system is stable, i.e., $A - BK^*$ is Hurwitz, by Theorem 2.14. Moreover, the optimal performance index $J^*(x)$ defined by

$$J^*(x) := J(x;u^*) = \min_u J(x;u) \tag{2.16}$$

can be represented by a quadratic form of the state x as

$$J^*(x) = x^T P^* x. \tag{2.17}$$

2.3.3 Lyapunov Equations

The Lyapunov equation is a matrix equation of a variable X defined by

$$XA + A^T X + M = 0, \tag{2.18}$$

where A and M are real square matrices of the same dimensions. In what follows, two useful theorems are introduced corresponding to the existence and the inertia property of the solution X to the Lyapunov equation in (2.18).

Theorem 2.15 ([5, Corrollary 8.2.1])**.** *Given real square matrices A and M, the Lyapunov equation in* (2.18) *has a unique solution $X = X^T$ if and only if $\sigma(A) \cap \sigma(-A) = \varnothing$.*

Theorem 2.16 ([55, Theorem 4.6])**.** *Given real square matrices A and $M \succeq 0$, suppose that (A, M) is observable. If $X = X^T$ is a solution to the Lyapunov equation in* (2.18)*, then $\delta(A) = 0$ and $\mathrm{In}(A) = \mathrm{In}(-X)$.*

Theorem 2.15 implies that the Lyapunov equation in (2.18) can have either many solutions or no solution if $\sigma(A) \cap \sigma(-A) \neq \varnothing$. Further assuming the observability of the pair (A, M), the following lemma characterizes that the solution set can only be empty.

Lemma 2.17. *Given real square matrices A and $M \succeq 0$, suppose that (A, M) is observable. Then, a solution $X = X^T$ to the Lyapunov equation in* (2.18) *exists if and only if $\sigma(A) \cap \sigma(-A) = \varnothing$.*

Proof. The "if" part is a direct result of Theorem 2.15. To prove the "only if" part, suppose that $\sigma(A) \cap \sigma(-A) \neq \varnothing$. Let x and y be the left eigenvectors of A and $-A$ associated with the common eigenvalue λ of A and $-A$, respectively, as

$x^T A = \lambda x^T$ and $y^T A = -\lambda y^T$. Put $X_1 = xy^T + yx^T$. To obtain a contradiction, suppose that there is a solution X_0 to (2.18), which is nonsingular as (A, M) is observable from Theorem 2.16. Then, $X = X_0 + \alpha X_1$ for any $\alpha \in \mathbb{C}$ is a solution to (2.18). It follows that

$$\delta(-A) = \delta(X_0) = \delta(X_0 + \alpha X_1) = 0 \tag{2.19}$$

from Theorem 2.16. Since there always exists $\alpha \in \mathbb{C}$ such that $\det(X_0 + \alpha X_1) = 0$, or equivalently $\delta(X_0 + \alpha X_1) > 0$. This contradicts (2.19), which implies that there is no solution X_0 to (2.18). $\square$

2.4 The Kleinman Iteration

The Kleinman iteration solves a series of Lyapunov equations to obtain the optimal stabilizing solution $P^* \succ 0$ asymptotically, where P^* is the solution to $\mathcal{R}(P^*) = 0$ in (2.14). The convergence of the Kleinman iteration is ensured by the following theorem [1].

Theorem 2.18 (The Kleinman iteration)**.** *Let K_0 be any stabilizing feedback gain matrix, and let P_k be the symmetric positive definite solution to the Lyapunov equation given by*

$$P_k(A - BK_k) + (A - BK_k)^T P_k + Q + K_k^T R K_k = 0, \tag{2.20}$$

where K_k is defined recursively by

$$K_{k+1} = R^{-1} B^T P_k, \tag{2.21}$$

for all $k = 0, 1, \ldots$. Then, the following properties hold.

1. *$A - BK_k$ is Hurwitz,*
2. *$P_k \succeq P_{k+1} \succeq P^*$,*
3. *$\lim_{k\to\infty} K_k = K^*$, and $\lim_{k\to\infty} P_k = P^*$.*

Theorem 2.18 requires that the initial feedback gain K_0 stabilizes the closed-loop system for the Kleinman iteration, i.e., $A - BK_0$ must be Hurwitz. The subsequent lemma demonstrates that the Kleinman iteration preserves the inertia of closed-loop system matrices. This elucidates why the Kleinman iteration fails to converge to the optimal stabilizing solution when initiated with an unstable feedback gain.

Lemma 2.19. *In the Kleinman iteration, given an initial feedback gain K_0, suppose that there exists a unique symmetric solution P_0. Then,*

$$\mathrm{In}(A - BK_0) = \mathrm{In}(A - BK_k) \tag{2.22}$$

for all $k = 1, 2, \ldots$.

Proof. Let $A_k = A - BK_k$. The proof is by induction on k. It is shown that if A_k is unstable and there exists a unique symmetric P_k, then A_{k+1} is also unstable and there exists a unique symmetric P_{k+1}. Let $Q_k := Q + K_k^T R K_k$. Because P_k satisfies

$$P_k A_k + A_k^T P_k + Q_k = 0, \tag{2.23}$$

and the pair (A_k, Q_k) is observable from Theorem 2.13, it follows that

$$\mathrm{In}(A_k) = \mathrm{In}(-P_k) \tag{2.24}$$

from Theorem 2.16. Rewrite (2.23) using (2.21) as

$$P_k A_{k+1} + A_{k+1}^T P_k + \tilde{Q}_k = 0, \tag{2.25}$$

where

$$\tilde{Q}_k := Q + (K_{k+1} - K_k)^T R (K_{k+1} - K_k) + K_{k+1}^T R K_{k+1}. \tag{2.26}$$

Because the pair $(A_{k+1}, \tilde{Q}_k)$ is also observable from Theorem 2.13, it follows from Theorem 2.16 that

$$\mathrm{In}(A_{k+1}) = \mathrm{In}(-P_k). \tag{2.27}$$

From (2.24) and (2.27), $\mathrm{In}(A_k) = \mathrm{In}(A_{k+1})$, which implies that A_{k+1} is unstable. Since P_k is a symmetric solution to (2.25), it follows that

$$\sigma(A_{k+1}) \cap \sigma(A_{k+1}) = \varnothing \tag{2.28}$$

by Lemma 2.17. Then, it can be concluded that P_{k+1} also exists. This completes the proof by induction. $\square$

The inherent property of inertia preservation in the Kleinman iteration by Lemma 2.19 reveals that if $\pi(A - BK_0) > 0$, indicating the initial closed-loop system matrix is unstable, then all subsequent closed-loop system matrices satisfy $\pi(A - BK_k) > 0$ for $k = 1, 2, \ldots,$ meaning they are all unstable.

2.5 Meta-Learning

Meta-learning is the process of acquiring knowledge or experience about how to learn, aimed at improving the efficiency and efficacy of learning new tasks. It consists of two key learning components: a base learner that is utilized for each specific task and a meta learner that enhances the base learner's capabilities to improve its efficiency and effectiveness for learning new tasks. The goal of the meta learner may differ from that of the base learner due to varying objectives including rapid adaptation to a new task or the reduction of computational burden [56].

2.5.1 Optimization Problem Formulations

The parameter of the meta learner, denoted by w, can be any of several components of the base learner, including but not limited to an optimization solver, a loss function, initial parameters, or pre-trained networks employed for input embedding. The meta learner optimizes w over a task distribution $p(\mathcal{T})$ based on a loss function $\mathcal{L}$, where a task $\mathcal{T}_i \sim p(\mathcal{T})$ is composed of a loss function $\mathcal{L}_i$ and a dataset $\mathcal{D}_i := (\mathcal{D}_i^{\text{train}}, \mathcal{D}_i^{\text{val}})$ with the training and validation datasets. Then, the meta-learning algorithm can be interpreted as the following bi-level optimization problem:

$$w^* = \arg\min_{w} \mathbb{E}_{\mathcal{T}_i \sim p(\mathcal{T})} \Big[\mathcal{L}\Big(\mathcal{D}_i^{\text{val}}; w_i^*(w), w \Big) \Big], \tag{2.29}$$

$$w_i^*(w) = \arg\min_{w_i} \mathcal{L}_i\big(D_i^{\text{train}}; w_i, w \big), \tag{2.30}$$

where the meta and base learners correspond to the optimization problems in (2.29) and (2.30), respectively. Here, w_i denotes the parameter of the base learner for $\mathcal{T}_i$. It is important to emphasize that the parameters w and w_i

used to represent the meta-learner and base-learner parameters are conceptual notations and do not necessarily represent specific vectors or matrices. They serve as placeholders to denote the trainable parameters in the meta-learning framework without implying a specific mathematical form. More comprehensive understanding of meta-learning can be found in Hospedales et al. [57].

2.5.2 Closed-Form Base Learners

Selecting appropriate base learners is crucial for successful meta-learning. In [58], the feasibility of utilizing fast solvers with closed-form solutions as the base learner was investigated to enhance the efficiency of adapting to new learning problems. In this framework, a linear predictor $F_i\phi(x;w) \in \mathbb{R}^{n_o}$ and a least-square loss function

$$\mathcal{L}_i\big(\mathcal{D}_i^{\text{train}}; w_i, w\big) = \sum_{j=1}^{N_i} \|F_i\phi(x_j;w) - y_j\|^2 \tag{2.31}$$

with a dataset $\mathcal{D}_i^{\text{train}} = \{(x_j, y_j)\}_{j=1}^{N_i}$ is assigned to the base learner for each task $\mathcal{T}_i$, where $y_j \in \mathbb{R}^{n_o}$ is the output vector, $\phi(x_j;w) \in \mathbb{R}^{n_e}$ is an embedding function of the input x_j parameterized by the meta-learner parameter w, and the matrix $F_i \in \mathbb{R}^{n_o \times n_e}$ denotes the base-learner parameter w_i. Then, the closed-form solution $F_i^*(w)$ for the base-learner problem in (2.30) is given by

$$F_i^*(w) = \begin{bmatrix} y_1 & \cdots & y_{N_i} \end{bmatrix} \begin{bmatrix} \phi(x_1;w) & \cdots & \phi(x_{N_i};w) \end{bmatrix}^{\dagger}, \tag{2.32}$$

where $(\cdot)^{\dagger}$ denotes the Moore-Penrose inverse. With the aid of the closed-form solution in (2.32), the gradient of $F_i^*(w)$ with respect to w can be readily computed using standard automatic differentiation packages, such as PyTorch Autograd. This gradient can be utilized in the backpropagation procedure to solve (2.29).

Chapter 3

Data-Driven Optimal Control for Unknown Linear Systems

This section describes a policy iteration scheme for unknown linear systems, which allows an unstable initial policy. An *implicit value function* is proposed to replace the value function for an unstable policy, and the uniqueness and existence of the implicit value function are investigated under a mild condition of the closed-loop system.

3.1 Implicit Value Functions

Let us consider a linear autonomous system

$$\dot{x}(t) = A_c x(t), \tag{3.1}$$

where $x(t) \in \mathbb{R}^n$ is the state vector, and $A_c \in \mathbb{R}^{n \times n}$ is the system matrix. A closed-loop system matrix for (2.10) with a linear feedback control $u = -Kx$ is given by $A - BK$ which can be regarded as the system matrix A_c in (3.1).

The performance index $J(x_0; u)$ in (2.12) is typically considered as a value of the initial state x_0 using the control input function u. In the ADP literature, $J(x_0; u)$ is typically defined as the value function which is a functional of a state $x_0 \in \mathbb{R}^n$ and an input function $u : [0, \infty) \to \mathbb{R}^m$. If $u = -Kx$, the value

function can be rewritten as

$$J_c(x_0) \coloneqq J(x_0; -Kx) = \int_0^\infty x(t)^T Q_c x(t)\,\mathrm{d}t\,, \tag{3.2}$$

where $Q_c = Q + K^T RK$. In this definition, the value function $J_c(x_0)$ can be infinite if the closed-loop system matrix $A_c = A - BK$ is not Hurwitz, which renders the requirement of the admissible control input for the value function being well-defined [59]. Therefore, it is necessary to define a different value function well-defined even for unstable A_c's.

Definition 3.1 (Implicit value functions)**.** Suppose that there exists $P \in \mathbb{S}^n$ such that a quadratic function of the state $x_0 \in \mathbb{R}^n$ given by $V(x_0) = x_0^T P x_0$ satisfies

$$V(x_0) - V(x(t)) = \int_0^t x(\tau)^T Q_c x(\tau)\,\mathrm{d}\tau \tag{3.3}$$

for all $x_0 \in \mathbb{R}^n$ and for all $t \geq 0$, where $Q_c \in \mathbb{S}^n$ and $x(\cdot)$ is the state trajectory of (3.1) with $x(0) = x_0$. Then, the function V is called the *implicit value function* for the system (3.1).

If A_c is Hurwitz, then the implicit value function $V(x_0)$ is consistent with the value function $J_c(x_0)$ in (3.2). In particular, if A_c is Hurwitz, the linear system in (3.1) is exponentially stable, and thus $\lim_{t\to\infty} x(t) = 0$, which is followed by $\lim_{t\to\infty} V(x(t)) = 0$. Moreover, it is well-known that there exists $P_J \in \mathbb{S}^n_{++}$ such that $J_c(x_0) = x_0^T P_J x_0$ for all $x_0 \in \mathbb{R}^m$ if and only if A_c is Hurwitz [1]. Since the implicit value function $V(x_0)$ satisfies (3.3) for all $t \geq 0$, it follows that

$$\begin{aligned} V(x_0) &= \lim_{t\to\infty} \left(V(x(t)) + \int_0^t x(\tau)^T Q_c x(\tau)\,\mathrm{d}\tau \right) \\ &= \int_0^\infty x(\tau)^T Q_c x(\tau)\,\mathrm{d}\tau = x_0^T P_J x_0 \end{aligned} \tag{3.4}$$

for all $x_0 \in \mathbb{R}^n$, which implies $P = P_J$.

To examine the existence of the implicit value function for an arbitrary A_c that may not necessarily be Hurwitz, consider the Lyapunov equation of $P \in \mathbb{S}^n$ for the system in (3.1), given by

$$\mathcal{L}(P) := PA_c + A_c^T P + Q_c = 0, \tag{3.5}$$

where $\mathcal{L}$ is called the Lyapunov operator.

Proposition 3.2 (Existence)**.** *For the system* (3.1)*, if* $\sigma(A_c) \cap \sigma(-A_c) = \varnothing$*, there exists an implicit value function.*

Proof. From $\sigma(A_c) \cap \sigma(-A_c) = \varnothing$, Theorem 2.15 implies that there exists a matrix $P \in \mathbb{S}^n$ which is a solution to $\mathcal{L}(P) = 0$ in (3.5). Consider the state transition matrix of (3.1) defined by $\Phi(\tau) = e^{A_c \tau}$ for any $\tau \geq 0$, which has the following properties: $x(\tau) = \Phi(\tau)x(0)$ and

$$\Phi(\tau)A_c = A_c\Phi(\tau) = \frac{\mathrm{d}}{\mathrm{d}\tau}\Phi(\tau). \tag{3.6}$$

It follows from (3.5) that

$$\Phi(\tau)^T \mathcal{L}(P)\Phi(\tau) = \frac{\mathrm{d}}{\mathrm{d}\tau}\big(\Phi(\tau)^T P\Phi(\tau)\big) + \Phi(\tau)^T Q_c \Phi(\tau) = 0. \tag{3.7}$$

Integrating (3.7) yields

$$\Phi(0)^T P\Phi(0) - \Phi(t)^T P\Phi(t) = \int_0^t \Phi(\tau)^T Q_c \Phi(\tau)\,\mathrm{d}\tau \tag{3.8}$$

for any $t \geq 0$. Pre-multiplying $x(0)^T$ and post-multiplying $x(0)$ to (3.8) yields (3.3), which completes the proof. □

It is important to note that the implicit equation presented in (3.3) needs to hold true for all time instances $t \geq 0$, in contrast to the typical Bellman

equation which is applicable for a fixed time step $t > 0$. This difference implies that the implicit value function can be uniquely defined for unstable control input at least for linear systems.

Proposition 3.3 (Uniqueness)**.** *Given the linear system in* (3.1)*, suppose that* $\sigma(A_c) \cap \sigma(-A_c) = \varnothing$*. Then, there exists a unique implicit value function in* (3.3) *with the unique solution* P *to* $\mathcal{L}(P) = 0$ *in* (3.5)*.*

Proof. From Proposition 3.2 and Theorem 2.15, there is an implicit value function $\bar{V}(x_0)$ with $\bar{P}$ which is a unique solution to $\mathcal{L}(P) = 0$. Suppose that there is another implicit value function $V(x_0)$ with $P \neq \bar{P} \in \mathbb{S}^n$. Consider the same state transition matrix $\Phi(\tau)$ defined in the proof of Proposition 3.2. Because the implicit value function $V(x_0)$ satisfies (3.3) for all x_0, it follows that

$$P - \Phi(t)^T P \Phi(t) = \int_0^t \Phi(\tau)^T Q_c \Phi(\tau) \, \mathrm{d}\tau \,. \tag{3.9}$$

Using the property of $\Phi(\tau)$ in (3.6), pre-multiplying A_c^T and post-multiplying A_c to (3.9) yield

$$\begin{aligned}
&\big(P - \Phi(t)^T P \Phi(t)\big) A_c + A_c^T \big(P - \Phi(t)^T P \Phi(t)\big) \\
&= \int_0^t \big(\Phi(\tau)^T Q_c \Phi(\tau) A_c + A_c^T \Phi(\tau)^T Q_c \Phi(\tau)\big) \, \mathrm{d}\tau \\
&= \int_0^t \frac{\mathrm{d}}{\mathrm{d}\tau} \big(\Phi(\tau)^T Q_c \Phi(\tau)\big) \, \mathrm{d}\tau \\
&= \Phi(t)^T Q_c \Phi(t) - \Phi(0)^T Q_c \Phi(0).
\end{aligned} \tag{3.10}$$

Because the inverse of a state transition matrix exists for all $t \geq 0$, rearranging (3.10) using (3.5) yields

$$\mathcal{L}(P) \Phi(t)^{-1} - \Phi(t)^T \mathcal{L}(P) = 0 \tag{3.11}$$

for any $t \geq 0$, which can be regarded as a Sylvester equation of the variable $\mathcal{L}(P)$. Because the Lyapunov equation $\mathcal{L}(P) = 0$ has the unique solution $P \neq \bar{P}$, the Sylvester equation (3.11) has a non-trivial solution $\mathcal{L}(\bar{P}) \neq 0$, which implies that the two matrices $\Phi(t)^{-1}$ and $\Phi(t)^T$ share one or more eigenvalues [5, Theorme 8.2.1] for all $t \geq 0$.

By the definition of the state transition matrix $\Phi(t)$, it follows that

$$\sigma\big(\Phi(t)^T\big) = \big\{e^{\lambda t} \mid \lambda \in \sigma(A_c)\big\}, \tag{3.12}$$

$$\sigma\big(\Phi(t)^{-1}\big) = \big\{e^{\lambda t} \mid \lambda \in \sigma(-A_c)\big\}, \tag{3.13}$$

for any $t > 0$. Therefore, if there exists $\mu \in \sigma\big(\Phi(t)^T\big) \cap \sigma\big(\Phi(t)^{-1}\big)$, then $\lambda = t^{-1}\log(\mu) \in \sigma(A_c) \cap \sigma(-A_c)$, which contradicts the assumption that $\sigma(A_c) \cap \sigma(-A_c) = \varnothing$. $\square$

From Propositions 3.2 and 3.3, it can be confirmed that the solution to Lyapunov equation in the Kleinman iteration in (2.23) corresponds to the implicit value function of the feedback gain K_k. Because the Kleinman iteration preserves the inertia of the closed-loop system matrix by Lemma 2.19 even when the implicit value function is well-defined, it can be inferred that the issue lies in the policy improvement step.

3.2 The Surrogate Q-Learning

This section provides the main algorithm, called the *surrogate Q-learning*, based on the concept of the implicit value function introduced in Section 3.1. The proposed algorithm is a reinforcement learning algorithm that solves the infinite-horizon optimal control problem in (2.13) for the linear system given by (2.10). It is also a data-driven model-free algorithm that does not require any knowledge of the system, namely the system matrices A and B, and an initial admissible policy. Moreover, the algorithm is off-policy, allowing the utilization of any behavior policies to obtain the necessary dataset for the proposed data-driven method.

The problem of obtaining an optimal control input using data, without knowledge of the system matrices A and B, has been extensively studied using policy iteration-based ADP techniques [2, 60]. However, these techniques are fundamentally based on the Kleinman iteration [1], which requires knowledge of an initial admissible control input [20]. The proposed algorithm offers a significant advantage, i.e., not requiring any information on the initial admissible control inputs, despite being a policy iteration method by nature.

First, a continuous-time action-value function is defined using the concept of implicit value function. Then, a policy iteration algorithm is proposed where the policies are improved based on the extremum of the implicit value function.

3.2.1 Surrogate Q-Functions for Continuous-Time Systems

For discrete-time systems, the Q-function plays a vital role in the off-policy temporal difference RL, often called the Q-learning [61]. The Q-function depends on a policy to be trained, called the training policy, and takes the state

and control input variables at the current time-step and evaluates the expected cumulative cost following the training policy after the current time-step. For continuous-time systems, however, it is ambiguous how to separate the input in the current time step from the subsequent time steps.

To avoid this ambiguity, an alternative concept is introduced to evaluate the value of the current state and control input. For the linear system in (2.10), suppose that a policy $u = -Kx$ is given, where the closed-loop system matrix $A - BK$ is allowed to be unstable. For each state $x_0 \in \mathbb{R}^n$ and each control input $u_0 \in \mathbb{R}^m$, consider a *virtual* linear autonomous system with $\xi(0) = x_0$ and $\mu(0) = u_0$. Now, the control input augmented dynamics is given by

$$\begin{bmatrix} \dot{\xi}(t) \\ \dot{\mu}(t) \end{bmatrix} = \begin{bmatrix} A & B \\ -KA - sK & -KB - sI_m \end{bmatrix} \begin{bmatrix} \xi(t) \\ \mu(t) \end{bmatrix}, \tag{3.14}$$

where the scalar $s > 0$ is the design parameter which will be discussed later.

An implicit value function, called the *surrogate Q-function*, is introduced for (3.14) corresponding to the infinite-time horizon optimal control problem in (2.13). The surrogate Q-function is defined by a function $\mathcal{Q}(x_0, u_0)$ that satisfies

$$\mathcal{Q}(x_0, u_0) - \mathcal{Q}(\xi(t), \mu(t)) = \int_0^t \left(\xi(\tau)^T Q \xi(\tau) + \mu(\tau)^T R \mu(\tau)\right) \mathrm{d}\tau \tag{3.15}$$

for all $t \geq 0$, where $Q \in \mathbb{S}_+^n$ and $R \in \mathbb{S}_{++}^m$ are the same as in (2.12). The following proposition presents some conditions for the existence and the uniqueness of the implicitly defined surrogate Q-function.

Proposition 3.4. *Suppose that the sets $\sigma(A - BK)$, $\sigma(-A + BK)$, and $\{s\}$ are disjoint, and that (A, Q) is observable. Then, there exists a unique surrogate*

Q-function satisfying (3.15)*, given by*

$$\mathcal{Q}(x_0, u_0) = \begin{bmatrix} x_0^T & u_0^T \end{bmatrix} M \begin{bmatrix} x_0 \\ u_0 \end{bmatrix} \tag{3.16}$$

for some $M \in \mathbb{S}^{n+m}$.

Proof. Define a matrix $U \in \mathbb{R}^{(n+m)\times(n+m)}$ which is invertible for any $K \in \mathbb{R}^{m\times n}$ as

$$U = \begin{bmatrix} I_n & 0 \\ -K & I_m \end{bmatrix}, \quad U^{-1} = \begin{bmatrix} I_n & 0 \\ K & I_m \end{bmatrix}. \tag{3.17}$$

Then, the corresponding equivalence transform [51, Definition 4.1] yields

$$\bar{A} := U^{-1} A^\circ U = \begin{bmatrix} A - BK & B \\ 0 & -sI_m \end{bmatrix}, \quad \bar{Q} := Q^\circ U = \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix}, \tag{3.18}$$

where

$$A^\circ = \begin{bmatrix} A & B \\ -KA - sK & -KB - sI_m \end{bmatrix}, \quad Q^\circ = \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix}. \tag{3.19}$$

Because $\sigma(A^\circ) = \sigma(\bar{A}) = \sigma(A - BK) \cup \{-s\}$, the hypothesis that the sets $\sigma(A-BK)$, $\sigma(-A+BK)$, and $\{s\}$ are disjoint implies that $\sigma(A^\circ) \cap \sigma(-A^\circ) = \varnothing$. Consequently, applying Propositions 3.2 and 3.3 for the autonomous system in (3.14) and the implicit equation in (3.15) completes the proof. □

The surrogate Q-function presented in this study is designed for continuous-time systems. Although it shares some similarities with the action-value function or Q-function for discrete-time systems, it is not exactly the same. Therefore, the surrogate Q-function is introduced as an approximation of the Q-function for continuous-time systems.

The relationship between the surrogate Q-function and the (implicit) value function of continuous-time systems is distinguished from that of discrete-time systems. In particular, if a linear feedback $u = -Kx$ is evaluated using the implicit value function $V(x_0; u)$ in (3.3), it is the actual control input applied to the system dynamics in (2.10). However, the matrix K evaluated in (3.15) serves as a target control input $\mu_T(t) \coloneqq -K\xi(t)$. It can be seen from (3.14) as

$$\frac{\mathrm{d}}{\mathrm{d}t}(\mu(t) - \mu_T(t)) = -s(\mu(t) - \mu_T(t)), \tag{3.20}$$

which implies that the error $\mu(t) - \mu_T(t)$ exponentially converges to zero from $u_0 + Kx_0$. Therefore, $\mathcal{Q}(x_0, u_0)$ evaluates the pair of a state x_0 and an instant control input u_0 using the infinite-horizon integral of the quadratic cost by *assuming* that the rest of the state trajectory $x(t)$ is obtained by a control input $u(t)$ that exponentially converges to $-Kx(t)$ with $x(0) = x_0$ and $u(0) = u_0$.

Given a policy μ, the performance index $J(x_0; \mu)$ of the optimal control problem in (2.13) can be defined only if μ is admissible. On the other hand, the surrogate Q-function corresponding to J can only be defined for autonomous systems, regardless of whether the system is stable or not. The following proposition presents a relationship between the surrogate Q-function and the performance index.

Proposition 3.5. *Given a policy $u = -Kx$, the corresponding surrogate Q-function $\mathcal{Q}$ in* (3.15) *and a performance index $J(x_0; u + \nu)$, where $\nu(t) = e^{-st}(u_0 + Kx_0)$, satisfy*

$$\mathcal{Q}(x_0, u_0) = J(x_0; u + \nu) \tag{3.21}$$

for any $(x_0, u_0) \in \mathbb{R}^n \times \mathbb{R}^m$, where $J(x_0; u + \nu)$ exists.

Proof. From (3.20) with $\mu_T = -K\xi$, it follows that

$$\mu(t) = -K\xi(t) + e^{-st}(\mu(0) + K\xi(0)) = -K\xi(t) + \nu(t) \tag{3.22}$$

using $\mu(0) = u_0$ and $\xi(0) = x_0$. Then, the dynamics of $\xi(t)$ in (3.14) is identical to (2.10) with the control input $\mu = u + \nu$, i.e., $x(t) = \xi(t)$ for all $t \geq 0$, which completes the proof using the definition in (2.12). □

Note that $\mathcal{Q}(x_0, u_0)$ corresponding to a policy $u = -Kx$, and $J(x_0; u)$ are different in general, except for the case when $u_0 = -Kx_0$. Indeed, in this case, $\nu(t) = 0$ in Proposition 3.5, which implies $\mathcal{Q}(x_0, u_0) = J(x_0; u)$. Nevertheless, the surrogate Q-function is utilized to obtain the optimal control input in (2.15) using a policy iteration method introduced in the following section.

3.2.2 The Surrogate Q-Learning Algorithm

The surrogate Q-function is a policy iteration method that iteratively evaluates the policy using the surrogate Q-function and improves the policy according to the evaluation for the next iteration.

The Policy Evaluation

Let $\mathcal{Q}_k(x_0, u_0)$ be the surrogate Q-function of the state $x_0 \in \mathbb{R}^n$ and the control input $u_0 \in \mathbb{R}^m$ for the policy $u = -K_k x$, where $K_k \in \mathbb{R}^{m \times n}$ denotes the gain matrix at the k-th iteration for the design parameter $s_k > 0$. By differentiating (3.15) with respect to t at $t = 0$, it follows that $\mathcal{Q}_k(x_0, u_0)$ should satisfy

$$\boldsymbol{\nabla}_x \mathcal{Q}_k(x_0, u_0)^T \dot{\xi}(0) + \boldsymbol{\nabla}_u \mathcal{Q}_k(x_0, u_0)^T \dot{\mu}(0) + x_0^T Q x_0 + u_0^T R u_0 = 0 \tag{3.23}$$

for all $(x_0, u_0) \in \mathbb{R}^n \times \mathbb{R}^m$, where $\dot{\xi}(0)$ and $\dot{\mu}(0)$ can be obtained from (3.14) with $K = K_k$ as a linear function of x_0 and u_0. As discussed in Section 3.2.1, $\mathcal{Q}(x_0, u_0)$ represents the infinite-horizon cost of the policy $u = -K_k x$, where the virtual autonomous system (3.14) with $(\xi(0), \mu(0)) = (x_0, u_0)$.

The Policy Improvement

Once the policy $u = -K_k x$ is evaluated by $\mathcal{Q}_k(x_0, u_0)$, the policy improvement is defined by finding a new policy $u = -K_{k+1}x$ for $K_{k+1} \in \mathbb{R}^{m \times n}$ that satisfies

$$\boldsymbol{\nabla}_u \mathcal{Q}_k(x_0, u)|_{u=-K_{k+1}x_0} = 0 \tag{3.24}$$

for all $x \in \mathbb{R}^n$. The improved control input $u = -K_{k+1}x_0$ is an extremum point of $\mathcal{Q}_k(x_0, u)$ for each $x_0 \in \mathbb{R}^n$ with respect to u but is not necessarily a minimum point.

To investigate the role of (3.24), let us consider the optimal stable policy $u^* = -K^* x$, where K^* denotes the optimal stable gain of the optimal control problem in (2.12). Fix the design parameters as $s_k = s_{k+1} = s$, and let

$$K_k = K_{k+1} = K^*. \tag{3.25}$$

The policy u^* is admissible, and therefore the corresponding performance index $J(x_0; u^*)$ in (2.12) is well-defined for all $x_0 \in \mathbb{R}^n$. It follows that there exists a small neighborhood $\mathcal{N} \subset \mathbb{R}^m$ around $u_0^* := -K^* x_0$, such that for any $u_0 \in \mathcal{N}$, the performance index $J(x_0; u^* + \nu)$ in Proposition 3.5 is well-defined, where $\nu(t) = e^{-st}(u_0 - u^*)$. It follows that

$$\mathcal{Q}(x_0, u_0) = J(x_0; u^* + \nu) \geq \min_{u_0 \in \mathcal{N}} J(x_0; u^* + \nu) \tag{3.26}$$

for any $(x_0, u_0) \in \mathbb{R}^n \times \mathcal{N}$, where the equality is satisfied if $u_0 = u_0^*$. Therefore, the condition (3.24) can be regarded as a locally necessary condition for $K_k = K_{k+1} = K^*$.

The Equivalent Matrix Iteration

If the hypotheses of Proposition 3.4 are satisfied for $K = K_k$, there exists a unique surrogate Q-function $\mathcal{Q}_k(x_0, u_0)$ given by

$$\mathcal{Q}_k(x_0, u_0) = \begin{bmatrix} x_0^T & u_0^T \end{bmatrix} M_k \begin{bmatrix} x_0 \\ u_0 \end{bmatrix} \tag{3.27}$$

for some $M_k \in \mathbb{S}^{n+m}$, which satisfies the policy evaluation equation in (3.23) for all $(x_0, u_0) \in \mathbb{R}^n \times \mathbb{R}^m$. Substituting (3.27) into (3.23) yields

$$2 \begin{bmatrix} x_0^T & u_0^T \end{bmatrix} M_k \begin{bmatrix} \dot{\xi}(0) \\ \dot{\mu}(0) \end{bmatrix} + \begin{bmatrix} x_0^T & u_0^T \end{bmatrix} Q^\circ \begin{bmatrix} x_0 \\ u_0 \end{bmatrix} = 0, \tag{3.28}$$

where Q° is defined in (3.19). From (3.14) with $K = K_k$, it follows that

$$\begin{bmatrix} \dot{\xi}(0) \\ \dot{\mu}(0) \end{bmatrix} = \begin{bmatrix} A & B \\ -K_k A - s_k K_k & -K_k B - s_k I_m \end{bmatrix} \begin{bmatrix} x_0 \\ u_0 \end{bmatrix} =: A_k^\circ \begin{bmatrix} x_0 \\ u_0 \end{bmatrix}. \tag{3.29}$$

Substituting (3.29) into (3.28) and requiring that (3.28) holds for all (x_0, u_0) yield the following matrix equation:

$$M_k A_k^\circ + A_k^{\circ T} M_k + Q^\circ = 0. \tag{3.30}$$

Let us define a matrix $U_k \in \mathbb{R}^{(n+m)\times(n+m)}$, which is invertible for any $K_k \in \mathbb{R}^{m\times n}$, as

$$U_k = \begin{bmatrix} I_n & 0 \\ -K_k & I_m \end{bmatrix}, \quad U_k^{-1} = \begin{bmatrix} I_n & 0 \\ K_k & I_m \end{bmatrix}. \tag{3.31}$$

Pre-multiplying U_k^T and post-multiplying U_k to the both sides of (3.30) yield an equivalent matrix equation to the policy evaluation in (3.23) given by

$$H_k S_k + S_k^T H_k + Q_k = 0, \tag{3.32}$$

where $H_k := U_k^T M_k U_k \in \mathbb{S}^{n+m}$, and

$$S_k := \begin{bmatrix} A - BK_k & B \\ 0 & -s_k I_m \end{bmatrix}, \quad Q_k := \begin{bmatrix} Q + K_k^T R K_k & -K_k^T R \\ -RK_k & R \end{bmatrix} \tag{3.33}$$

using the facts that $U_k^{-1} A_k^\circ U_k = S_k$ and $U_k^T Q^\circ U_k = Q_k$.

On the other hand, substituting (3.27) into (3.24) and using (3.31) yield

$$\begin{bmatrix} 0 & I_m \end{bmatrix} M_k \begin{bmatrix} I_n \\ -K_{k+1} \end{bmatrix} x_0 = \begin{bmatrix} 0 & I_m \end{bmatrix} H_k \begin{bmatrix} I_n \\ K_k - K_{k+1} \end{bmatrix} x_0 = 0 \tag{3.34}$$

for all $x_0 \in \mathbb{R}^n$, which yields an equivalent matrix equation to the policy improvement in (3.24) as

$$K_{k+1} = K_k + G_k^{-1} W_k, \tag{3.35}$$

where the matrix H_k is decomposed as

$$H_k = \begin{bmatrix} P_k & W_k^T \\ W_k & G_k \end{bmatrix} \tag{3.36}$$

for $P_k \in \mathbb{R}^{n \times n}$, $W_k \in \mathbb{R}^{m \times n}$, and $G_k \in \mathbb{R}^{m \times m}$.

Consequently, the convergence of the surrogate Q-learning represented by the policy evaluation in (3.23) and the policy improvement in (3.24) is equivalent to the convergence of the iteration composed of matrix equations in (3.32) and (3.35). This iteration is referred to as the *extended Kleinman iteration*. The next section will introduce the off-policy and model-free extension of the surrogate Q-learning before presenting the detailed convergence analysis of the extended Kleinman iteration.

3.2.3 The Data-Driven Surrogate Q-Learning

Let us first assume that the hypotheses of Proposition 3.4 with $K = K_k$ and $s = s_k$ are satisfied for all iteration step $k \geq 0$ in this section. Note that this assumption will be relaxed in the subsequent section. Under this assumption, for any $k \geq 0$, there exist a unique solution M_k to (3.30) and the corresponding H_k in (3.32). Although, the next-step policy K_{k+1} in (3.35) can be obtained using only H_k and K_k, the matrix M_k in (3.30) for calculating H_k from $H_k = U_k^T M_k U_k$ requires the knowledge of system matrices A and B, which contradicts the objective of developing a model-free method. The typical approach to relieve this requirement in the literature of ADP methods is utilizing a dataset acquired from the system [2].

Rearranging (3.14) at $t = 0$ yields

$$\dot{\xi}(0) = Ax_0 + Bu_0 =: \dot{x}_0, \tag{3.37a}$$

$$\dot{\mu}(0) = -K_k\dot{x}_0 - s_k(u_0 + K_k x_0), \tag{3.37b}$$

which implies that a tuple $(x_0, u_0, \dot{x}_0)$ can fully determine the vectors in (3.28), which is required to calculate the matrix M_k. For a dataset $\mathcal{D} = \{(x_i, u_i, \dot{x}_i)\}_{i=1}^{n_d}$, the i-th tuple $(x_i, u_i, \dot{x}_i)$ in the dataset is considered as the initial point of the corresponding *virtual* trajectories of $\xi^{(i)}(t)$ and $\mu^{(i)}(t)$ satisfying $\xi^{(i)}(0) = x_i$ and $\mu^{(i)}(0) = u_i$ in (3.14), and $\dot{\xi}^{(i)}(0) = \dot{x}_i$ which can be obtained by exerting u_i to the system (2.10) at the state x_i.

For the i-th tuple $(x_i, u_i, \dot{x}_i)$ in the dataset $\mathcal{D}$, let

$$\zeta_i := \begin{bmatrix} x_i \\ u_i \end{bmatrix}, \quad \dot{\zeta}_i := \begin{bmatrix} \dot{x}_i \\ -K_k\dot{x}_i - s_k(u_i + K_k x_i) \end{bmatrix}. \tag{3.38}$$

Then, (3.28) can be rewritten as

$$\left\{\left(\dot{\zeta}_i^T \otimes \zeta_i^T\right) + \left(\zeta_i^T \otimes \dot{\zeta}_i^T\right)\right\}\mathrm{vec}(M_k) = -\left(\zeta_i^T \otimes \zeta_i^T\right)\mathrm{vec}(Q^\circ), \tag{3.39}$$

where $\otimes$ denotes the Kronecker product, and $\mathrm{vec}(\cdot)$ denotes the vectorization operator. Let

$$X_k := \begin{bmatrix} \left(\dot{\zeta}_1^T \otimes \zeta_1^T\right) + \left(\zeta_1^T \otimes \dot{\zeta}_1^T\right) \\ \vdots \\ \left(\dot{\zeta}_{n_d}^T \otimes \zeta_{n_d}^T\right) + \left(\zeta_{n_d}^T \otimes \dot{\zeta}_{n_d}^T\right) \end{bmatrix}, \quad Z := \begin{bmatrix} \zeta_1^T \otimes \zeta_1^T \\ \vdots \\ \zeta_{n_d}^T \otimes \zeta_{n_d}^T \end{bmatrix} \mathrm{vec}(Q^\circ). \tag{3.40}$$

Then, the solution M_k can be obtained from the following data-driven policy evaluation:

$$\mathrm{vec}(M_k) = X_k^\dagger Z \in \mathbb{R}^{(n+m)\times(n+m)}. \tag{3.41}$$

By construction of the matrices X_k and Z using the Kronecker product, the element of $\mathrm{vec}(M_k)$ inherently produces the symmetric matrix M_k provided that there exist $n_d \geq (n+m)(n+m+1)/2$ independent rows in X_k, which are the data points. Each row corresponds to a data tuple in the dataset, indicating that the proposed algorithm ultimately requires a minimum of $(n+m)(n+m+1)/2$ data points. Although this data requirement is larger than $n(n+1)/2$, which is the minimum number of data required for Kleinman iteration-based ADP technique [2], it can be seen as a trade-off for stabilizing the initial unstable control inputs.

It should be noted that the process of obtaining M_k using the data-driven policy evaluation steps as defined by (3.41) does not require knowledge of the system matrices A and B, thereby making it a model-free method. Additionally, any control input u_i can be utilized to construct the dataset $\mathcal{D}$, resulting in

the acquisition of the data tuple $(x_i, u_i, \dot{x}_i)$. Consequently, this method can be categorized as an off-policy approach.

3.3 The Extended Kleinman Iteration

This section presents a detailed analysis of the extended Kleinman iteration introduced in Section 3.2.2, including the proof of convergence. Similar to the Kleinman iteration, each iteration of the algorithm consists of two steps: the policy evaluation step (3.32) and the policy improvement step (3.35). The proposed method, however, converges to the optimal stabilizing solution with both stable and unstable initial gains under a minimal assumption for solving the initial policy evaluation step.

The formal definition of the extended Kleinman iteration is given below.

Definition 3.6 (The extended Kleinman iteration)**.** Given matrices $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $Q \in \mathbb{S}^n_+$, $R \in \mathbb{S}^m_{++}$, and $K_0 \in \mathbb{R}^{m\times n}$, the *extended Kleinman iteration* recursively solves the following Lyapunov equation of $H_k \in \mathbb{S}^{n+m}$, called the policy evaluation step:

$$H_k S_k + S_k^T H_k + Q_k = 0, \tag{3.42}$$

and finds the next-step policy K_{k+1}, called the policy improvement step, as follows:

$$K_{k+1} = K_k + G_k^{-1} W_k, \tag{3.43}$$

for all $k = 0, 1, \ldots,$ where S_k and Q_k are defined in (3.33), and G_k and W_k are defined in (3.36).

The conditions for the existence of the solutions are presented in the next section. First, the existence of the optimal stabilizing solution to the ARE is revisited, and then mild conditions for solving (3.42) to obtain H_k for all steps $k \geq 0$ are revealed based on the matrix inertia theorem.

3.3.1 Existence of Solutions

The Optimal Stabilizing Solution

In the extended Kleinman iteration, the optimal stabilizing solution is defined by $P^* \in \mathbb{S}^n_{++}$ that satisfies $\mathcal{R}(P^*) = 0$ in (2.14). Theorem 2.14 guarantees the existence of P^* if (A, B) is controllable and (A, Q) is observable, and moreover, the optimal stabilizing feedback gain defined by $K^* = R^{-1}B^T P^*$ satisfies that $A_c^* := A - BK^*$ is Hurwitz. Although there is a result for the existence of the optimal stabilizing solution under much relaxed conditions that (A, B) is stabilizable and that (A, Q) is detectable [53], this extension is out of the scope of the dissertation.

Solutions to Policy Iteration Steps

At each step $k \geq 0$, a feedback gain $K_k \in \mathbb{R}^{m\times n}$ is given, and the corresponding closed-loop system matrix is denoted by $A_k := A - BK_k$. Let the design parameter s_k in (3.33) be chosen from a set $\mathcal{S}_k$ defined by

$$\mathcal{S}_k := \{s \in \mathbb{R} \mid s > 0,\ s \notin \sigma(A_k),\ \delta(G_k) = 0\}, \tag{3.44}$$

where the set $\{s \in \mathbb{R} \mid s > 0, s \notin \sigma(A_k)\}$ is always nonempty for any finite-dimensional matrix $A_k \in \mathbb{R}^{n\times n}$. Note that given $s_k \in \mathcal{S}_k$, $\sigma(S_k) \cap \sigma(-S_k) = \varnothing$ if and only if $\sigma(A_k) \cap \sigma(-A_k) = \varnothing$. Therefore, there exists a unique solution $H_k \in \mathbb{S}^{n+m}$ to the Lyapunov equation in the policy evaluation step (3.42) if and only if $\sigma(A_k) \cap \sigma(-A_k) = \varnothing$ by Theorem 2.15. On the other hand, by partitioning H_k as in (3.36), the next-step policy K_{k+1} in the policy improvement step in (3.43) exists if and only if the matrix G_k is nonsingular, i.e, $\delta(G_k) = 0$.

In summary, each iteration step requires that $\sigma(A_k) \cap \sigma(-A_k) = \varnothing$ and that G_k is nonsingular. The next lemma states that whenever the solution to (3.42) exists at the initial step, $k = 0$, the solutions to the subsequent iteration steps are all well-defined under some mild assumptions.

Lemma 3.7. *In the extended Kleinman iteration, suppose that (A, Q) is observable. Given K_0 satisfying $\sigma(A_0) \cap \sigma(-A_0) = \varnothing$, if $s_k \in \mathcal{S}_k$ for all $k \geq 0$, then the following conditions hold for all $k \geq 0$.*

(i) $\sigma(A_k) \cap \sigma(-A_k) = \varnothing$.

(ii) there is a unique nonsingular solution H_k to (3.42).

(iii) $\mathrm{In}(P_k) = \mathrm{In}(-A_k)$.

Proof. Note that the Lyapunov equation in (3.42) can be equivalently expressed as

$$P_k A_k + A_k^T P_k + Q + K_k^T R K_k = 0, \tag{3.45}$$

$$W_k(A_k - s_k I_n) + B^T P_k - R K_k = 0, \tag{3.46}$$

$$W_k B + B^T W_k^T - 2 s_k G_k + R = 0. \tag{3.47}$$

Since $\sigma(A_0) \cap \sigma(-A_0) = \varnothing$, the proof is by induction on k. Let $\sigma(A_k) \cap \sigma(-A_k) = \varnothing$ for some $k > 0$. Because $s_k \notin \sigma(A_k)$, it follows that $\sigma(S_k) \cap \sigma(-S_k) = \varnothing$. Then, there exists a unique nonsingular H_k by applying Theorem 2.16 to (3.42). Similarly, $\sigma(A_k) \cap \sigma(-A_k) = \varnothing$ implies that $\mathrm{In}(P_k) = \mathrm{In}(-A_k)$ from Theorem 2.16. Hence, the conditions (ii) and (iii) are direct results of the condition (i).

Since G_k is invertible by $s_k \in \mathcal{S}_k$, K_{k+1} is well-defined from (3.43), and the Schur complement of G_k in H_k can be defined by $H_k/G_k = P_k - W_k^T G_k^{-1} W_k$.

From (3.45) to (3.47), it can be observed that H_k/G_k satisfies

$$(H_k/G_k)A_{k+1} + A_{k+1}^T(H_k/G_k) + Q + K_{k+1}^T R K_{k+1} = 0, \tag{3.48}$$

which implies that H_k/G_k is a solution to (3.45) with $k+1$. It follows that $\sigma(A_{k+1}) \cap \sigma(-A_{k+1}) = \varnothing$ from Lemma 2.17, which completes the proof. $\square$

3.3.2 Selection of Design Parameters

Consider the following simple rule for the choice of $s_k \in \mathcal{S}_k$.

$$s_0 \in \mathcal{S}_0, \quad s_k = \begin{cases} s_{k-1} & \text{if } s_{k-1} \notin \mathcal{S}_k, \\ s_k^+ \in \mathcal{G}_k & \text{otherwise,} \end{cases} \tag{3.49}$$

where $\mathcal{G}_k := \{s \in \mathcal{S}_k \mid \nu(G_k) \geq 1\} \subset \mathcal{S}_k$, which implies that s_k remains constant unless it becomes one of the eigenvalues of A_k.

It rarely happens that $s_{k-1} \in \sigma(A_k)$ in practice. Even if it happens, however, the following lemma ensures that the set $\mathcal{G}_k$ is nonempty in the neighbor of s_{k-1} that does not contain s_{k-1}, denoted by $\mathcal{N}(s_{k-1}) \subset \mathcal{S}_k$. Therefore, s_k^+ can be easily selected around s_{k-1} by inspecting $\nu(G_k)$.

Lemma 3.8. *In the extended Kleinman iteration, suppose that the hypotheses of Lemma 3.7 hold and that (A, B) is controllable. If $s_{k-1} \in \sigma(A_k)$, then $\mathcal{G}_k \cap \mathcal{N}(s_{k-1}) \neq \varnothing$.*

Proof. At the step k, K_k satisfies $\sigma(A_k) \cap \sigma(-A_k) = \varnothing$ by Lemma 3.7 (i), and therefore P_k is well-defined from (3.45) regardless of the choice of s_k. However, W_k and G_k depend on the choice of s_k from (3.46) and (3.47). From (3.46), (3.47), and (3.53), it follows that

$$2s_k G_k = 2\lambda G_{k-1} + (\lambda + s_k)V_k + (\lambda + s_k)V_k^T, \tag{3.50}$$

where $V_k := W_{k-1}(A_k - s_k I_n)^{-1}B$, and $0 < \lambda = s_{k-1} \in \sigma(A_k)$. Let the spectral decomposition of A_k be given by $\Psi^{-T}\Lambda\Psi^T$, where (j,j)-element of Λ is λ. Put $e_j := [0,\ldots,0,1,0,\ldots,0]^T$, with a plus one in the j-th component and zeros elsewhere, so that $\Psi e_j = \psi$, where ψ^T is the left eigenvector of A_k associated with λ. Since (A,B) is controllable, (A_k,B) is also controllable [62, Theorem 1.1], which implies that $\psi^T B \neq 0$. Then, for all $w \in \mathcal{W} := \{w \in \mathbb{R}^m \mid \psi^T B w \neq 0\}$, which is nonempty, it follows that

$$\begin{aligned}\lim_{s_k\to\lambda}(\lambda - s_k)w^T V_k w &= w^T W_{k-1}\Psi^{-T}\lim_{s_k\to\lambda}(\lambda - s_k)(\Lambda - s_k I_n)^{-1}\Psi^T B w \\ &= w^T W_{k-1}\Psi^{-T} e_j e_j^T \Psi^T B w \\ &= w^T W_{k-1} v \psi^T B w,\end{aligned} \tag{3.51}$$

where $v := \Psi^{-T}e_j = \Psi^{-T}\Psi^{-1}\psi$ is the right eigenvector of A_k associated with λ. If $W_{k-1}v = 0$, then

$$\lambda v = A_k v = A_{k-1}v - BG_{k-1}^{-1}W_{k-1}v = A_{k-1}v, \tag{3.52}$$

which contradicts $\lambda = s_{k-1} \notin \sigma(A_{k-1})$. Therefore, $W_{k-1}v \neq 0$, and there exists $w \in \mathcal{W}$ such that $w^T W_{k-1} v \neq 0$ and that the limit of $(\lambda - s_k)w^T G_k w$ is nonzero as s_k approaches λ from (3.50) and (3.51). This implies that there exists $s_k^+ \in \mathcal{N}(\lambda)$ such that $w^T G_k w < 0$, which completes the proof. $\square$

3.4 Convergence Analysis

The convergence analysis of the extended Kleinman iteration is presented in this section. The investigation starts with the analysis of the monotonic stabilization property in the sense of the matrix inertia. Then, the local convergence is analyzed using the Fréchet derivatives, and finally the proof of the following global convergence theorem is presented using the local convergence property.

Theorem 3.9. *In the extended Kleinman iteration with the design parameter selection rule in* (3.49)*, suppose that* (A, B) *is controllable and* (A, Q) *is observable. If* K_0 *satisfies* $\sigma(A_0) \cap \sigma(-A_0) = \varnothing$*, and* $s_k \in \mathcal{S}_k$*, there exists a finite integer* $N \geq 0$ *such that the following properties hold for all* $k \geq N$*.*

1. A_k *is Hurwitz,*
2. $P_k \succeq P_{k+1} \succeq P^*$*,*
3. $\lim_{k\to\infty} K_k = K^*$ *and* $\lim_{k\to\infty} P_k = P^*$*,*

where P^* *is a unique positive definite solution to* $\mathcal{R}(P) = 0$*, and* $K^* = R^{-1}B^T P^*$*.*

Proof. The proof is given in Section 3.4.3. □

It is assumed, without loss of generality, that B has full column rank throughout the convergence analysis.

3.4.1 Monotonic Stabilization

From (3.48), it can be observed that $H_k/G_k = P_{k+1}$, and further,

$$P_{k+1} = P_k - W_k^T G_k^{-1} W_k =: P_k - D_k \tag{3.53}$$

using (3.45) to (3.47). Note that if $G_k \succ 0$, P_k is monotonically decreasing as $P_k \succeq P_{k+1}$, which is the similar result of the Kleinman iteration when $P_k \succ 0$ for all $k \geq 0$, but gives little information when $\nu(P_k) \geq 1$. Following lemma demonstrates an interesting property of the update law in (3.43), showing that $\nu(P_k)$, or equivalently $\pi(A_k)$ according to Lemma 3.7 (iii), monotonically decreases.

Lemma 3.10. *Under the hypotheses of Lemma 3.7,* $\pi(A_{k+1}) = \pi(A_k) - \nu(G_k)$ *for all* $k \geq 0$.

Proof. From (3.42) and Theorem 2.16, it follows that

$$\nu(H_k) = \nu(-S_k) = \pi(A_k). \tag{3.54}$$

Since G_k is nonsingular for all k by $s_k \in \mathcal{S}_k$, applying Haynsworth's inertia theorem [63, Theorem 1] yields $\mathrm{In}(H_k) = \mathrm{In}(G_k) + \mathrm{In}(H_k/G_k)$, which is followed by $\nu(P_{k+1}) = \nu(H_k) - \nu(G_k) = \pi(A_k) - \nu(G_k)$ from (3.53) and (3.54). Then, Lemma 3.7 (iii) gives $\nu(P_{k+1}) = \pi(A_{k+1})$, which completes the proof. □

Although Lemma 3.10 guarantees that $\pi(A_k)$ is monotonically decreasing for any choice of $s_k \in \mathcal{S}_k$ and provides a strict decreasing condition, which is given by $\nu(G_k) \geq 1$, it is desirable for the proposed iteration algorithm to have that $\pi(A_k)$ decreases to 0, or equivalently, A_k becomes Hurwitz, in a finite number of iterations. Theorem 3.9 states that the extended Kleinman iteration with the mild assumptions for the existence of the solutions in Section 3.3.1 is enough to guarantee that $\pi(A_k) = 0$ in a finite number of iterations and that P_k and K_k converge to their optimal stable solution.

Remark 3.11. Unlike inexact Kleinman iteration methods such as [64], $\mathrm{In}(A_k)$ can be directly obtained from $\mathrm{In}(P_k)$ by Lemma 3.7 (iii). Hence, it can be determined when A_k becomes Hurwitz by examining $\mathrm{In}(P_k)$. This is important

because whenever A_k is Hurwitz, the subsequent A_k's are all Hurwitz by Theorem 3.9.

Remark 3.12. The Kleinman iteration is a Newton method, and therefore it has a quadratic convergence rate, while the proposed policy iteration does not possess such a fast convergence property. However, the proposed method converges with an arbitrary initial feedback gain, and from Theorem 3.9, the policy will be stable in a finite number of iterations. Therefore, for a better convergence speed, it is recommended to consider a hybrid approach: use the extended Kleinman iteration first, and switch to the Kleinman iteration once the policy becomes stable.

3.4.2 Local Convergence

This section presents that the proposed algorithm converges *locally* to the optimal stabilizing solution in terms of discrete-time Lyapunov stability. A sequence of K_k generated by Theorem 3.9 can be regarded as the solution to a nonlinear discrete-time system determined by the initial state. Given an equilibrium point of the discrete-time system, the local convergence of the equilibrium point can be evaluated by the spectral radius of the linearized system matrix.

Following proposition gives the stability relation between linearized continuous-time and discrete-time system matrices, when the two matrices have a relation similar to the bilinear transformation.

Proposition 3.13. *Given two matrices $A_1, A_2 \in \mathbb{R}^{n\times n}$, suppose that a scalar $s > 0$ satisfies $A_2 = (A_1 + sI_n)(A_1 - sI_n)^{-1}$, and that s is not an eigenvalue of A_1. Then, A_1 is Hurwitz if and only if $\rho(A_2) < 1$.*

Following lemma states that the optimal stabilizing solution P^* is the only locally stable equilibrium point under the extended Kleinman iteration.

Lemma 3.14. *Under the hypotheses of Theorem 3.9, $K^* := R^{-1}B^TP^*$ is the unique locally stable equilibrium of K_k, where P^* is the positive definite solution of $\mathcal{R}(P) = 0$.*

Proof. The subsequent K_k is completely determined by an initial gain K_0 and a sequence $\{s_k\}$ under the policy iteration (3.42) and (3.43). Therefore, K_k can be viewed as a solution to a discrete-time nonlinear dynamical system. It will be shown that K^* is the unique locally stable equilibrium of the discrete-time system.

To find all the equilibriums of the system, suppose that there exists $N \geq 0$ such that for all $k \geq N$, $K_{k+1} = K_k =: K$. Since $\sigma(A_k)$ is unchanged for all $k \geq N$, it follows that $s_{k+1} = s_k =: s$. Then, let $H_k = H$ be the unique symmetric solution to (3.42), which is given by

$$H = \begin{bmatrix} P & W^T \\ W & G \end{bmatrix}. \tag{3.55}$$

From (3.43), it follows that $G^{-1}W = G_k^{-1}W_k = K_{k+1} - K_k = 0$. Because G is nonsingular from $s \in \mathcal{S}_k$, it follows that $W = 0$, hence $K = R^{-1}B^TP$ from (3.46). Substituting K into (3.45) yields that P is a symmetric solution to $\mathcal{R}(P) = 0$.

It is now shown that only $K^* = R^{-1}B^TP^*$ is locally stable among the equilibriums corresponding to the symmetric solutions of $\mathcal{R}(P) = 0$. By continuity, there exists $\delta > 0$ such that $s \notin \sigma(A - BK - BE_k)$ for all E_k satisfying $\|E_k\|_F < \delta$. Put $K_k = K + E_k$, where $\|E_k\|_F < \delta \ll 1$, and let $s_k = s$. Because

W_k and G_k are the functions of K_k, which can be deduced from (3.42), let us define a function $f : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ to be $f(K_k) = K_k + G_k^{-1}W_k$, which implies that $K_{k+1} = f(K_k)$ from (3.43). The first-order Taylor expansion of f at K using the Fréchet derivative is given by

$$f(K_k) \simeq f(K) + L_f(K, E_k) = K + \mathcal{D}[f] \tag{3.56}$$

by denoting $\mathcal{D}[\cdot] = L_{(\cdot)}(K, E_k)$ for short. Let $A_c \coloneqq A - BK$. From (3.45), P_k is a matrix function of K_k, and therefore it follows that

$$\mathcal{D}[P_k]A_c + A_c^T\mathcal{D}[P_k] = 0, \tag{3.57}$$

from $\mathcal{D}\big[K_k^T RK_k\big] = E_k^T RK + K^T RE_k$. Then, (3.57) becomes the Lyapunov equation which only has the unique trivial solution $\mathcal{D}[P_k] = 0$ because $\sigma(A_c) \cap \sigma(-A_c) = \varnothing$ from Lemma 3.7 (i).

Meanwhile, it follows that

$$\mathcal{D}[W_k(A_k - sI_n)] = \mathcal{D}[W_k](A_c - sI_n) \tag{3.58}$$

using $W = 0$, and therefore

$$\mathcal{D}[W_k] = RE_k(A_c - sI_n)^{-1} \tag{3.59}$$

using $W_k(A_k - sI_n) = RK_k - B^T P_k$ from (3.46) and $\mathcal{D}[K_k] = E_k$. Using (3.59) and $R = 2sG$ from (3.47), it follows that

$$\begin{aligned}\mathcal{D}[f] &= E_k + \mathcal{D}\big[G_k^{-1}\big]W + G^{-1}\mathcal{D}[W_k] \\ &= E_k + 2sE_k(A_c - sI_n)^{-1}.\end{aligned} \tag{3.60}$$

From (3.56) and (3.60), the update law for E_k can be rewritten as follows.

$$E_{k+1} = E_k(A_c + sI_n)(A_c - sI_n)^{-1}. \tag{3.61}$$

From Proposition 3.13, E_k, and therefore K_k, is locally stable if and only if A_c is Hurwitz, or equivalently when $P = P^*$, which is the unique stabilizing solution to $\mathcal{R}(P) = 0$. $\square$

Remark 3.15. As shown in the proof of Lemma 3.14, all symmetric solutions of $\mathcal{R}(P) = 0$ are also equilibriums, although the stabilizing solution P^* is only stable. Therefore, the iteration may become stuck on trivial unstable solutions, which are $W_k = 0$ and not positive definite P_k. However, whenever the iteration is stuck with $W_k = 0$, it can be checked if $P_k \succ 0$, and if not, the iteration can be reinitialized with different K_0. In the following section, it is assumed that the iteration is not stuck on trivial unstable solutions.

Lemma 3.14 can only guarantee the local convergence. However, this result will be used to prove the global convergence of the extended Kleinman iteration in the next section.

3.4.3 Global Convergence

The proof of global convergence is be divided into two steps. First, it will be shown that A_k becomes Hurwitz in a finite number of iterations, and then P_k and K_k converge to their optimal stable solutions, respectively.

For the first step, it is sufficient to show that for all integer $k_i \geq 0$ such that A_{k_i} is unstable, there exists an integer $N_i \geq k_i$ satisfying $\nu(G_{N_i}) \geq 1$. Hence, the objective of the global convergence proof is to show that it is impossible to have $G_k \succ 0$ for all $k \geq k_i$ when A_{k_i} is not Hurwitz, or equivalently, $\pi(A_{k+1}) = \pi(A_k) \geq 1$ for all $k \geq k_i$. Whenever $s_{k-1} \in \sigma(A_k)$, the scalar $s_k \in \mathcal{G}_k$ yields $\nu(G_k) \geq 1$ by definition, which implies $\pi(A_{k+1}) \leq \pi(A_k) - 1$. Therefore, in the remainder of this section, it is assumed that $s_k = s$ and $k_i = 0$ without loss of

generality to prove the first step.

Let P^* be the symmetric, positive definite solution to $\mathcal{R}(P) = 0$, and $A_c^* = A - BK^*$, where $K^* = R^{-1}B^T P^*$. Since $\pi(A_c^*) = 0$, it follows that $(A_c^* - sI_n)^{-1}$ is nonsingular for any $s > 0$, and therefore an auxiliary matrix $\bar{G}_k$ can be defined by

$$\bar{G}_k \coloneqq C_k^T G_k C_k, \tag{3.62}$$

where $C_k \coloneqq I_m - (K_k - K^*)U$, and $U \coloneqq (A_c^* - sI_n)^{-1}$.

If C_k is singular, then there exists a nonzero vector $v \in \mathbb{R}^m$ such that $C_k v = 0$. Because $BC_k = (A_k - sI_n)U$, and B has full column rank, it follows that Uv is an eigenvector of A_k with an eigenvalue s. This contradicts $s \notin \sigma(A_k)$, which implies that C_k is nonsingular. It follows that the two real symmetric matrices $\bar{G}_k$ and G_k are congruent from (3.62), and therefore

$$\mathrm{In}(\bar{G}_k) = \mathrm{In}(G_k) \tag{3.63}$$

from Sylvester's law of inertia in Theorem 2.15.

The auxiliary matrix $\bar{G}_k$ can be further rewritten using (3.45) to (3.47) as $\bar{G}_k = U^T(P_k - P)U + \frac{1}{2s}R$, which has the following update equation.

$$\bar{G}_{k+1} = \bar{G}_k - U^T D_k U. \tag{3.64}$$

Lemma 3.16. *Under the hypotheses of Theorem 3.9, suppose that there exists a sequence of K_k, for $k = 0, 1, \ldots,$ and the corresponding P_k, which is the solution to* (3.45)*. If the sequence of P_k is bounded, then the sequence of $\|K_k\|_F$ is also bounded.*

Proof. Suppose that $\|K_k\|_F$ is not bounded, and let $\beta_k \coloneqq \|K_k\|_F$. Put $Y_k \coloneqq K_k/\beta_k$, which is bounded and satisfies $\lim_{k\to\infty} Y_k \neq 0$. Dividing both sides

of (3.45) by β_k^2, it follows that

$$P_k\left(\frac{1}{\beta_k^2}A - \frac{1}{\beta_k}BY_k\right) + \left(\frac{1}{\beta_k^2}A - \frac{1}{\beta_k}BY_k\right)^T P_k + \frac{1}{\beta_k^2}Q + Y_k^T RY_k = 0. \quad (3.65)$$

Because P_k and Y_k are bounded and $\lim_{k\to\infty}\beta_k = \infty$, it can be concluded that

$$\lim_{k\to\infty} Y_k^T RY_k = 0, \quad (3.66)$$

which contradicts $\lim_{k\to\infty} Y_k \neq 0$. □

Lemma 3.17. *Given $U \in \mathbb{R}^{n\times m}$ such that (A_k, U) is controllable, suppose that there is a sequence of K_k such that the unique symmetric solution P_k to* (3.45) *exists, and the sequence of $U^T P_k U$ is bounded. Then, the sequence of P_k is also bounded.*

Proof. Suppose that P_k is not bounded. Since $U^T P_k U$ is bounded, there is an orthonormal basis (V_{k1}, V_{k2}, V_{k3}) such that $\mathrm{Im}(U) = \mathrm{Im}(V_{k1})$, and that P_k can be decomposed as

$$P_k = \begin{bmatrix} V_{k1} & V_{k2} & V_{k3} \end{bmatrix} \begin{bmatrix} \Sigma_{k1} & 0 & 0 \\ 0 & \Sigma_{k2} & 0 \\ 0 & 0 & \Sigma_{k3} \end{bmatrix} \begin{bmatrix} V_{k1}^T \\ V_{k2}^T \\ V_{k3}^T \end{bmatrix}, \quad (3.67)$$

where Σ_{k1} and Σ_{k2} are bounded, but Σ_{k3} is not bounded. From [65, Lemma 2.1], P_k satisfies

$$P_k A_k^i + \left(A_k^i\right)^T P_k + \eta_i(K_k) = 0 \quad (3.68)$$

for all $i \geq 1$, where η_i is a polynomial function, which is bounded when $\|K_k\|_F$ is bounded.

If $\|K_k\|_F$ is bounded, it follows from (3.68) that $V_{k3}^T A_k^i U$ vanishes for all $i \geq 1$, which contradicts the assumption that (A_k, U) is controllable. Hence,

$\|K_k\|_F$ is unbounded, and therefore A_k has at least one unbounded eigenvalue since B has full column rank.

Let the real Schur decomposition of A_k to be

$$A_k = \begin{bmatrix} \Psi_k & \psi_k \end{bmatrix} \begin{bmatrix} T_k & t_k \\ 0 & \lambda_k \end{bmatrix} \begin{bmatrix} \Psi_k^T \\ \psi_k^T \end{bmatrix}, \tag{3.69}$$

where λ_k is the unbounded eigenvalue with the left eigenvector ψ_k^T, T_k is an upper triangular matrix whose diagonal elements are the remaining eigenvalues, and $[\Psi_k \ \psi_k]$ is a unitary matrix.

From (3.45) and (3.67), since Σ_{k1} is bounded, $V_{k1}^T A_k V_{k1}$ is also bounded by Lemma 3.16. Consequently, it follows that $\psi_k^T V_{k1} = 0$ from (3.69). It follows that $\psi_k^T U = 0$, which contradicts that (A_k, U) is controllable. This completes the proof. □

Lemma 3.18. *Under the hypotheses of Theorem 3.9, $G_k \succ 0$ for all $k \geq 0$ if and only if A_0 is Hurwitz.*

Proof. First, it is proven that if A_0 is Hurwitz, then $G_k \succ 0$ for all $k \geq 0$. From Lemma 3.7 (iii), $\nu(P_0) = \pi(A_0) = 0$. From Lemma 3.10, $\nu(P_k) = \pi(A_k) = 0$ for all $k \geq 0$, and therefore $\nu(G_k) = \nu(P_{k+1}) - \nu(P_k) = 0$ for all $k \geq 0$.

To prove that if $G_k \succ 0$ for all $k \geq 0$, then A_0 is Hurwitz, consider the contrapositive: if A_0 is not Hurwitz, then there exists a finite integer $N \geq 0$ satisfying $\nu(G_N) > 0$. Conversely, suppose that $G_k \succ 0$ for all $k \geq 0$. Because A_0 is not Hurwitz, it follows that

$$\nu(P_0) = \pi(A_0) > 0, \tag{3.70}$$

from Lemma 3.7 (iii). Then, $\nu(P_k) = \nu(P_0) > 0$ for all $k \geq 0$ from Lemma 3.10. Also, $D_k \succeq 0$ from $G_k \succ 0$ and (3.53). From the update law (3.64), the congruent

matrix $\bar{G}_k \succ 0$ from (3.63). From (3.64), both $\bar{G}_k$ and $\sum_{i=0}^{k} U^T D_i U$ converge to a positive definite matrix and a positive semidefinite matrix, respectively. It follows that $U^T P_k U$ also converges from (3.53). Since (A_k, U) is controllable [62, Theorem 1.1], P_k is also bounded by Lemma 3.17, and therefore the decreasing sequence of P_k converges as $\lim_{k\to\infty} P_k = P_0 - \sum_{i=0}^{\infty} D_i =: \bar{P}$, which is followed by

$$\lim_{k\to\infty} D_k = \lim_{k\to\infty} (K_{k+1} - K_k)^T G_k (K_{k+1} - K_k) = 0. \tag{3.71}$$

From (3.46) with bounded $\|K_k\|_F$ and P_k, it follows that $\|W_k\|_F$ is also bounded. Hence, G_k is bounded by (3.47), which implies that $\lim_{k\to\infty} (K_{k+1} - K_k) = 0$ by (3.71). Using $\|K_k\|_F$ is bounded, it can be concluded that $\lim_{k\to\infty} K_k = \bar{K}$. However, from Lemma 3.14, $\bar{K}$ must be $R^{-1}B^T\bar{P}$ where $\bar{P}$ is the positive definite solution of $\mathcal{R}(P) = 0$, contrary to (3.70). □

Remark 3.19. Because the iteration, (3.42) and (3.43), starts with an arbitrary initial gain K_0, Lemma 3.18 states that for any $k_i \geq 0$, if A_{k_i} is Hurwitz, then $G_k \succ 0$ for all $k \geq k_i$. And conversely, if A_{k_i} is not Hurwitz, then there exists a finite integer $N_i \geq k_i$ such that G_{N_i} is a symmetric non-positive definite matrix, or equivalently $\nu(G_{N_i}) \geq 1$.

Finally, the proof of Theorem 3.9 is given below.

Proof of Theorem 3.9. The first step of the proof shows that for any K_0, there exists a finite integer $N \geq 0$ such that $P_k \succ 0$, $G_k \succ 0$ and A_k is Hurwitz for all $k \geq N$. If $A_0 = A - BK_0$ is Hurwitz, $P_k \succ 0$ and A_k is Hurwitz for all $k \geq 0$ by Lemma 3.10, and therefore $N = 0$. If A_0 is not Hurwitz, $\pi(A_0) > 0$ and there exists an integer $N_0 \geq 0$ such that G_{N_0} is not positive definite by Lemma 3.18. Because G_{N_0} is nonsingular by $s_k \in \mathcal{S}_k$, it follows that $\nu(G_{N_0}) \geq 1$.

By Lemma 3.10, $\pi(A_k)$ is strictly decreasing between the steps N_0 and N_0+1 as $\pi(A_{N_0+1}) = \pi(A_{N_0}) - \nu(G_{N_0}) < \pi(A_{N_0})$. By induction, it can be concluded that there exists an integer $N \geq N_0$ such that $\pi(A_N) = 0$, or equivalently A_N is Hurwitz since $\delta(A_N) = 0$ from Lemma 3.7 (i). Similarly, it follows that $P_k \succ 0$, $G_k \succ 0$, and A_k is Hurwitz for all $k \geq N$ by $s_k \in \mathcal{S}_k$ and Lemma 3.18.

Next, it is proven that $P_k \succeq P_{k+1} \succeq P^*$, $\lim_{k\to\infty} K_k = K^*$, and $\lim_{k\to\infty} P_k = P^*$ for all $k \geq N$. Since $G_k \succ 0$ for all $k \geq N$, it follows that $P_k \succeq P_{k+1} \succ 0$ from (3.53), and therefore $\lim_{k\to\infty} P_k =: \bar{P} \succ 0$. It follows that $\lim_{k\to\infty} K_k = R^{-1}B^T\bar{P} =: \bar{K}$, similar to the analysis in the proof of Lemma 3.18. Consequently, $\bar{P} = P^*$ and $\bar{K} = K^*$ by Lemma 3.14, which completes the proof. $\square$

3.5 Illustrative Numerical Examples

3.5.1 Validation of the Extended Kleinman Iteration

In this section, the extended Kleinman iteration is validated using a linearized model of the short-period dynamics of AFTI/F-16 aircraft which has unstable short period mode [66, Example 5.2-3]. The system matrices of the linear system

$$\dot{x} = Ax + Bu \tag{3.72}$$

are given by

$$A = \begin{bmatrix} -1.341 & 0.9933 & 0 & -0.1689 & -0.2518 \\ 43.223 & -0.8693 & 0 & -17.251 & -1.5766 \\ 1.341 & 0.0067 & 0 & 0.1689 & 0.2518 \\ 0 & 0 & 0 & -20 & 0 \\ 0 & 0 & 0 & 0 & -20 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 20 & 0 \\ 0 & 20 \end{bmatrix}. \tag{3.73}$$

The weight matrices for the linear quadratic regulator are defined as

$$Q = I_5, \quad R = I_2. \tag{3.74}$$

Because A has two unstable eigenvalues at 0 and 5.4514, simply choosing $K_0 = 0$ cannot satisfy $\sigma(A_0) \cap \sigma(-A_0) = \varnothing$. Instead, an arbitrary initial $K_0 \neq 0$ is chosen as follows:

$$K_0 = \begin{bmatrix} 8.4 & -5.1 & -4.1 & -1.1 & 0.4 \\ 5.8 & -7.4 & 3.1 & -5.2 & -5.1 \end{bmatrix}, \tag{3.75}$$

which is one of the extreme cases in which all the eigenvalues of $A - BK_0$ have positive real parts.

Three algorithms are compared: the Kleinman iteration, the extended Kleinman iteration, and the hybrid approach discussed in Remark 3.12. All three algorithms use the same initial gain K_0 in (3.75). The hybrid approach uses the proposed algorithm until $\nu(P_k) = 0$ and switches to the Kleinman iteration to exploit the quadratic convergence rate of the Newton method. The design parameter s_0 is simply set to be 1, which is not an eigenvalue of $A - BK_0$.

The convergence history of P_k and K_k to their optimal values, P^* and K^*, respectively, for each algorithm is presented in Fig. 3.1. The Kleinman iteration does not converge to the optimal stable solution, while the proposed methods converge. Because the extended Kleinman iteration ensures that the feedback gain becomes stable in a finite number of iterations, switching to the Kleinman iteration shows much faster convergence. The history of $\pi(A_k)$ for each algorithm is shown in Fig. 3.2. It can be confirmed that P_k does not monotonically converge to P^* when A_k is unstable under the proposed algorithms. But after a few steps, the closed-loop system eventually becomes stable, or equivalently, $\pi(A_k) = 0$.

3.5.2 Validation of the Data-Driven Surrogate Q-Learning

The extended Kleinman iteration can be identical to the surrogate Q-learning if the system matrices are available, as discussed in Section 3.2.2. When there is no prior knowledge of the system, the data-driven surrogate Q-learning proposed in Section 3.2.3 can be utilized with a dataset $\mathcal{D} = \{(x_i, u_i, \dot{x}_i)\}$ obtained from the system. However, the dataset may be prone to corruption due to noise, especially in the state derivative x_i. To demonstrate the efficacy of the proposed data-driven surrogate Q-learning, numerical simulations are conducted by in-

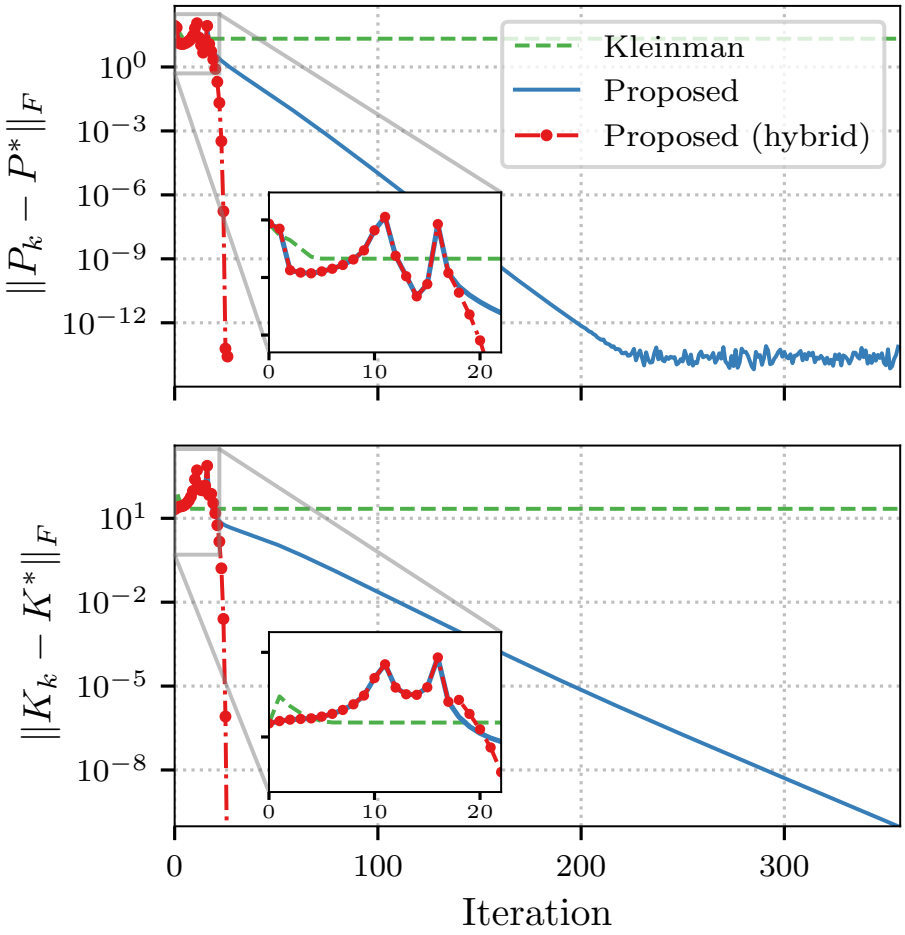

Figure 3.1 Convergence history of P_k and K_k to their optimal values.

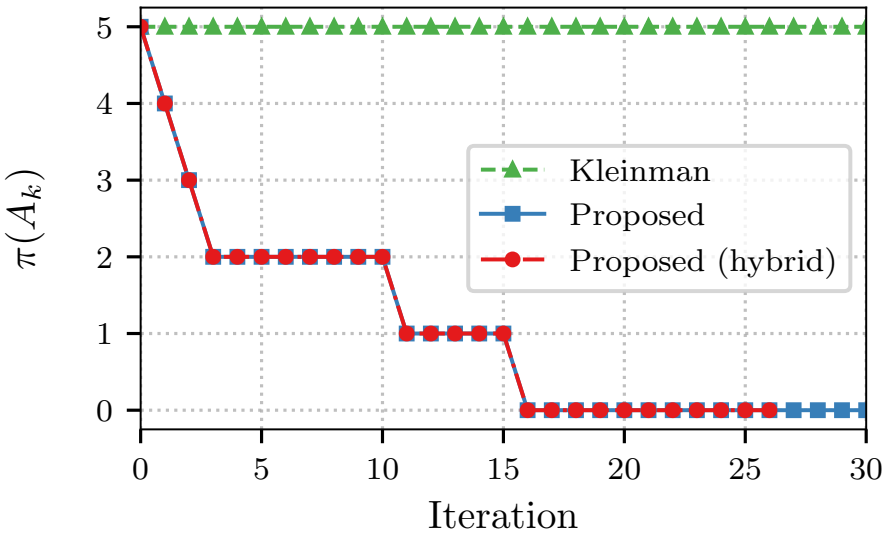

Figure 3.2 The convergence history of the number of eigenvalues of A_k with positive real parts.

troducing noise into the dataset and validating the algorithm.

The same short-period model and initial unstable feedback gain K_0 in Section 3.5.1 are used. The state x_i and the control input u_i are uniformly sampled from the set $[-3,3]^5 \times [-3,3]^2$, and the state derivative $\dot{x}_i$ is sampled as

$$\dot{x}_i \sim \mathcal{N}(Ax_i + Bu_i, 0.1), \tag{3.76}$$

where $\mathcal{N}$ denotes the normal distribution. Total 1,000 data tuples are collected for the dataset. For comparison of algorithms, an off-policy data-driven ADP is used [2, Algorithm 2.3.10] to represent the Kleinman iteration. The value of the design parameter for the data-driven surrogate Q-learning is set to $s = 1$. The hybrid approach switches to the data-driven surrogate Q-learning if $\pi(A_k) = 0$ and $\|K_k - K_{k-1}\|_F < 10^{-5}$.

Figures 3.3 and 3.4 present the convergence history of P_k and K_k as well as the history of $\pi(A_k)$ for each algorithm. An important consideration in the Kleinman iteration-based ADP method is the stability of the initial gain; if the initial gain is unstable, the method may fail to converge or stabilize, rendering the approach ineffective. In contrast, the proposed data-driven surrogate Q-learning is demonstrated to be robust to moderate noise and converges to a stabilizing feedback gain. In addition, the results obtained from the hybrid approach indicate that the extended Kleinman iteration proposed in this study is more robust to noise compared to the conventional Kleinman iteration.

However, it has been observed that when a certain level of noise is present, neither the existing Kleinman iteration-based ADP nor the proposed data-driven surrogate Q-learning methods can converge to optimal controllers or stabilize the control gains. The main reason is that the Gaussian noise applied to the derivative of the state variables follows a different distribution when solv-

ing the Lyapunov equation in the policy evaluation stage, which makes noise removal through pseudo-inverse less effective.

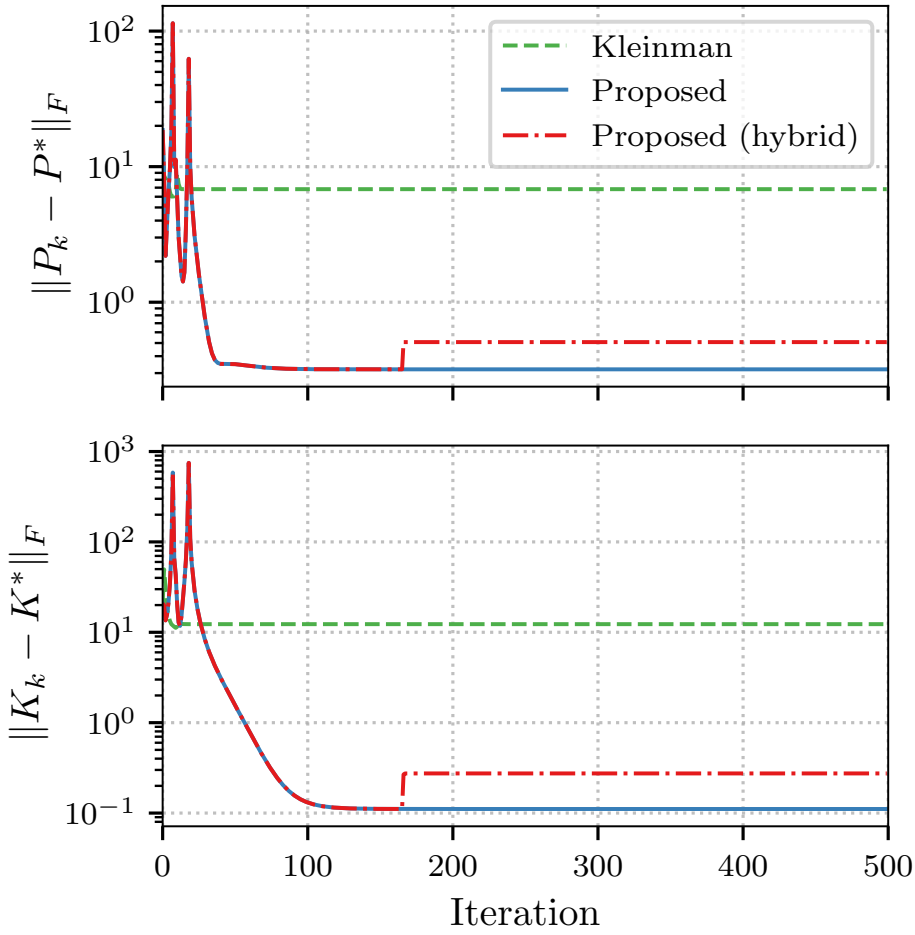

Figure 3.3 Convergence history of P_k and K_k to their optimal values.

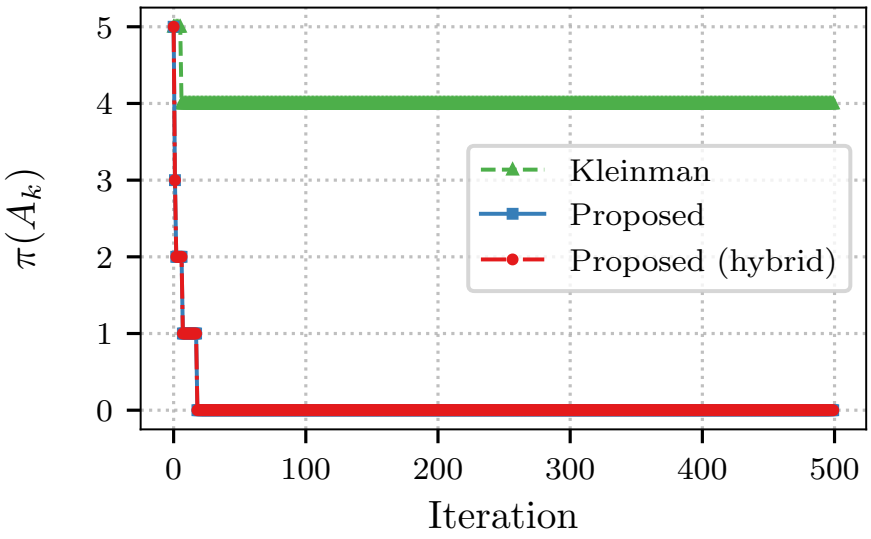

Figure 3.4 The convergence history of the number of eigenvalues of A_k with positive real parts.

Chapter 4

Application to Nonlinear Optimal Control Problems

In this chapter, the proposed surrogate Q-learning is applied to solve the infinite-horizon optimal control problems of nonlinear systems. The convergence proof of the extended Kleinman iteration is relied on linear algebra, and therefore the extension of the algorithm for the application of nonlinear systems is not straightforward. To overcome this difficulty, the Koopman operator theory is utilized. The *Koopman lifting linearization* can transform the nonlinear system into a linear system using nonlinear mappings called the *lifting*. Recently, various linear control syntheses for the Koopman lifting linearized system were proposed. However, not much research has been done on the controllability and observability of the Koopman lifting linearized system, which are sufficient conditions to apply the linear optimal control theory.

In this chapter, several conditions of the lifting are first provided for the controllability and observability of the Koopman lifting linearized system to apply the proposed surrogate Q-learning. Finding such lifting is very difficult in general, and therefore a meta-learning framework is proposed to train deep neural networks representing the lifting.

4.1 Nonlinear Optimal Control Problems

Consider a class of nonlinear dynamic systems with an affine control input given by

$$\dot{x} = f(x) + G(x)u, \tag{4.1}$$

where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^m$ is the control input, and the functions $f \in \mathcal{C}^1(\mathbb{R}^n)$ and $G \in \mathcal{C}^1(\mathbb{R}^n, \mathbb{R}^{n\times m})$. Without loss of generality, assume that $f(0) = 0$, which implies that $(x, u) = (0, 0)$ is an equilibrium point of (4.1). It is assumed that the nonlinear system (4.1) is controllable [67, Definition 11.1], which means that for any $x_0, x_1 \in \mathbb{R}^n$, there exists a control input u that steers the state from x_0 to x_1 in a finite time.

The design objective is to find an optimal control input function u_o^* that minimizes a performance index or a value function for the system in (4.1). The value function of a state variable $x_0 \in \mathbb{R}^n$ is defined by

$$V_o(x_0; u) = \int_0^\infty \left(y_o(t)^T y_o(t) + u(t)^T R u(t) \right) \mathrm{d}t\,, \tag{4.2}$$

where $R \in \mathbb{S}^m_{++}$,

$$y_o(t) = h(x(t)) \in \mathbb{R}^q \tag{4.3}$$

denotes the performance output, and the function $h : \mathbb{R}^n \to \mathbb{R}^q$ satisfies $h(0) = 0$. The state trajectory $x(t)$ follows (4.1) with the initial state $x(0) = x_0$ and the control input function $u : [0, \infty) \to \mathbb{R}^m$. Assume that the nonlinear system (4.1) is zero-state observable [68, Definition 6.5] with respect to the output $y_o(t)$, which means that for $u \equiv 0$, the output trajectory $y_o \equiv 0$ implies $x \equiv 0$.

It is further assumed that there exists an optimal control input u_o^* that is

the solution to the following optimal control problem:

$$u_o^* = \arg\inf_u V_o(x; u) \tag{4.4}$$

for all $x \in \mathbb{R}^n$. Because $R \in \mathbb{S}_{++}^m$ in (4.2), the optimal control input can be represented by a state-feedback form as

$$u_o^*(x) = -\frac{1}{2} R^{-1} G(x)^T \boldsymbol{\nabla} V_o^*(x) \tag{4.5}$$

with a slight abuse of notation, and $V_o^* \in \mathcal{C}^1(\mathbb{R}^n, \mathbb{R})$ is a solution to the Hamilton-Jacobi-Bellman (HJB) equation given by

$$0 = y_o^T y_o + u_o^*(x)^T R u_o^*(x) + \boldsymbol{\nabla} V_o^*(x)^T (f(x) + G(x) u_o^*(x)) \tag{4.6}$$

for all $x \in \mathbb{R}^n$ with a boundary condition $V_o^*(0) = 0$. The function V_o^* is indeed the optimal value function as

$$V_o^*(x) = V_o(x; u_o^*) = \min_u V_o(x; u) \tag{4.7}$$

for all $x \in \mathbb{R}^n$ [54].

4.2 Koopman Operators for Optimal Control Problems

As described in Section 4.1, the nonlinear optimal control input can be found as (4.5) when a solution V_o^* to the HJB equation (4.6) is obtained. However, it is difficult to find the solution to the HJB equation, because it is a nonlinear partial differential equation. Several studies reported workaround methods using a linear optimal control for a finite-dimensional linear system obtained by the Koopman operator [30].

This section provides mathematically rigorous conditions for constructing a linear system of which the linear optimal control is identical to the nonlinear optimal control in (4.5). Moreover, a sufficient condition is introduced to ensure the controllability and observability of the linear system.

4.2.1 Koopman Lifting Linearization

Consider a nonlinear autonomous system

$$\dot{x} = f(x), \tag{4.8}$$

which is the system (4.1) with $u \equiv 0$. Assume that there exists a N-dimensional invariant subspace of the infinitesimal generator of the Koopman operator for the system (4.8) [41].

Definition 4.1 (The lifting)**.** Given an infinitesimal generator of the Koopman operator for the nonlinear autonomous system (4.8), suppose that there exist $\phi_i \in \mathcal{F}$, $i = 1, \ldots, N$, spanning the invariant subspace of the infinitesimal generator. Then, a vector-valued function

$$\phi(x) = [\phi_1(x), \ldots, \phi_N(x)]^T \in \mathbb{R}^N \tag{4.9}$$

of $x \in \mathbb{R}^n$ is called the *lifting* of (4.1), if $\boldsymbol{\nabla}\phi(x)^T \in \mathbb{R}^{N\times n}$ is injective for all $x \in \mathbb{R}^n$.

If there exists a lifting of (4.1), then the elements of the lifting are observables of the infinitesimal generator satisfying (2.9). Therefore, there exists a matrix $A \in \mathbb{R}^{N\times N}$ such that [41]

$$\frac{\mathrm{d}}{\mathrm{d}t}\phi(x(t)) = A\phi(x(t)) = \boldsymbol{\nabla}\phi(x)^T f(x(t)), \tag{4.10}$$

where $x(t)$ is the state trajectory of (4.8) with any initial state $x(0) \in \mathbb{R}^n$. Note that the dynamics $f(x)$ can be recovered from $\frac{\mathrm{d}}{\mathrm{d}t}\phi(x)$ because $\boldsymbol{\nabla}\phi(x)^T$ is injective.

Consider a linear system of a state vector $z \in \mathbb{R}^N$ with the same system matrix A in (4.10) given by

$$\dot{z} = Az. \tag{4.11}$$

The above linear system yields the state trajectory $z(t) = \phi(x(t))$ for all $t \geq 0$ if and only if $z(0) = \phi(x(0))$. In other words, the state $z(t)$ of (4.11) with an arbitrary initial state $z(0) = z_0 \in \mathbb{R}^N$ may not necessarily satisfy $z(t) = \phi(x)$ for any $x \in \mathbb{R}^n$. However, it is still useful to analyze the state $z(t)$ of the linear system (4.11) to describe the nonlinear behavior of the state $x(t)$ of the original system (4.8).

Definition 4.2 (The Koopman lifting linearization)**.** The *Koopman lifting linearization* of (4.1) is defined by a linear system

$$\dot{z} = Az + Bu, \tag{4.12}$$

where $z \in \mathbb{R}^N$ is the lifted state vector, and $A \in \mathbb{R}^{N\times N}$ and $B \in \mathbb{R}^{N\times m}$, if the lifted state trajectory $z(t)$ satisfies $z(t) = \phi(x(t))$; $\phi(x)$ is a lifting of (4.1) and

$x(t)$ is the state trajectory of (4.1) with the initial state $x(0) \in \mathbb{R}^n$ and any control input function $u(t)$.

The linear control affine term Bu in (4.12) is necessary to facilitate linear optimal control theories while it requires an additional condition for the lifting $\phi(x)$.

Proposition 4.3 ([41])**.** *Given a nonlinear system* (4.1) *and a corresponding lifting* $\phi(x)$*, the dynamics of* $z(t) = \phi(x(t))$ *can be represented by the Koopman lifting linearization* (4.12) *if and only if* $\boldsymbol{\nabla}\phi(x)^T G(x) \in \mathbb{R}^{N\times m}$ *is constant for all* $x \in \mathbb{R}^n$*.*

4.2.2 Equilibrium Points

Suppose that (x_e, u_e) is an equilibrium point of (4.1) such that $f(x_e) + G(x_e)u_e = 0$. Subtracting it from (4.1) yields

$$\dot{\tilde{x}} = f(x) - f(x_e) + G(x)u - G(x_e)u_e = \tilde{f}(\tilde{x}) + \tilde{G}(\tilde{x})\tilde{u}, \tag{4.13}$$

where $\tilde{f}(\tilde{x}) \coloneqq f(\tilde{x} + x_e) - f(x_e) + G(\tilde{x} + x_e)u_e - G(x_e)u_e$, $\tilde{G}(\tilde{x}) \coloneqq G(\tilde{x} + x_e)$, $\tilde{x} \coloneqq x - x_e$, and $\tilde{u} \coloneqq u - u_e$. On the other hand, if $u(t) \equiv u_e$ and $x(0) = x_e$, then $x(t) \equiv x_e$ and $z(t) \equiv \phi(x_e)$ for all $t \geq 0$, and therefore $\dot{z} \equiv 0$. It follows from (4.12) that $A\phi(x_e) + Bu_e = 0$, which implies

$$\dot{\tilde{z}} = A\tilde{z} + B\tilde{u} \tag{4.14}$$

from (4.12), where $\tilde{z}(t) \coloneqq \phi(x) - \phi(x_e)$. Since the Koopman operator is linear, if $\phi(x)$ is a lifting of (4.1), then $\phi(x) - \phi(x_e)$ is also a lifting of (4.1). By defining $\tilde{\phi}(\tilde{x}) = \phi(\tilde{x} + x_e) - \phi(x_e)$, the dynamics of $\tilde{z} = \tilde{\phi}(\tilde{x})$ can be represented by (4.14) by Proposition 4.3 using the fact that

$$\boldsymbol{\nabla}\tilde{\phi}(\tilde{x})^T \tilde{G}(\tilde{x}) = \boldsymbol{\nabla}\phi(\tilde{x} + x_e)^T G(\tilde{x} + x_e) = \boldsymbol{\nabla}\phi(x)^T G(x), \tag{4.15}$$

which is also constant. Therefore, in the rest of this study, it is assumed without loss of generality that the equilibrium of the nonlinear system in (4.1) is at the origin, or equivalently, $f(0) = 0$.

4.2.3 Lifted Optimal Control Problems

This section provides a rigorous theoretical proof that the optimal controller obtained using the Koopman lifting linearized system (4.12) is indeed the nonlinear optimal controller (4.5) for the original nonlinear system (4.1) under mild assumptions for constructing the lifting $\phi(x)$.

The lifted performance index for the Koopman lifting linearized system (4.12) corresponding to (4.2) is defined as

$$V(z_0; u) = \int_0^\infty \left(z(t)^T Q z(t) + u(t)^T R u(t) \right) \mathrm{d}t \,, \tag{4.16}$$

where $z(0) = z_0$, the matrix R is defined in (4.2), and the matrix $Q \in \mathbb{S}_+^N$ satisfies $Q = C^T C$ for a matrix C satisfying the assumption given below.

Assumption 4.4. Given a lifting $\phi(x)$ of (4.1), there exists a matrix $C \in \mathbb{R}^{q \times N}$ such that $C\phi(x) = h(x)$ for all $x \in \mathbb{R}^n$ with $h(x)$ defined in (4.3).

Under Assumption 4.4, consider the performance output of the lifted performance index (4.16) defined by

$$y(t) = Cz(t) \tag{4.17}$$

for all $t \geq 0$. If $z(0) = \phi(x(0))$, then $y(t) = y_o(t)$ for all $t \geq 0$. However, $y(t) \neq y_o(t)$ in general, which implies that the zero-state observability of the original nonlinear system (4.1) with the performance output (4.3) does not

imply the observability of the Koopman lifting linearized system (4.12), or equivalently, a matrix pair (A, C).

The optimal control input u^* of the system (4.12) minimizing the performance index (4.16) can be obtained by solving the following lifted optimal control problem:

$$u^* = \arg\inf_u V(z_0; u) \tag{4.18}$$

for all $z_0 \in \mathbb{R}^N$. As discussed in Section 4.2.1, the state trajectory $z(t)$ of (4.12) may not be relevant to $x(t)$ unless $z_0 = \phi(x(0))$, and moreover the optimal control input u^* may not exist for all $z_0 \in \mathbb{R}^N$ even if u_o^* in (4.5) exists for all $x_0 \in \mathbb{R}^n$. The following proposition states that if there exists u^*, then the nonlinear optimal control u_o^* for any initial state $x_0 \in \mathbb{R}^n$ can be obtained using (4.12) with the initial state $z_0 = \phi(x_0)$.

Proposition 4.5. *Suppose that there exists an optimal control u^* that solves the lifted optimal control problem* (4.18) *for all $z_0 \in \mathbb{R}^N$. Then, for any $x_0 \in \mathbb{R}^n$, the optimal control u_o^* of* (4.4) *satisfies $u_o^* = u^*$ if $z_0 = \phi(x_0)$.*

Proof. Suppose that $u^* \neq u_o^*$. Then,

$$V_o(x_0; u_o^*) < V_o(x_0; u^*) \tag{4.19}$$

from (4.4). Because $z(0) = \phi(x_0)$, it follows that $z(t) = \phi(x(t))$, which implies

$$z(t)^T Q z(t) = \phi(x(t))^T C^T C \phi(x(t)) = y(t)^T y(t) \tag{4.20}$$

by Assumption 4.4. Therefore, from (4.2) and (4.16), it can be concluded that $V(z_0; u) = V_o(x_0; u)$ for any u. From (4.19), it follows that

$$V(z_0; u_o^*) = V_o(x_0; u_o^*) < V_o(x_0; u^*) = V(z_0; u^*), \tag{4.21}$$

which contradicts (4.18). □

The infinite-horizon optimal control u^* for (4.18) is indeed a linear quadratic regulator for the linear system (4.12) and the lifted performance function with the quadratic cost as defined in (4.16). Therefore, a sufficient condition for the existence of u^* is that a matrix pair (A, B) is controllable and a matrix pair (A, C) is observable. As discussed above, however, it is not clear to guarantee that the Koopman lifting linearized system is controllable and observable for an arbitrary lifting $\phi(x)$ of (4.1).

The following two lemmas provide an equivalent condition of $\phi(x)$ to the controllability of (A, B) and a sufficient condition of $\phi(x)$ for the observability of (A, C).

Lemma 4.6. *Suppose that the nonlinear system* (4.1) *is controllable. Then, the Koopman lifting linearized system* (4.12) *is controllable if and only if the lifting* $\phi(x)$ *of* (4.1) *is surjective.*

Proof. First, it is shown that if the lifting $\phi(x)$ of (4.1) is surjective, then a matrix pair (A, B) of (4.12) is controllable. For any $z_0, z_1 \in \mathbb{R}^N$, there exist $x_0, x_1 \in \mathbb{R}^n$ such that

$$z_0 = \phi(x_0), \quad z_1 = \phi(x_1). \tag{4.22}$$

Because the nonlinear system (4.1) is controllable, there exists a control input $u_1(t)$ satisfying $x(0) = x_0$ and $x(t_1) = x_1$ for some $t_1 \geq 0$. It follows from (4.22) that the same control input $u_1(t)$ steers the state of (4.12) from $z(0) = \phi(x(0)) = z_0$ to $z(t_1) = \phi(x(t_1)) = z_1$, which implies that the system (4.12) is controllable.

To show the converse, consider an arbitrary vector $z_2 \in \mathbb{R}^N$. Because the system (4.12) is controllable, for any $x_0 \in \mathbb{R}^n$, there exists a control input

$u_2(t)$ satisfying $z(0) = \phi(x_0)$ and $z(t_2) = z_2$ for some $t_2 > 0$. Let $x(t)$ be the state trajectory of (4.1) using the same control input $u_2(t)$ with an initial state $x(0) = x_0$, and let $x_2 = x(t_2) \in \mathbb{R}^n$. Since $\phi(x(0)) = z(0)$ by construction, it follows that $z_2 = z(t_2) = \phi(x(t_2)) = \phi(x_2)$. Therefore, it can be concluded that for any $z_2 \in \mathbb{R}^N$, there exists $x_2 \in \mathbb{R}^n$ such that $\phi(x_2) = z_2$, which completes the proof. □

Lemma 4.7. *Suppose that the nonlinear system* (4.1) *is zero-state observable with the output* (4.3)*, the corresponding lifting* $\phi(x)$ *is surjective, and Assumption 4.4 holds. Then, the Koopman lifting linearized system* (4.12) *is observable with the output* (4.17) *if and only if* $\phi(0) = 0$.

Proof. First, it is shown that if $\phi(0) = 0$, then a matrix pair (A, C) of (4.12) is observable. Suppose that (A, C) is not observable. Then, a matrix-valued function of $t \geq 0$ defined by

$$W_O(t) = \int_0^t e^{A^T\tau} C^T C e^{A\tau} \, \mathrm{d}\tau \tag{4.23}$$

satisfies that $W_O(t_1)$ is singular for some $t_1 > 0$ [51, Theorem 6.4]. It follows that there exists a vector $z_0 \neq 0 \in \mathbb{R}^N$ such that $z_0^T W_O(t_1) z_0 = 0$, which implies that

$$Ce^{At} z_0 = 0, \quad \forall t \in [0, t_1]. \tag{4.24}$$

Because $\phi(x)$ is surjective and $\phi(0) = 0$, for the nonzero vector z_0, there exists $x_0 \neq 0 \in \mathbb{R}^n$ such that $\phi(x_0) = z_0$. Then, with a control input $u \equiv 0$, the state $z(t)$ of (4.12) with an initial state $z(0) = z_0$ satisfies $z(t) = e^{At} z_0 = \phi(x(t))$ for all $t \in [0, t_1]$. From (4.24) and Assumption 4.4, it follows that $y(t) = C\phi(x(t)) = y_o(t) = 0$ for all $t \in [0, t_1]$, which contradicts the zero-state observability of the original nonlinear system.

To show the converse, suppose that (A, C) is observable but $\phi(0) = z_0 \neq 0$. Since $x \equiv 0$ if $u \equiv 0$ and $x(0) = 0$ from (4.12), it follows that $z(t) = z_0$, thus $\dot{z}(t) = 0$ for all $t \geq 0$. Therefore, it can be concluded that $z_0 \in \ker(A)$. On the other hand, from Assumption 4.4 and the definition of h in (4.3), it follows that

$$Cz_0 = C\phi(0) = h(0) = 0, \tag{4.25}$$

which implies $z_0 \in \ker(C) \cap \ker(A)$. This contradicts that (A, C) is observable, which completes the proof. □

Now, it is clear that the sufficient condition of $\phi(x)$ for the controllability and observability of (4.12) is that $\phi(x)$ is surjective. If this is the case and the matrices A and B are known, the unique optimal control input u^* can be found using the ARE [54].

The next theorem summarizes the conditions for finding nonlinear optimal control input using the Koopman lifting linearization.

Theorem 4.8. *Suppose that the system* (4.1) *is controllable and zero-state observable with the performance output* $y_o(t) = h(x(t))$ (4.3)*, and that* $\mathcal{A}$ *is the infinitesimal generator of the Koopman operator for* (4.1)*. If there exist a mapping* $\phi : \mathbb{R}^n \to \mathbb{R}^N$ *and a matrix* $C \in \mathbb{R}^{q \times N}$ *such that*

C1: the mapping ϕ *is surjective,*

C2: the matrix $\boldsymbol{\nabla}\phi(x)^T$ *is injective for all* $x \in \mathbb{R}^n$*,*

C3: the mapping ϕ *spans the invariant subspace of* $\mathcal{A}$*,*

C4: $\boldsymbol{\nabla}\phi(x)^T G(x)$ *is constant for all* $x \in \mathbb{R}^n$*,*

C5: $\phi(0) = 0$*, and*

C6: $C\phi(x) = h(x)$ *for all* $x \in \mathbb{R}^n$*,*

then the optimal control u_o^* *in* (4.4) *is given by*

$$u_o^*(t) = -R^{-1}B^T P\phi(x(t)), \tag{4.26}$$

where $P \in \mathbb{S}_{++}^N$ *is the unique positive definite solution to the algebraic Riccati equation given by*

$$PA + A^T P + Q - PBR^{-1}B^T P = 0 \tag{4.27}$$

corresponding to a Koopman lifting linearization (4.12) *and the lifted optimal control problem* (4.18).

Proof. Since the lifting $\phi(x)$ is surjective and $\phi(0) = 0$, the matrix pairs (A, B) and (A, C) of (4.12) are controllable and observable, respectively, by Lemmas 4.6 and 4.7. Then, there exist a unique symmetric positive definite solution P to (4.27) and a unique optimal control u^* for (4.18), which has the linear state feedback form $u^*(t) = -R^{-1}B^T Pz(t)$ [54, Theorem 6.1]. Then, by Proposition 4.3, the optimal control u_o^* for (4.4) is identical to u^* with $z(0) = \phi(x(0))$, and from $z(t) = \phi(x(t))$, it can be concluded that u_o^* has the form in (4.26). □

4.3 The Meta-Learning Framework

In this section, a meta-learning framework is proposed to train a lifting of (4.1) to satisfy all of the conditions in Theorem 4.8 for a group of uncertain systems. In the Koopman lifting linearization (4.12), there are three components that define the system dynamics: the system matrices A, B, and the lifting $\phi(x)$. Because the Koopman operator can represent dynamic characteristics of systems, the group of uncertain systems is defined by a set of systems that share a common Koopman invariant subspace, but possibly have different system matrices. If the common lifting is known for the group, the model-free data-driven surrogate Q-learning algorithm proposed in Section 3.2 can be applied even when the system matrices and an initial admissible policy are uncertain for each system in the group. Therefore, the proposed framework can be categorized as a meta-learning method due to its strategy of identifying a common feature among a group of systems and utilizing it to quickly adapt to a new uncertain system within the same group.

4.3.1 Koopman Groups and Common Liftings

Definition 4.9 (Koopman groups)**.** A set of dynamical systems is said to be a *Koopman group* if there exists a common mapping, called the *common lifting* of the group, that satisfies the conditions C1–C6 in Theorem 4.8 for each system in the set.

Let $\mathcal{G}$ be the Koopman group of nonlinear systems, where the system $\mathcal{S}_p \in \mathcal{G}$ indexed by an integer p has the nonlinear dynamics similar to (4.1) as

$$\dot{x} = f_p(x) + G_p(x)u, \tag{4.28}$$

where $x \in \mathbb{R}^n$ is the state vector, and $u \in \mathbb{R}^m$ is the control input vector. Let $\phi : \mathbb{R}^n \to \mathbb{R}^N$ be the common lifting of the Koopman group $\mathcal{G}$. Because ϕ satisfies the conditions C2–C4 in Theorem 4.8 for all systems in $\mathcal{G}$, there exists a Koopman lifting linearization (4.12) for a system $\mathcal{S}_p \in \mathcal{G}$ by Proposition 4.3, which can be represented by

$$\frac{\mathrm{d}}{\mathrm{d}t}\phi(x) = A_p\phi(x) + B_p u, \tag{4.29}$$

where the constant matrices $A_p \in \mathbb{R}^{N\times N}$ and $B_p \in \mathbb{R}^{N\times m}$ depend on the system $\mathcal{S}_p$. Figure 4.1 illustrates the relationship between the Koopman group, the common lifting, and the systems in the group.

Given a Koopman group of uncertain systems, applying a known lifting $\phi(x)$ can effectively mitigate the associated uncertainties. Specifically, if the lifting is already known, the only sources of uncertainty for such systems would be the system matrices A_p and B_p. However, it is generally difficult to find such a common lifting. In following sections, a meta-learning framework is proposed to train deep neural networks that can approximate the common lifting, if it exists.

4.3.2 Diffeomorphic Lifting Approximation

The conditions C1 and C2 in Theorem 4.8 are sufficient for the controllability and observability of the Koopman lifting linearization in the Koopman group. These conditions are automatically satisfied with a special class of the lifting $\phi(x)$.

Proposition 4.10. *If a continuously differentiable mapping $\phi : \mathbb{R}^n \to \mathbb{R}^n$ is a global diffeomorphism, then it satisfies the conditions C1 and C2 in Theorem 4.8.*

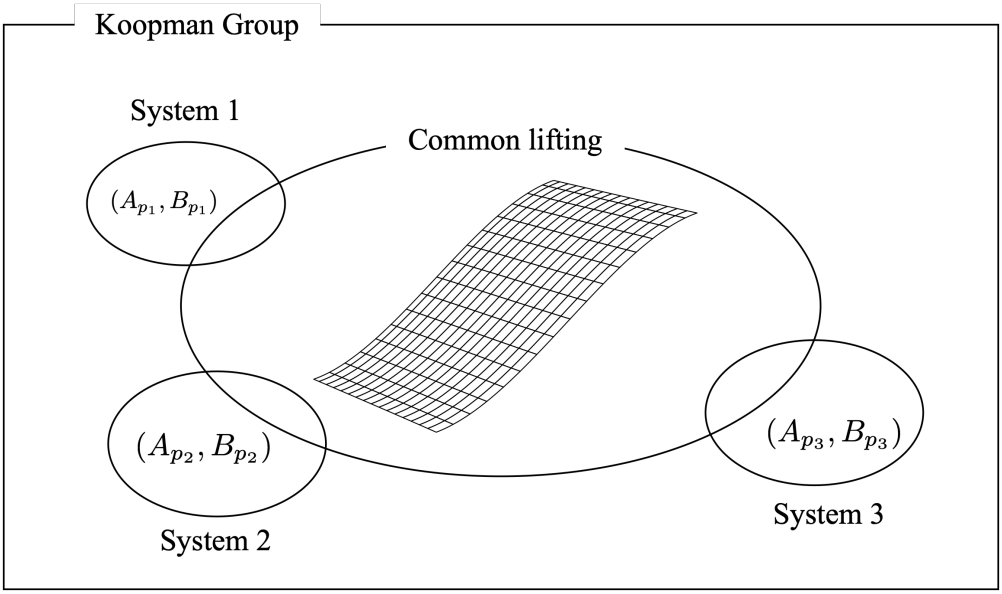

Figure 4.1 A diagram of a Koopman group.

Proof. The diffeomorphism ϕ is surjective because there is an inverse mapping by definition, and the Jacobian $\boldsymbol{\nabla}\phi(x)^T \in \mathbb{R}^{n\times n}$ is invertible, and thus injective for all $x \in \mathbb{R}^n$ [69]. □

The use of diffeomorphisms to represent the infinitesimal generators of Koopman operators has received increasing attention in recent research [70,71]. It has been shown that diffeomorphic liftings preserve the stability of the original autonomous nonlinear systems [72]. Bevanda et al. utilized invertible neural networks (INNs) [72], specifically the coupling flow-based INNs (CF-INNs) [73], to realize diffeomorphic liftings because CF-INNs can universally approximate diffeomorphisms [74].

In this study, the generative flow (Glow) [75] is employed to realize the diffeomorphic lifting. The Glow is a CF-INN with trainable convolution-based

permutations. The single-scale Glow approach is implemented using K-step Flows, with each Flow step consisting of an activation normalization layer, an 1×1 convolution layer with LU decomposition, and an affine coupling layer, which are all invertible. In addition to the output, each layer returns the log-determinant to calculate the probability density of the input data. The detailed implementation of the Glow can be found in Appendix A.

Let $\hat{\phi}(x; w_\phi)$ denote the Glow, which is used to approximate the common lifting $\phi(x)$, where w_ϕ is the network parameters to be trained. By denoting each Flow by $\hat{\phi}_i$ for $i = 1, \ldots, K$, the Glow can be written as

$$\hat{\phi} = \hat{\phi}_K \circ \cdots \circ \hat{\phi}_2 \circ \hat{\phi}_1. \tag{4.30}$$

And, the probability density of the input x, called the log-likelihood of x, is given by

$$\log p(x) = \log p(z) + \sum_{i=1}^{K} \log \left| \det \left(\frac{\partial z_i}{\partial z_{i-1}} \right) \right|, \tag{4.31}$$

where $z_i \coloneqq \hat{\phi}_i(z_{i-1}; w_{\phi_i})$ for $i = 0, \ldots, K$, $z \coloneqq z_K = \hat{\phi}(x)$, and $z_0 \coloneqq x$. The probability density function $p(z)$ is typically chosen as the probability density function of a simple distribution such as the Gaussian distribution $\mathcal{N}(0, 1)$. Maximizing the log-likelihood, or minimizing the negative log-likelihood, can be regarded as approximation of the data distribution in the dataset. In the context of the diffeomorphic lifting approximation, this implies that the resulting lifting maps the original state distribution to the predefined simple distribution in the lifted state space, which may help the data-driven surrogate Q-learning implementation in the lifted state space.

It follows from Proposition 4.10 that using the Glow for diffeomorphic lifting approximation guarantees the satisfaction of the conditions C1 and C2 in The-

orem 4.8. The remaining conditions C3–C6 are considered in the meta-learning framework. In particular, the base learner is formulated for the conditions C3 and C4, and the meta learner trains the network to satisfy the conditions C5 and C6.

4.3.3 Base Learner Formulation

Given a dataset $\mathcal{D}_p = \{(x_i, u_i, \dot{x}_i)\}_{i=1}^{n_p}$ acquired from a system $\mathcal{S}_p \in \mathcal{G}$, if there is a hand-crafted lifting $\phi(x)$ of $\mathcal{S}_p$, extended dynamic mode decomposition (EDMD) methods [28, 32] are commonly used to find A_p and B_p in (4.29). The EDMD method can be represented by the following optimization problem:

$$\min_{A_p, B_p} \frac{1}{n_p} \sum_{i=1}^{n_p} \left\| \boldsymbol{\nabla} \phi(x_i)^T \dot{x}_i - A_p \phi(x_i) - B_p u_i \right\|^2. \tag{4.32}$$

Several studies synthesized optimization problems using deep neural networks $\hat{\phi}(x; w_\phi)$ to approximate liftings as follows [46, 76]:

$$\min_{A_p, B_p, w_\phi} \frac{1}{n_p} \sum_{i=1}^{n_p} \left\| \boldsymbol{\nabla} \hat{\phi}(x_i; w_\phi)^T \dot{x}_i - A_p \hat{\phi}(x_i; w_\phi) - B_p u_i \right\|^2. \tag{4.33}$$

The above methods are designed to obtain a Koopman lifting linearization for a single system, $\mathcal{S}_p$. However, as discussed in Section 4.3.1, a meta-learning framework is proposed in this study to obtain a common lifting for a Koopman group. By applying the proposed framework to multiple systems within the group, a single diffeomorphism can be learned to represent the common lifting.

By observing the similarity between the closed-form base learner problem in Section 2.5.2 and the optimization problem for Koopman lifting linearization (4.33), the base learner problem is proposed as follows:

$$\min_{F_p} \mathcal{L}_{\text{base}}(\mathcal{D}_p; F_p, w_\phi) = \min_{A_p, B_p} \frac{1}{n_p} \sum_{i=1}^{n_p} \left\| \dot{\hat{\phi}}_i(w_\phi) - A_p \hat{\phi}_i(w_\phi) - B_p u_i \right\|^2, \tag{4.34}$$

where $F_p = [A_p, B_p] \in \mathbb{R}^{n\times(n+m)}$ and

$$\hat{\phi}_i(w_\phi) := \hat{\phi}(x_i; w_\phi), \tag{4.35}$$

$$\dot{\hat{\phi}}_i(w_\phi) := \boldsymbol{\nabla}\hat{\phi}(x_i; w_\phi)^T \dot{x}_i \tag{4.36}$$

for $i = 1, \ldots, n_p$.

Given a system $\mathcal{S}_p \in \mathcal{G}$, if there exists a network parameter w_ϕ such that

$$\min_{F_p} \mathcal{L}_{\text{base}}(\mathcal{D}_p; F_p, w_\phi) = 0 \tag{4.37}$$

for any dataset $\mathcal{D}_p$ acquired from the system $\mathcal{S}_p$, the network $\hat{\phi}(x; w_\phi)$ satisfies the conditions C3 and C4 in Theorem 4.8. Moreover, if the network is a diffeomorphism, it follows from Propositions 4.3 and 4.10 that there exists a Koopman lifting linearization (4.12) of $\mathcal{S}_p$ with

$$[A, B] = F_p^*(w_\phi) := \arg\min_{F_p} \mathcal{L}_{\text{base}}(\mathcal{D}_p; F_p, w_\phi). \tag{4.38}$$

Considering w_ϕ as a meta-learner parameter, the optimization problem (4.34) has a closed-form solution as in Section 2.5.2, given by

$$F_p^*(w_\phi) = Y_p(w_\phi)\Psi_p(w_\phi)^\dagger =: \left[A_p^*(w_\phi), B_p^*(w_\phi)\right], \tag{4.39}$$

where

$$Y_p(w_\phi) = \begin{bmatrix} \dot{\hat{\phi}}_1(w_\phi) & \cdots & \dot{\hat{\phi}}_{n_p}(w_\phi) \end{bmatrix}, \tag{4.40a}$$

$$\Psi_p(w_\phi) = \begin{bmatrix} \hat{\phi}_1(w_\phi) & \cdots & \hat{\phi}_{n_p}(w_\phi) \\ u_1 & \cdots & u_{n_p} \end{bmatrix}, \tag{4.40b}$$

which implies that it can be used as the closed-form base-learner problem (2.31) to obtain the approximated common lifting for the Koopman group within a meta-learning framework.

4.3.4 Meta-Learner Formulation

The overarching purpose of the meta learner is to obtain the common lifting that not only allows the Koopman lifting linearization for each system of the Koopman group but also ensures that the linearized system is controllable and observable with the output $y(t)$ defined in (4.17).

Suppose that the p-th task $\mathcal{T}_p$ of the meta-learning problem is defined as the dataset $\mathcal{D}_p$ obtained from the system $\mathcal{S}_p \in \mathcal{G}$, because the loss functions for the base learner are the same for all systems as defined in (4.34). Let $p(\mathcal{G})$ be the distribution of the task $\mathcal{T}_p = \mathcal{D}_p$ in the Koopman group $\mathcal{G}$. The meta-learner problem (2.29) is formulated as follows:

$$\min_{w_\phi} \mathbb{E}_{\mathcal{D}_p \sim p(\mathcal{G})} \big[\mathcal{L}_{\text{meta}}\big(\mathcal{D}_p; F_p^*(w_\phi), w_\phi\big)\big], \tag{4.41}$$

where $\mathcal{L}_{\text{meta}}$ denotes the meta-learner loss function for the common lifting, w_ϕ is the parameter of the diffeomorphic lifting approximation network, which is regarded as the meta-learner parameters, and $F_p^*(w_\phi)$ is the closed-form solution of the base learner given in (4.39). The meta-learner loss $\mathcal{L}_{\text{meta}}$ is composed of four losses as

$$\mathcal{L}_{\text{meta}} = \eta_{\text{lin}}\mathcal{L}_{\text{lin}} + \eta_{\text{orig}}\mathcal{L}_{\text{orig}} + \eta_{\text{out}}\mathcal{L}_{\text{out}} + \eta_{\text{nll}}\mathcal{L}_{\text{nll}}, \tag{4.42}$$

where $\mathcal{L}_{\text{lin}}$ denotes the Koopman lifting linearization loss, $\mathcal{L}_{\text{orig}}$ denotes the origin loss, $\mathcal{L}_{\text{out}}$ denotes the output representation loss, $\mathcal{L}_{\text{nll}}$ denotes the negative log-likelihood loss, and $\eta_{(\cdot)}$ are the weights of the corresponding losses.

The Koopman lifting linearization loss $\mathcal{L}_{\text{lin}}$ encourages that a common lifting $\hat{\phi}(x; w_\phi)$ satisfies the conditions C3 and C4 in Theorem 4.8 for all systems in the Koopman group, defined by

$$\mathcal{L}_{\text{lin}}(\mathcal{D}_p; F_p^*(w_\phi), w_\phi) = \left\| F_p^*(w_\phi)\Psi_p(w_\phi) - Y_p(w_\phi) \right\|_F^2, \tag{4.43}$$

where $F_p^*(w_\phi)$ is the closed-form solution defined in (4.39), and the functions Ψ_p and Y_p are defined in (4.40). All of these functions depend on the dataset $\mathcal{D}_p$ which involves the dynamics information of $\mathcal{S}_p$.

The origin loss $\mathcal{L}_{\text{orig}}$ is designed to ensure the condition C5, defined by

$$\mathcal{L}_{\text{orig}}(w_\phi) = \left\| \hat{\phi}(0; w_\phi) \right\|^2, \tag{4.44}$$

which is independent of the dataset $\mathcal{D}_p$, thus it can be considered as regularization of the network $\hat{\phi}(x; w_\phi)$.

The output representation loss $\mathcal{L}_{\text{out}}$ is defined to enforce the condition C6 in Theorem 4.8. It is defined by using the closed-form solution $C^*(w_\phi)$ as

$$\mathcal{L}_{\text{out}}(\mathcal{D}_p; w_\phi) = \frac{1}{n_p} \sum_{i=1}^{n_p} \left\| h(x_i) - C^*(w_\phi)\hat{\phi}_i(w_\phi) \right\|^2, \tag{4.45}$$

where $C^*(w_\phi) \in \mathbb{R}^{q \times n}$ is defined by

$$C^*(w_\phi) = \begin{bmatrix} h(x_1) & \cdots & h(x_{n_p}) \end{bmatrix} \begin{bmatrix} \hat{\phi}_1(w_\phi) & \cdots & \hat{\phi}_{n_p}(w_\phi) \end{bmatrix}^\dagger. \tag{4.46}$$

Although the loss $\mathcal{L}_{\text{out}}$ includes the state x_i from the dataset $\mathcal{D}_p$, the dynamics information of $\mathcal{S}_p$ is not involved. In other words, the state x_i need not belong to the dataset $\mathcal{D}_p$ but may be any real vector in $\mathbb{R}^n$.

Along with a diffeomorphic lifting approximation $\hat{\phi}(x; w_\phi)$, the losses $\mathcal{L}_{\text{lin}}$, $\mathcal{L}_{\text{orig}}$ and $\mathcal{L}_{\text{out}}$ are designed to satisfy the conditions C1–C6 in Theorem 4.8. Therefore, the optimal control u_o^* is equivalent to the lifted optimal control u^* of the lifted optimal control problem, given in (4.18). Using the data-driven surrogate Q-learning method proposed in Chapter 3, the lifted optimal control can be learned using a dataset $\mathcal{D}_{\text{lift}} := \left\{ \left(\hat{\phi}_i, u_i, \dot{\hat{\phi}}_i \right) \right\}$. Therefore, the performance of the data-driven surrogate Q-learning is heavily dependent on the distribution

of the data points in $\mathcal{D}_{\text{lift}}$ in the sense of regression theory. The negative log-likelihood loss $\mathcal{L}_{\text{nll}}$ is introduced in this perspective, defined by

$$\mathcal{L}_{\text{nll}}(\mathcal{D}_p; w_\phi) = -\frac{1}{n_p}\sum_{i=1}^{n_p} \log p(x_i), \tag{4.47}$$

where the log-likelihood $\log p(x)$ is defined in (4.31).

4.3.5 Offline and Online Learning Synthesis

The reinforcement learning framework utilizing meta-learning proposed in this study is divided into two stages: offline learning and online learning. In offline learning, diffeomorphic lifting approximation is learned through meta-learning using a dataset obtained from any system belonging to the Koopman group, as discussed above. This dataset can be obtained through experiments or numerical simulations based on knowledge of the dynamics of similar systems. The advantage of the proposed meta-learning framework is that it can utilize data obtained by various controllers or tuned controllers in various environments. The learned diffeomorphic lifting approximation is then installed in the online learning system in the actual system. The parameters of the diffeomorphic lifting approximation are fixed in the online learning stage. The online learning requires actual data for the system being controlled. However, the proposed surrogate Q-learning can quickly learn the optimal controller with very little data, as described in Section 3.2.3, compared to general reinforcement learning algorithms. The overall procedure of online and offline learning is illustrated in Fig. 4.2.

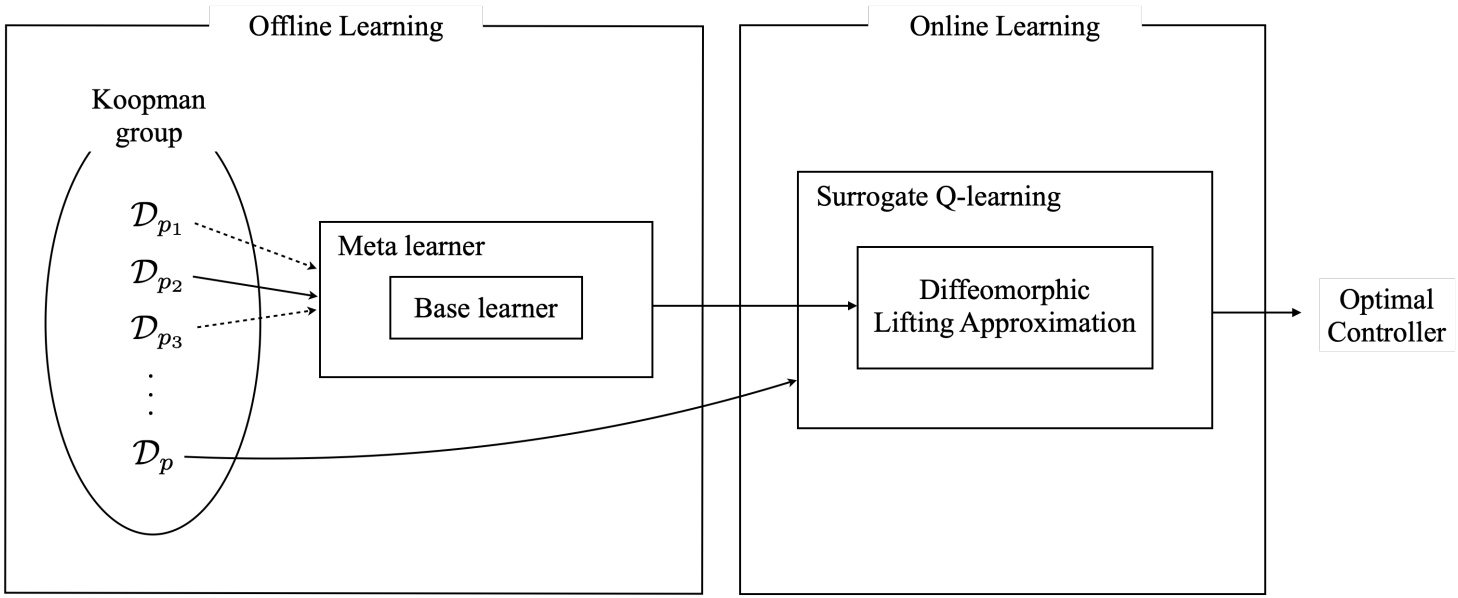

Figure 4.2 The proposed meta-learning and reinforcement learning scheme.

Chapter 5

Numerical Simulation

In this chapter, numerical simulation is performed to demonstrate the effectiveness of the method proposed in this study.

5.1 Koopman Group of Nonlinear Systems

Consider a nonlinear system given by

$$\begin{aligned} \dot{x}_1 &= x_1^3 + x_2 + u, \\ \dot{x}_2 &= p_1 x_1 + \left(p_2 - 3x_1^2\right)\left(x_1^3 + x_2\right) + \left(1 - 3x_1^2\right)u, \end{aligned} \tag{5.1}$$

where p_1 and p_2 are constant parameters satisfying

$$(p_1, p_2) \in \{(p_1, p_2) \mid 1 \leq p_1 \leq 2, 1 \leq p_2 \leq 2\} =: \mathcal{P}. \tag{5.2}$$

Let a system with a parameter tuple $p := (p_1, p_2)$ be denoted by $\mathcal{S}_p$, and let the group of such systems be $\mathcal{G}$. Consider the following optimal control problem:

$$\inf_u \int_0^\infty \left(x_1(t)^2 + u(t)^2\right) \mathrm{d}t \tag{5.3}$$

for any $x_0 \in \mathbb{R}^2$. The performance output is given by

$$y_o(t) = h(x(t)) = x_1(t), \tag{5.4}$$

where $x = [x_1, x_2]^T$.

Let a mapping $\phi : \mathbb{R}^2 \to \mathbb{R}^2$ be given by

$$\phi(x) = \begin{bmatrix} x_1 \\ x_1^3 + x_2 \end{bmatrix}, \tag{5.5}$$

which is surjective (C1) and satisfies $\phi(0) = 0$ (C5) by construction. It can be easily confirmed that the Jacobian

$$\boldsymbol{\nabla}\phi(x)^T = \begin{bmatrix} 1 & 0 \\ 3x_1^2 & 1 \end{bmatrix} \tag{5.6}$$

is injective for all $x \in \mathbb{R}^2$ (C2). The mapping $\phi(x)$ in (5.5) transforms the system (5.1) into a linear system given by

$$\dot{\phi} = \begin{bmatrix} 0 & 1 \\ p_1 & p_2 \end{bmatrix} \phi + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u =: A_p\phi + B_p u, \tag{5.7}$$

for any $\mathcal{S}_p \in \mathcal{G}$, which implies that $\phi(x)$ satisfies the conditions C3 and C4. Finally, for any $x \in \mathbb{R}^2$,

$$h(x) = \begin{bmatrix} 1 & 0 \end{bmatrix} \phi(x) =: C\phi(x), \tag{5.8}$$

where $h(x)$ is given in (5.4), which implies C6. It follows that all of the conditions C1–C6 of Theorem 4.8 are satisfied with $\phi(x)$, and therefore it can be confirmed that $\mathcal{G}$ is the Koopman group and $\phi(x)$ is the common lifting for the group. Furthermore, (A_p, B_p) is controllable and (A_p, C) is observable for all $p \in \mathcal{P}$.

By Theorem 4.8, the solution to the optimal control problem (5.3) for the nonlinear system $\mathcal{S}_p$ (5.1) is given by

$$u_p^*(x) = -B_p^T P_p^* \phi(x), \tag{5.9}$$

where P_p^* is the positive definite solution to the ARE given by

$$P_p^* A_p + A_p^T P_p^* + C^T C - P_p^* B_p B_p^T P_p^* = 0 \tag{5.10}$$

with A_p and B_p in (5.7), and C in (5.8).

5.2 The Meta-Learning Stage

5.2.1 Meta-Learning Setups

The single-scale Glow in Section 4.3.2 is implemented for the diffeomorphic lifting approximation, where the detailed hyper-parameters are presented in Table 5.1. The glow is constructed by 8 Flows connected sequentially. Two fully connected deep neural networks are employed for each affine coupling layer. For each layer in the fully connect networks, the exponential linear unit (ELU) is used for the activation function for differentiability.

At each iteration of the meta-training stage, 16 tasks (systems) are randomly generated using the parameter set $\mathcal{P}$ defined in (5.2). For each task, 1,000 pairs of $(x_i, u_i, \dot{x}_i)$ are generated by uniformly sampling x_i from $[-1, 1] \times [-1, 1]$ and u_i from $[-6, 6]$. The meta-learner loss $\mathcal{L}_{\text{meta}}$ (4.42) is averaged over all the sampled task dataset, and back-propagated using the Adam optimizer [77]. The iteration is repeated 3,000 times.

The number of Flows and the size of the deep neural networks in each affine coupling layer significantly affect the representational power of the diffeomorphic lifting approximation. Generally, a larger number of Flows and a larger-sized deep neural network can effectively model complex nonlinear dynamics, but beyond a certain level, there is little difference in representational power. Furthermore, increasing the number of Flows or the size of the deep neural network prolongs the time required for the meta-learning process and the computation of actual control inputs. Therefore, it is important to strike a balance between having adequate representational power and avoiding overly deep networks.

The weights for each loss determine the order in which the losses decrease during the initial stages of meta-learning. However, as the training time increases, all losses converge to very small values, resulting in less pronounced impact of the weight proportions on each loss.

5.2.2 Meta-Learning Results

Numerical results in this section demonstrate the feasibility of finding the common lifting for the Koopman group using the proposed meta-learning framework.

After the meta-learning, the approximated common lifting $\hat{\phi}(x; w_\phi)$ is obtained. Let

$$\hat{\phi}(x; w_\phi) =: \begin{bmatrix} \hat{\phi}_1(x; w_\phi) \\ \hat{\phi}_2(x; w_\phi) \end{bmatrix}. \tag{5.11}$$

Each component of $\hat{\phi}(x; w_\phi)$ is presented in Fig. 5.1. Because the lifting is approximated by differentiable invertible neural networks, the diffeomorphism between x and $\hat{\phi}$ can be observed, which implies that the conditions C1 and C2 are satisfied.

To demonstrate the Koopman lifting linearization performance of the proposed meta-learning framework, the parameter tuple is randomly sampled from $\mathcal{P}$, where the corresponding system is denoted by $\mathcal{S}_p \in \mathcal{G}$. Indeed, the sampled parameter tuple is $p = (1.55, 1.72)$. From (5.1), let

$$f(x) = \begin{bmatrix} x_1^3 + x_2 \\ p_1 x_1 + \left(p_2 - 3x_1^2\right)\left(x_1^3 + x_2\right) \end{bmatrix} =: \begin{bmatrix} f_1(x) \\ f_2(x) \end{bmatrix}, \tag{5.12}$$

$$G(x) = \begin{bmatrix} 1 \\ 1 - 3x_1^2 \end{bmatrix} =: \begin{bmatrix} G_1(x) \\ G_2(x) \end{bmatrix}, \tag{5.13}$$

Table 5.1 Meta-learning parameters.

Description	Variable	Value
Number of Flows	-	8
Units of hidden layers[1]	-	$[16, 64, 64]$
Weights of losses	η_{lin}	10
	η_{orig}	1
	η_{out}	5
	η_{nll}	1
Learning rate	-	10^{-3}
Weight decay	-	10^{-5}
Number of batch tasks	-	16
Size of a dataset	n_p	10^3

[1] For both of two fully connected layers, $s(x_1; w_s)$ and $t(x_1; w_t)$, in each affine coupling layers. See Appendix A.1.3.

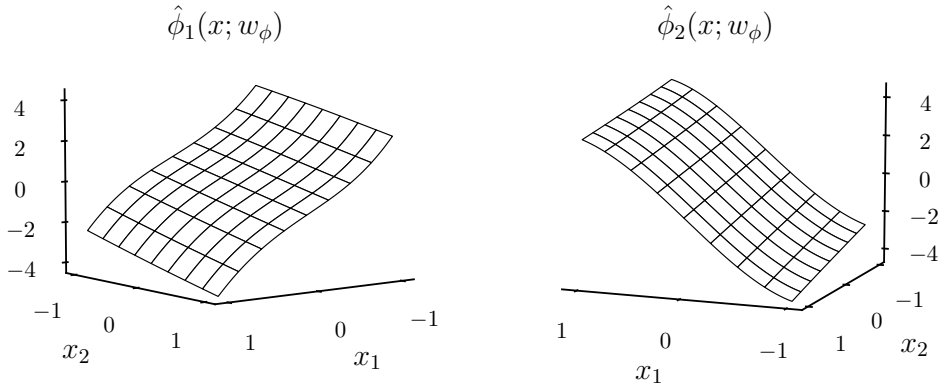

Figure 5.1 The approximated common lifting.

using the expression in (4.1). Figures 5.2 and 5.3 illustrate the functions $f(x)$ and $G(x)$, which demonstrate that the system dynamics is highly nonlinear.

According to the Koopman lifting linearization loss $\mathcal{L}_{\text{lin}}$ (4.43), the dynamics of the lifting

$$\frac{\mathrm{d}}{\mathrm{d}t}\hat{\phi}(x; w_\phi) = \boldsymbol{\nabla}\hat{\phi}(x; w_\phi)^T (f(x) + G(x)u) \tag{5.14}$$

should be linearized as

$$\frac{\mathrm{d}}{\mathrm{d}t}\hat{\phi} = \hat{A}_p\hat{\phi} + \hat{B}_p u \tag{5.15}$$

with some matrices $\hat{A}_p \in \mathbb{R}^{2\times 2}$ and $\hat{B}_p \in \mathbb{R}^{2\times 1}$. In other words, the followings should be satisfied for the learned lifting $\hat{\phi}(x; w_\phi)$:

$$\boldsymbol{\nabla}\hat{\phi}(x; w_\phi)^T f(x) = \hat{A}_p\hat{\phi}(x; w_\phi), \tag{5.16}$$

$$\boldsymbol{\nabla}\hat{\phi}(x; w_\phi)^T G(x) = \hat{B}_p. \tag{5.17}$$

These expectations can be confirmed in Figs. 5.4 and 5.5. The straight contours in Fig 5.4 imply that the function $\boldsymbol{\nabla}\hat{\phi}(x; w_\phi)^T f(x)$ is linear to $\hat{\phi}(x; w_\phi)$. The function $\boldsymbol{\nabla}\hat{\phi}(x; w_\phi)^T G(x)$ is almost constant for all $\hat{\phi}(x; w_\phi)$ as illustrated in Fig. 5.5.

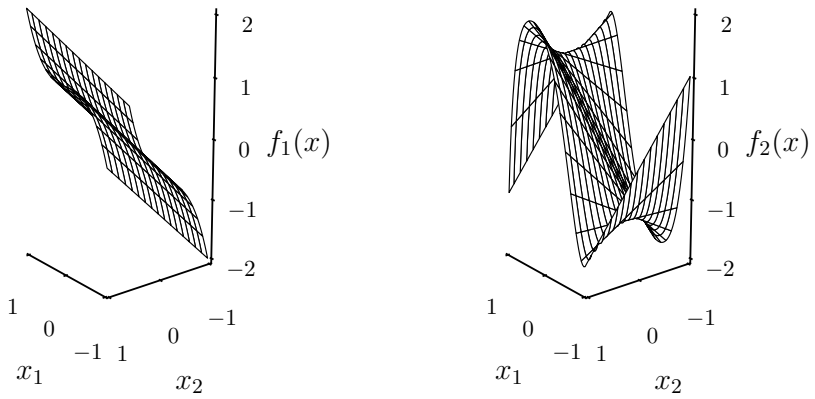

Figure 5.2 The functions $f_1(x)$ and $f_2(x)$.

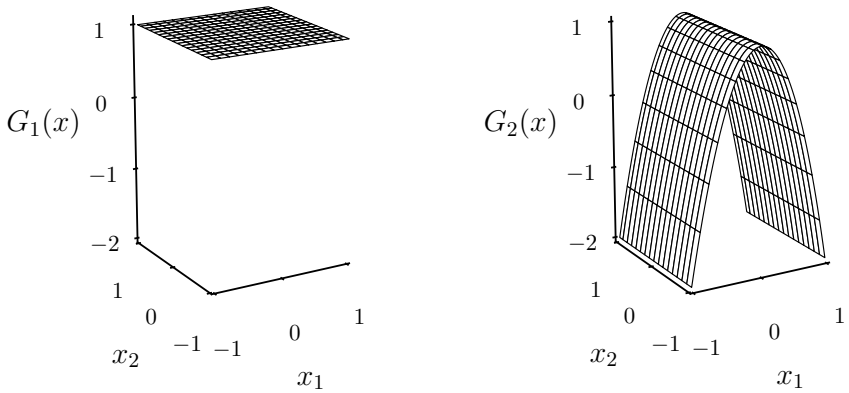

Figure 5.3 The functions $G_1(x)$ and $G_2(x)$.

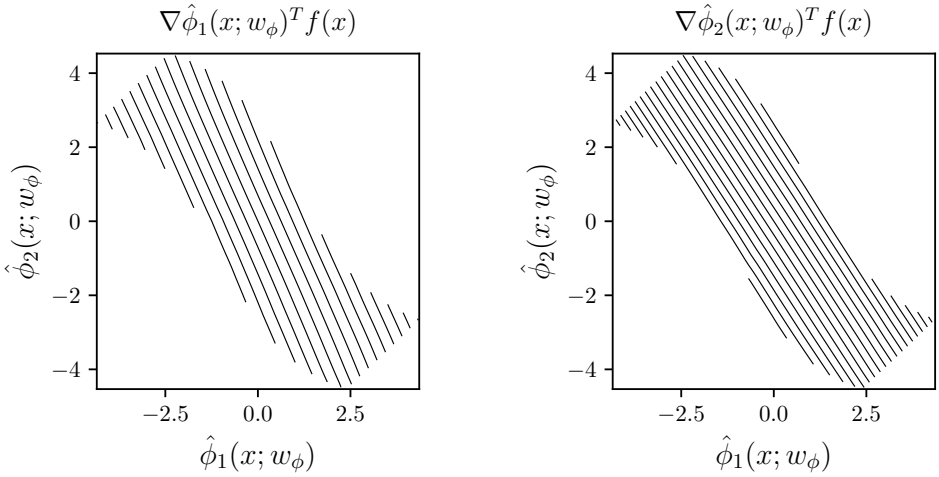

Figure 5.4 The contour plots of $\boldsymbol{\nabla}\hat{\phi}_1(x;w_\phi)^T f(x)$ and $\boldsymbol{\nabla}\hat{\phi}_2(x;w_\phi)^T f(x)$.

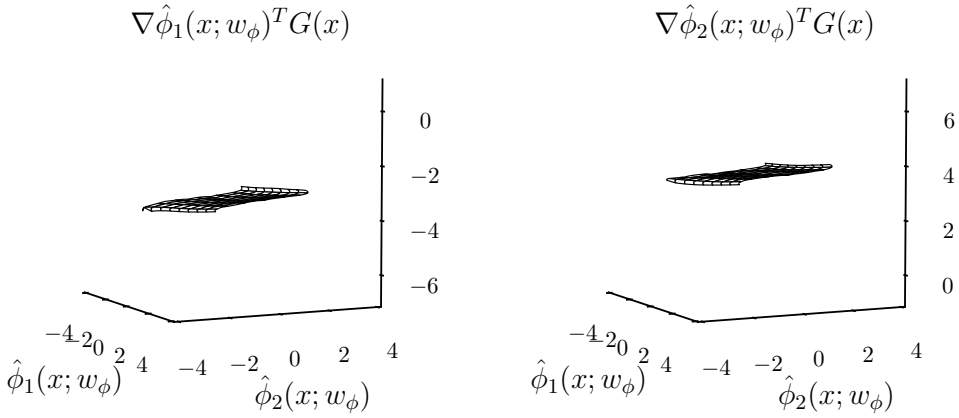

Figure 5.5 The functions $\boldsymbol{\nabla}\hat{\phi}_1(x;w_\phi)^T G(x)$ and $\boldsymbol{\nabla}\hat{\phi}_2(x;w_\phi)^T G(x)$.

Table 5.2 Mean-square linearization errors.

System	Parameter	Nonlinear System	KLL
$\mathcal{S}_{p_1}$	$(1.77, 1.81)$	9.97	1.09×10^{-3}
$\mathcal{S}_{p_2}$	$(1.28, 1.32)$	10.0	8.6×10^{-4}
$\mathcal{S}_{p_3}$	$(1.57, 1.00)$	10.2	8.6×10^{-4}

The mean-square errors of least-square linearization for both of the nonlinear system (5.1) and the Koopman lifting linearization (KLL) (5.14) are given in Table 5.2 with three different randomly sampled systems, $\mathcal{S}_{p_1}$, $\mathcal{S}_{p_2}$ and $\mathcal{S}_{p_3}$. It can be confirmed that the meta-learned common lifting can linearize an arbitrary system in the group, which implies that the condition C3 and C4 are satisfied. In addition, the condition C5 can be confirmed from

$$\hat{\phi}(0; w_\phi) = (0.0021, -0.0013). \tag{5.18}$$

Finally, Fig. 5.6 indicates the satisfaction of the last condition C6, which implies that the performance output $y = x_1$ can be recovered from the lifting. The mean-square error of $h(x) - C\hat{\phi}(x; w_\phi)$ is 3.92×10^{-7}, where the matrix C is

$$C = \begin{bmatrix} -0.140 & 0.147 \end{bmatrix}. \tag{5.19}$$

As demonstrated above, the trained diffeomorphic lifting approximation satisfies the conditions C1–C6 of Theorem 4.8 under small linearization errors presented in Table 5.2. This means that an arbitrary system in the Koopman group follows the dynamics in (5.15) with some unknown matrices $\hat{A}_p$ and $\hat{B}_p$.

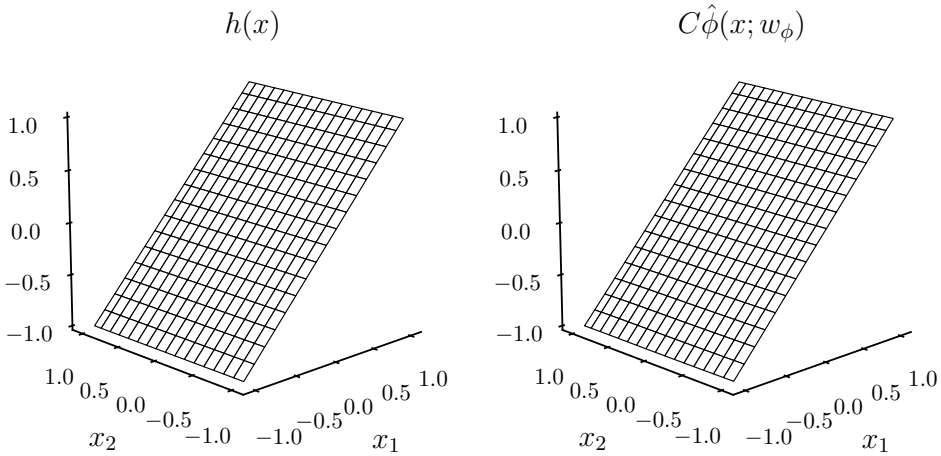

Figure 5.6 The performance output of the nonlinear system and the Koopman lifting linearization.

5.3 The Surrogate Q-Learning Stage

5.3.1 Surrogate Q-Learning Setups

After meta-learning stage, diffeomorphic lifting approximation $\hat{\phi}(x; w_\phi)$ and the corresponding output matrix C are obtained. The data-driven surrogate Q-learning is performed for the randomly sampled systems with Q° in (3.39) is constructed as

$$Q^\circ = \begin{bmatrix} C^T C & 0 \\ 0 & 1 \end{bmatrix}. \tag{5.20}$$

The surrogate Q-learning parameter is set as $s_k = 1$ for all $k \geq 0$, and the initial feedback gain K_0 is set to zero.

For each system, 10,000 data points of $(x_i, u_i, \dot{x}_i)$ are collected, and the

diffeomorphic lifting approximation $\hat{\phi}(x; w_\phi)$ transforms the dataset into

$$\mathcal{D}_p = \left\{ \left(\hat{\phi}(x_i; w_\phi), u_i, \boldsymbol{\nabla} \hat{\phi}(x_i; w_\phi)^T \dot{x}_i \right) \right\} =: \left\{ \left(\phi_i, u_i, \dot{\phi}_i \right) \right\}. \tag{5.21}$$

Note that the tuple $\left(\phi_i, u_i, \dot{\phi}_i\right)$ satisfies the linear dynamics (5.15). The surrogate Q-learning iteration is performed for 50 steps, although the solutions are converged much earlier.

5.3.2 Surrogate Q-Learning Results

To demonstrate the performance of the surrogate Q-learning, 20 systems are randomly sampled from the Koopman group $\mathcal{G}$, and run the surrogate Q-learning for each system. Figure 5.7 presents the learning history of the surrogate Q-learning for all systems in terms of $\nu(P_k)$, which is the number of eigenvalues with the negative real part of P_k, and the differences between each element of the learned feedback gains and the optimal gain, where

$$K_k =: \begin{bmatrix} K_{k,1} & K_{k,2} \end{bmatrix}, \quad K^* =: \begin{bmatrix} K_1^* & K_2^* \end{bmatrix}. \tag{5.22}$$

The upper plot in Fig. 5.7 illustrates that the initially unstable initial feedback gain is monotonically stabilized as the iteration progresses The simulation results demonstrate that the proposed surrogate Q-learning quickly stabilizes the feedback gain within a small number of iterations less than 5. The convergence of K_k can be confirmed from the middle and lower plots in Fig. 5.7. Figure 5.8 shows the full history of the differences $|K_{k,1} - K_1^*|$ and $|K_{k,2} - K_2^*|$ in a log scale. It can be observed that the control gain converges within approximately 25 steps with an error of around 10^{-10}.

To confirm that the surrogate Q-learning converged to the optimal control, the analytic optimal control inputs $u_p^*(x)$ (5.9), the trained control inputs $\hat{u}_p(x)$,

and the errors between the two control inputs are presented in Fig. 5.9 for the systems $\mathcal{S}_{p_1}$ to $\mathcal{S}_{p_3}$ in Table 5.2. Figure 5.10 illustrates the phase portraits of the closed-loop systems using the analytic optimal controller $u_p^*(x)$ and the trained controller $\hat{u}_p(x)$ for each system. The results show that the surrogate Q-learning closely converges to the analytically optimal controller for arbitrary systems using only the dataset acquired from each system.

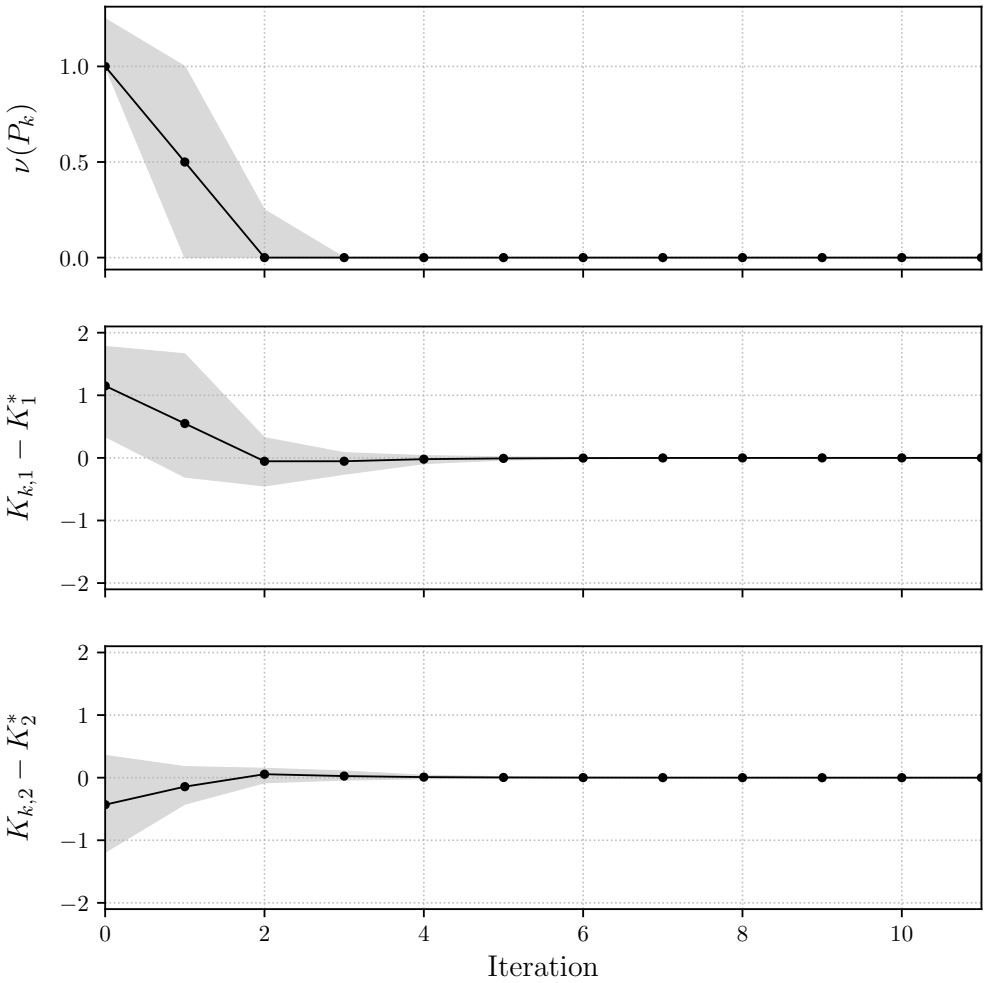

Figure 5.7 The learning history of the surrogate Q-learning for 20 different randomly sampled systems. The upper plot presents the median number of eigenvalues with the negative real part of P_k, and the middle and lower plots present the median error between the learned feedback gains and the optimal gain. The shaded area in each plot denotes the interquartile range (IQR).

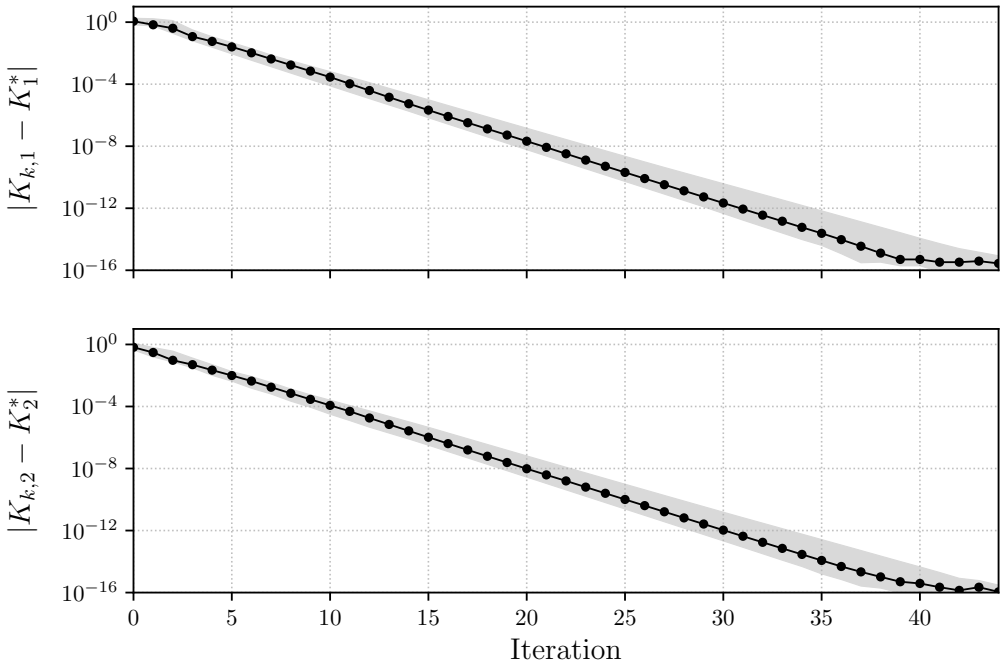

Figure 5.8 The feedback gain convergence histories of the surrogate Q-learning for 20 different randomly sampled systems. The median errors for each element between the learned feedback gains and the optimal gain are presented, and the shaded area denotes the IQR.

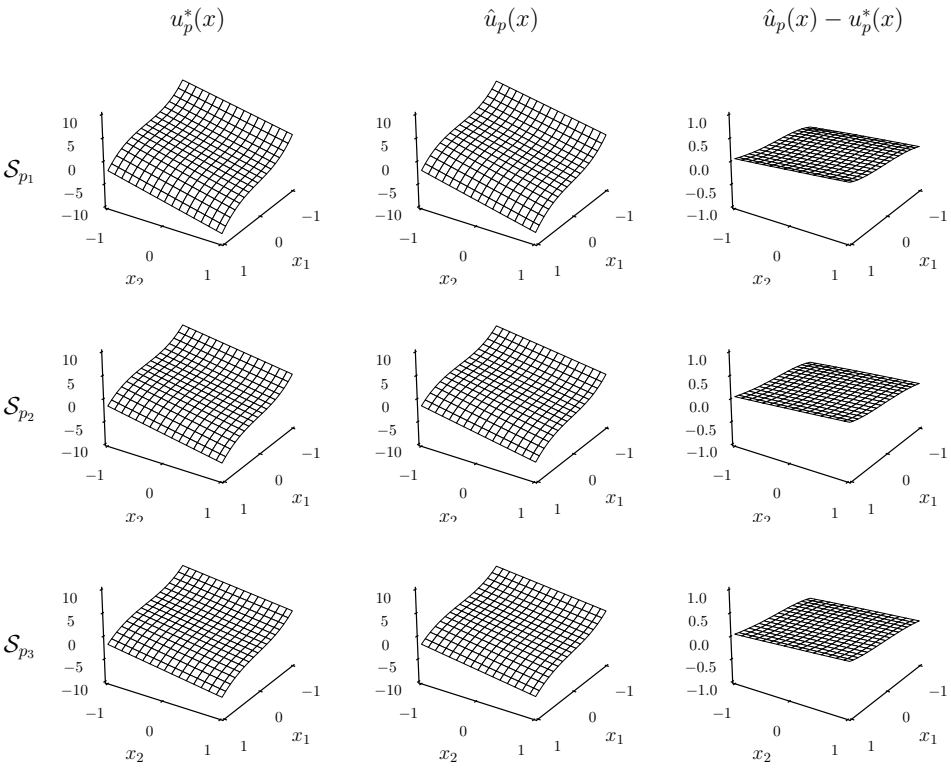

Figure 5.9 The optimal control inputs (left), the learned control inputs (middle), and the errors between the two (right) for the random systems $\mathcal{S}_{p_1}$ (top) to $\mathcal{S}_{p_3}$ (bottom).

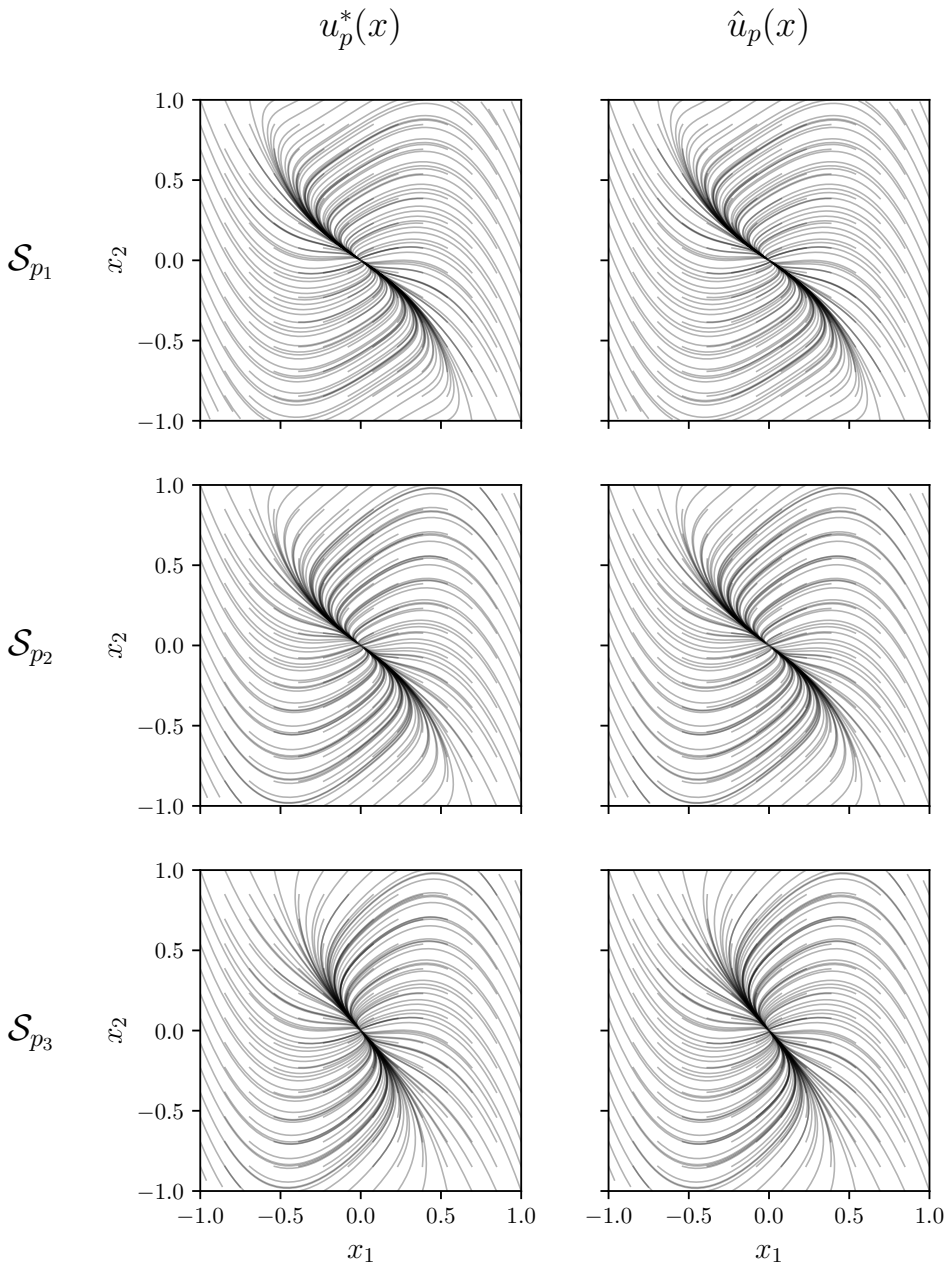

Figure 5.10 The phase portrait of the analytic optimal control (left) and the controller trained using the surrogate Q-learning (right) for each system.

Chapter 6

Conclusion

6.1 Concluding Remarks

A reinforcement learning algorithm for optimal control problems of dynamic systems is proposed utilizing a model-free off-policy approach. Widely acknowledged limitation in policy iteration regarding the use of unstable initial policies can be successfully overcome by the proposed policy iteration algorithm while maintaining the advantage of easy implementation. This substantial advancement will considerably broaden the scope of the reinforcement learning algorithm to accommodate a wide range of dynamic systems including inherently unstable systems. Moreover, meta-learning synthesis can be facilitated rapid acquisition of nonlinear optimal controllers by the proposed reinforcement learning algorithm, requiring only a small amount of actual data.

In this dissertation, the policy iteration algorithm is improved to accommodate unstable policies by redesigning the policy evaluation steps based on implicitly defined value functions. In the case of linear systems, the implicit value function corresponds to the unique symmetric solution of a Lyapunov equation. It is observed that the Kleinman iteration fails to stabilize the unstable initial policy. To overcome this limitation, the surrogate Q-learning is

introduced based on the implicit value function. The off-policy property of the proposed method enables the use of real data acquired from various control resources, such as human experts or experimentally obtained stable controllers, which makes the dataset acquisition processes much safer for dynamic systems including the aerospace systems.

The aforementioned characteristics and the convergence of the proposed algorithm are thoroughly examined through rigorous theoretical analysis, specifically for linear systems. The monotonic stabilization property of the extended Kleinman iteration is revealed using the matrix inertia theorem, where the closed-loop system can be stabilized in a finite number of iterations. The global convergence to the unique optimal stabilizing solution is rigorously proved based on the monotonic convergence theorem and the analysis of the local behavior of the iteration near the symmetric solutions to the ARE. In addition to these theoretical guarantee, the convergence property of the algorithm is demonstrated through illustrative numerical examples, which exhibits rapid convergence in only a few iterations.

The meta-learning framework is synthesized to apply the proposed reinforcement learning algorithm for linear systems to nonlinear systems. The properties of Koopman lifting linearization are thoroughly investigated to obtain the equivalence between the optimal control in the linearized system and in the original nonlinear system. Based on the theoretical findings, several conditions for the lifting were presented, and the diffeomorphic lifting approximation and meta-learning losses are proposed to satisfy the conditions. The feasibility and the efficacy of the proposed meta-learning framework are demonstrated using illustrative numerical simulations.

6.2 Direction for Further Research

The directions that follow are proposed as potential ways to extend and build upon the research presented in this dissertation.

Robustness Analysis of Surrogate Q-Learning

As discussed in Section 3.5.2, the proposed data-driven surrogate Q-learning algorithm possesses a certain level of robustness, although the exact level is unclear. The proposed algorithm requires state derivative data that is vulnerable to external disturbances and estimation noise. Therefore, analyzing the robustness of the algorithm is an important area for future research. The Moore-Penrose pseudo-inverse currently used in the policy evaluation stage ensures that the linear equation solution has a minimum norm error. It is necessary to analyze the physical significance of the surrogate Q-function obtained from the linear system and determine its impact on algorithm convergence and monotonic stabilization performance.

Relaxed Conditions for the Extended Kleinman Iteration

The analysis of the extended Kleinman iteration currently resides on the controllability and observability assumptions for the linear systems. However, it is well-known that the more relaxed stabilizability and detectability assumptions are enough to find a stable optimal controller [53]. Several iterative methods have been developed for the relaxed conditions, although they still require initial stable policies, see [78] and references therein. This relaxation can be particularly useful for the Koopman lifting linearization in a higher dimension, because the linearized system becomes uncontrollable with the mapping that is

not surjective by Lemma 4.6.

Koopman Groups Identification

The proposed meta-learning framework assumes the existence of a common subspace that is invariant to the Koopman operator for all systems in the group. Although uncertainties in the system dynamics, for example, variations in mass or moment of inertia in aerospace systems, are expected to satisfy this assumption to a sufficient degree, it is important from a practical control design perspective to verify whether this assumption holds for the implementation of the proposed framework.

Meta-Learning Framework for Adaptive Control Synthesis

Even if the Koopman group assumption is satisfied, the nature of deep neural network learning can leave some residual error in the Koopman lifting linearization. This may cause a performance degradation in surrogate Q-learning or even make the learned controller unstable. Therefore, it is practical to use traditional control techniques that can compensate for some level of system uncertainty when the trained controller is implemented, even after all the learning processes from meta-learning to surrogate Q-learning are completed. The development of a meta-learning framework for adaptive control synthesis is expected to enable the design of reliable learning-based controllers.

Bibliography

[1] Kleinman, D., "On an Iterative Technique for Riccati Equation Computations," *IEEE Transactions on Automatic Control*, Vol. 13, No. 1, 1968, pp. 114–115.
DOI:10.1109/TAC.1968.1098829

[2] Jiang, Y., and Jiang, Z.-P., *Robust Adaptive Dynamic Programming*, Wiley, Hoboken, NJ, 2017.

[3] Vrabie, D., Pastravanu, O., Abu-Khalaf, M., and Lewis, F., "Adaptive Optimal Control for Continuous-Time Linear Systems Based on Policy Iteration," *Automatica*, Vol. 45, No. 2, 2009, pp. 477–484.
DOI:10.1016/j.automatica.2008.08.017

[4] Jiang, Y., and Jiang, Z.-P., "Computational Adaptive Optimal Control for Continuous-Time Linear Systems with Completely Unknown Dynamics," *Automatica*, Vol. 48, No. 10, 2012, pp. 2699–2704.
DOI:10.1016/j.automatica.2012.06.096

[5] Datta, B. N., *Numerical Methods for Linear Control Systems*, Elsevier, Burlington, MA, 2003.

[6] Lancaster, P., and Rodman, L., "Existence and Uniqueness Theorems for the Algebraic Riccati Equation," *International Journal of Control*, Vol. 32,

No. 2, 1980, pp. 285–309.
DOI:10.1080/00207178008922858

[7] Lewis, F. L., Vrabie, D. L., and Syrmos, V. L., *Optimal Control*, 3rd ed., Wiley, Hoboken, NJ, 2012.

[8] Macfarlane, A. G. J., "An Eigenvector Solution of the Optimal Linear Regulator Problem," *Journal of Electronics and Control*, Vol. 14, No. 6, 1963, pp. 643–654.
DOI:10.1080/00207216308937540

[9] Laub, A., "A Schur Method for Solving Algebraic Riccati Equations," *IEEE Transactions on Automatic Control*, Vol. 24, No. 6, 1979, pp. 913–921.
DOI:10.1109/TAC.1979.1102178

[10] Balzer, L. A., "Accelerated Convergence of the Matrix Sign Function Method of Solving Lyapunov, Riccati and Other Matrix Equations," *International Journal of Control*, Vol. 32, No. 6, 1980, pp. 1057–1078.
DOI:10.1080/00207178008910040

[11] Byers, R., "Solving the Algebraic Riccati Equation with the Matrix Sign Function," *Linear Algebra and Its Applications*, Vol. 85, 1987, pp. 267–279.
DOI:10.1016/0024-3795(87)90222-9

[12] Sandell, N., "On Newton's Method for Riccati Equation Solution," *IEEE Transactions on Automatic Control*, Vol. 19, No. 3, 1974, pp. 254–255.
DOI:10.1109/TAC.1974.1100536

[13] Dieci, L., "Some Numerical Considerations and Newton's Method Revisited for Solving Algebraic Riccati Equations," *IEEE Transactions on Automatic Control*, Vol. 36, No. 5, 1991, pp. 608–616.
DOI:10.1109/9.76366

[14] Banks, H. T., and Ito, K., "A Numerical Algorithm for Optimal Feedback Gains in High Dimensional Linear Quadratic Regulator Problems," *SIAM Journal on Control and Optimization*, Vol. 29, No. 3, 1991, pp. 499–515.
DOI:10.1137/0329029

[15] Guo, C.-H., and Lancaster, P., "Analysis and Modification of Newton's Method for Algebraic Riccati Equations," *Mathematics of Computation*, Vol. 67, No. 223, 1998, pp. 1089–1106.
DOI:10.1090/S0025-5718-98-00947-8

[16] Howard, R. A., *Dynamic Programming and Markov Processes*, MIT Press, Cambridge, MA, 1960.

[17] Varga, A., "On Stabilization Methods of Descriptor Systems," *Systems & Control Letters*, Vol. 24, No. 2, 1995, pp. 133–138.
DOI:10.1016/0167-6911(94)00017-P

[18] Benner, P., and Byers, R., "An Exact Line Search Method for Solving Generalized Continuous-Time Algebraic Riccati Equations," *IEEE Transactions on Automatic Control*, Vol. 43, No. 1, 1998, pp. 101–107.
DOI:10.1109/9.654908

[19] Chehab, J.-P., and Raydan, M., "Inexact Newton's Method with Inner Implicit Preconditioning for Algebraic Riccati Equations," *Computational*

and Applied Mathematics, Vol. 36, No. 2, 2017, pp. 955–969.
DOI:10.1007/s40314-015-0274-8

[20] Murray, J., Cox, C., Lendaris, G., and Saeks, R., "Adaptive Dynamic Programming," *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, Vol. 32, No. 2, 2002, pp. 140–153.
DOI:10.1109/TSMCC.2002.801727

[21] Bellman, R., and Dreyfus, S., "Functional Approximations and Dynamic Programming," *Mathematics of Computation*, Vol. 13, No. 68, 1959, pp. 247–251.
DOI:10.1090/S0025-5718-1959-0107376-8

[22] Bian, T., and Jiang, Z.-P., "Value Iteration and Adaptive Dynamic Programming for Data-Driven Adaptive Optimal Control Design," *Automatica*, Vol. 71, 2016, pp. 348–360.
DOI:10.1016/j.automatica.2016.05.003

[23] Bian, T., and Jiang, Z.-P., "Reinforcement Learning and Adaptive Optimal Control for Continuous-Time Nonlinear Systems: A Value Iteration Approach," *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 33, No. 7, 2021, pp. 2781–2790.
DOI:10.1109/TNNLS.2020.3045087

[24] Lee, J. Y., Park, J. B., and Choi, Y. H., "On Integral Generalized Policy Iteration for Continuous-Time Linear Quadratic Regulations," *Automatica*, Vol. 50, No. 2, 2014, pp. 475–489.
DOI:10.1016/j.automatica.2013.12.009

[25] Budišić, M., Mohr, R., and Mezić, I., "Applied Koopmanism," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, Vol. 22, No. 4, Article 047510, 2012.
DOI:10.1063/1.4772195

[26] Mezic, I., "On Applications of the Spectral Theory of the Koopman Operator in Dynamical Systems and Control Theory," *IEEE 54th Conference on Decision and Control (CDC)*, Osaka, Japan, Dec. 2015.
DOI:10.1109/CDC.2015.7403328

[27] Williams, M. O., Kevrekidis, I. G., and Rowley, C. W., "A Data–Driven Approximation of the Koopman Operator: Extending Dynamic Mode Decomposition," *Journal of Nonlinear Science*, Vol. 25, No. 6, 2015, pp. 1307–1346.
DOI:10.1007/s00332-015-9258-5

[28] Williams, M. O., Hemati, M. S., Dawson, S. T., Kevrekidis, I. G., and Rowley, C. W., "Extending Data-Driven Koopman Analysis to Actuated Systems," *10th IFAC Symposium on Nonlinear Control Systems*, Vol. 49, Monterey, CA, Aug. 2016.
DOI:10.1016/j.ifacol.2016.10.248

[29] Abraham, I., de la Torre, G., and Murphey, T., "Model-Based Control Using Koopman Operators," *Robotics: Science and Systems XIII*, Cambridge MA, July 2017.
DOI:10.15607/RSS.2017.XIII.052

[30] Brunton, S. L., Brunton, B. W., Proctor, J. L., and Kutz, J. N., "Koopman Invariant Subspaces and Finite Linear Representations of Nonlinear Dynamical Systems for Control," *PLOS ONE*, Vol. 11, No. 2, Article e0150171, 2016.
DOI:10.1371/journal.pone.0150171

[31] Proctor, J. L., Brunton, S. L., and Kutz, J. N., "Generalizing Koopman Theory to Allow for Inputs and Control," *SIAM Journal on Applied Dynamical Systems*, Vol. 17, No. 1, 2018, pp. 909–930.
DOI:10.1137/16M1062296

[32] Korda, M., and Mezić, I., "Linear Predictors for Nonlinear Dynamical Systems: Koopman Operator Meets Model Predictive Control," *Automatica*, Vol. 93, 2018, pp. 149–160.
DOI:10.1016/j.automatica.2018.03.046

[33] Surana, A., "Koopman Operator Based Observer Synthesis for Control-Affine Nonlinear Systems," *IEEE 55th Conference on Decision and Control (CDC)*, Las Vegas, NV, Dec. 2016.
DOI:10.1109/CDC.2016.7799268

[34] Kaiser, E., Kutz, J. N., and Brunton, S. L., "Data-Driven Discovery of Koopman Eigenfunctions for Control," *Machine Learning: Science and Technology*, Vol. 2, No. 3, Article 035023, 2021.
DOI:10.1088/2632-2153/abf0f5

[35] Korda, M., and Mezic, I., "Optimal Construction of Koopman Eigenfunctions for Prediction and Control," *IEEE Transactions on Automatic Con-*

trol, Vol. 65, No. 12, 2020, pp. 5114–5129.
DOI:10.1109/TAC.2020.2978039

[36] Mamakoukas, G., Castano, M., Tan, X., and Murphey, T., "Local Koopman Operators for Data-Driven Control of Robotic Systems," *Robotics: Science and Systems XV*, Freiburg im Breisgau, Germany, June 2019.
DOI:10.15607/RSS.2019.XV.054

[37] Narasingam, A., and Kwon, J. S.-I., "Koopman Lyapunov-based Model Predictive Control of Nonlinear Chemical Process Systems," *AIChE Journal*, Vol. 65, No. 11, Article e16743, 2019.
DOI:10.1002/aic.16743

[38] Peitz, S., and Klus, S., "Koopman Operator-Based Model Reduction for Switched-System Control of Pdes," *Automatica*, Vol. 106, 2019, pp. 184–191.
DOI:10.1016/j.automatica.2019.05.016

[39] Klus, S., Nüske, F., Peitz, S., Niemann, J.-H., Clementi, C., and Schütte, C., "Data-Driven Approximation of the Koopman Generator: Model Reduction, System Identification, and Control," *Physica D: Nonlinear Phenomena*, Vol. 406, Article 132416, 2020.
DOI:10.1016/j.physd.2020.132416

[40] Bruder, D., Fu, X., Gillespie, R. B., Remy, C. D., and Vasudevan, R., "Data-Driven Control of Soft Robots Using Koopman Operator Theory," *IEEE Transactions on Robotics*, Vol. 37, No. 3, 2021, pp. 948–961.
DOI:10.1109/TRO.2020.3038693

[41] Otto, S. E., and Rowley, C. W., "Koopman Operators for Estimation and Control of Dynamical Systems," *Annual Review of Control, Robotics, and Autonomous Systems*, Vol. 4, No. 1, 2021, pp. 59–87.
DOI:10.1146/annurev-control-071020-010108

[42] Brunton, S. L., Budišić, M., Kaiser, E., and Kutz, J. N., "Modern Koopman Theory for Dynamical Systems," *SIAM Review*, Vol. 64, No. 2, 2022, pp. 229–340.
DOI:10.1137/21M1401243

[43] Yeung, E., Liu, Z., and Hodas, N. O., "A Koopman Operator Approach for Computing and Balancing Gramians for Discrete Time Nonlinear Systems," *American Control Conference (ACC)*, Milwaukee, WI, June 2018.
DOI:10.23919/ACC.2018.8431738

[44] Folkestad, C., Pastor, D., Mezic, I., Mohr, R., Fonoberova, M., and Burdick, J., "Extended Dynamic Mode Decomposition with Learned Koopman Eigenfunctions for Prediction and Control," *American Control Conference (ACC)*, Denver, CO, July 2020.
DOI:10.23919/ACC45564.2020.9147729

[45] Krolicki, A., Sutavani, S., and Vaidya, U., "Koopman-Based Policy Iteration for Robust Optimal Control," *American Control Conference (ACC)*, Atlanta, GA, June 2022.
DOI:10.23919/ACC53348.2022.9867541

[46] Shi, H., and Meng, M. Q.-H., "Deep Koopman Operator with Control for Nonlinear Systems," *IEEE Robotics and Automation Letters*, Vol. 7, No. 3,

2022, pp. 7700–7707.
DOI:10.1109/LRA.2022.3184036

[47] Zinage, V., and Bakolas, E., "Neural Koopman Lyapunov Control," *Neurocomputing*, Vol. 527, 2023, pp. 174–183.
DOI:10.1016/j.neucom.2023.01.029

[48] Higham, N. J., *Functions of Matrices: Theory and Computation*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2008.

[49] Rudin, W., *Functional Analysis*, International Series in Pure and Applied Mathematics, 2nd ed., McGraw-Hill, New York, NY, 1991.

[50] Mauroy, A., Mezić, I., and Susuki, Y., editors, *The Koopman Operator in Systems and Control: Concepts, Methodologies, and Applications*, No. 484 in Lecture Notes in Control and Information Sciences, Springer, Cham, Switzerland, 2020.

[51] Chen, C.-T., *Linear System Theory and Design*, 3rd ed., Oxford University Press, New York, NY, 1999.

[52] Lancaster, P., and Rodman, L., *Algebraic Riccati Equations*, Oxford Univ. Press, New York, NY, 1995.

[53] Wonham, W. M., *Linear Multivariable Control: A Geometric Approach*, 3rd ed., Springer-Verlag, New York, NY, 1985.

[54] Liberzon, D., *Calculus of Variations and Optimal Control Theory: A Concise Introduction*, Princeton Univ. Press, Princeton, NJ, 2012.

[55] Datta, B. N., "Stability and Inertia," *Linear Algebra and Its Applications*, Vol. 302–303, 1999, pp. 563–600.
DOI:10.1016/S0024-3795(99)00213-X

[56] Finn, C., Abbeel, P., and Levine, S., "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks," *34th International Conference on Machine Learning*, Sydney, Australia, Aug. 2017.

[57] Hospedales, T. M., Antoniou, A., Micaelli, P., and Storkey, A. J., "Meta-Learning in Neural Networks: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 44, No. 9, 2022, pp. 5149–5169.
DOI:10.1109/TPAMI.2021.3079209

[58] Bertinetto, L., Henriques, J. F., Torr, P. H. S., and Vedaldi, A., "Meta-Learning with Differentiable Closed-Form Solvers," *7th International Conference on Learning Representations*, New Orleans, LA, May 2019.

[59] Beard, R. W., Saridis, G. N., and Wen, J. T., "Galerkin Approximations of the Generalized Hamilton-Jacobi-Bellman Equation," *Automatica*, Vol. 33, No. 12, 1997, pp. 2159–2177.
DOI:10.1016/S0005-1098(97)00128-3

[60] Vamvoudakis, K. G., and Lewis, F. L., "Online Actor–Critic Algorithm to Solve the Continuous-Time Infinite Horizon Optimal Control Problem," *Automatica*, Vol. 46, No. 5, 2010, pp. 878–888.
DOI:10.1016/j.automatica.2010.02.018

[61] Sutton, R. S., and Barto, A. G., *Reinforcement Learning: An Introduction*, 2nd ed., A Bradford Book, Cambridge, MA, 2018.

[62] Fung, H.-K., "Linear Preservers of Controllability and/or Observability," *Linear Algebra and Its Applications*, Vol. 246, 1996, pp. 335–360.
DOI:10.1016/0024-3795(94)00364-5

[63] Haynsworth, E. V., "Determination of the Inertia of a Partitioned Hermitian Matrix," *Linear Algebra and Its Applications*, Vol. 1, No. 1, 1968, pp. 73–81.
DOI:10.1016/0024-3795(68)90050-5

[64] Feitzinger, F., Hylla, T., and Sachs, E. W., "Inexact Kleinman–Newton Method for Riccati Equations," *SIAM Journal on Matrix Analysis and Applications*, Vol. 31, No. 2, 2009, pp. 272–288.
DOI:10.1137/070700978

[65] Hu, Q., and Cheng, D., "The Polynomial Solution to the Sylvester Matrix Equation," *Applied Mathematics Letters*, Vol. 19, No. 9, 2006, pp. 859–864.
DOI:10.1016/j.aml.2005.09.005

[66] Stevens, B. L., Lewis, F. L., and Johnson, E. N., *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*, 3rd ed., Wiley-Blackwell, Hoboken, NJ, 2015.

[67] Sastry, S., *Nonlinear System: Analysis, Stability, and Control*, Vol. 10 of *Interdisciplinary Applied Mathematics*, Springer, New York, NY, 1999.

[68] Khalil, H. K., *Nonlinear Systems*, 3rd ed., Prentice Hall, Upper Saddle River, NJ, 2002.

[69] Wu, F., and Desoer, C., "Global Inverse Function Theorem," *IEEE Transactions on Circuit Theory*, Vol. 19, No. 2, 1972, pp. 199–201.
DOI:10.1109/TCT.1972.1083429

[70] Bevanda, P., Sosnowski, S., and Hirche, S., "Koopman Operator Dynamical Models: Learning, Analysis and Control," *Annual Reviews in Control*, Vol. 52, 2021, pp. 197–212.
DOI:10.1016/j.arcontrol.2021.09.002

[71] Bevanda, P., Kirmayr, J., Sosnowski, S., and Hirche, S., "Learning the Koopman Eigendecomposition: A Diffeomorphic Approach," *American Control Conference (ACC)*, Atlanta, GA, June 2022.
DOI:10.23919/ACC53348.2022.9867829

[72] Bevanda, P., Beier, M., Kerz, S., Lederer, A., Sosnowski, S., and Hirche, S., "Diffeomorphically Learning Stable Koopman Operators," *IEEE Control Systems Letters*, Vol. 6, 2022, pp. 3427–3432.
DOI:10.1109/LCSYS.2022.3184927

[73] Dinh, L., Sohl-Dickstein, J., and Bengio, S., "Density Estimation Using Real NVP," *5th International Conference on Learning Representations*, Toulon, France, April 2017.

[74] Teshima, T., Tojo, K., Ikeda, M., Ishikawa, I., Oono, K., and Sugiyama, M., "Coupling-Based Invertible Neural Networks Are Universal Diffeomorphism Approximators," *34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, Vancouver, BC, Canada, Dec. 2020.

[75] Kingma, D. P., and Dhariwal, P., "Glow: Generative Flow with Invertible 1x1 Convolutions," *32nd Conference on Neural Information Processing Systems (NeurIPS 2018)*, Vol. 31, Montréal, QC, Canada, Dec. 2018.

[76] Han, Y., Hao, W., and Vaidya, U., "Deep Learning of Koopman Representation for Control," *59th IEEE Conference on Decision and Control (CDC)*, Jeju Island, Republic of Korea, Dec. 2020.
DOI:10.1109/CDC42340.2020.9304238

[77] Kingma, D. P., and Ba, J., "Adam: A Method for Stochastic Optimization," *3rd International Conference on Learning Representations*, San Diego, CA, May 2015.

[78] Benner, P., Hernández, V., and Pastor, A., "The Kleinman Iteration for Nonstabilizable Systems," *Mathematics of Control, Signals and Systems*, Vol. 16, No. 1, 2003, pp. 76–93.
DOI:10.1007/s00498-003-0130-z

Appendix A

The Glow Implementation

The single-scale Glow used in this study has a simple form, without the stepwise scale expansion part for image processing, proposed in the multi-scale Glow [75]. As shown in Fig. A.1, the Glow has a structure where multiple layers called Flows are connected in succession.

A.1 Flows

Each Flow of the Glow network is composed of three invertible neural networks, which implies that the Flow itself is invertible. The architecture of one Flow is illustrated in Fig. A.2. In the following section, a detailed description is provided for each layer that constitutes the Flow.

A.1.1 Activation Layers

Given an input $x \in \mathbb{R}^n$, the output of the activation layer is defined by

$$y = s \odot x + b, \tag{A.1}$$

where $s \in \mathbb{R}^n$ and $b \in \mathbb{R}^n$ are trainable network parameters, and $\odot$ denotes the element-wise product. Given $y \in \mathbb{R}^n$, the inverse network produces $x \in \mathbb{R}^n$

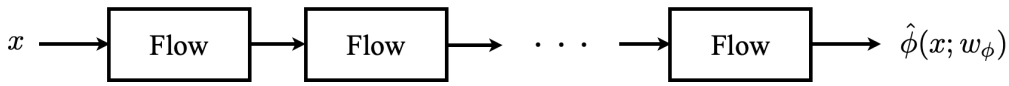

Figure A.1 The architecture of the Glow.

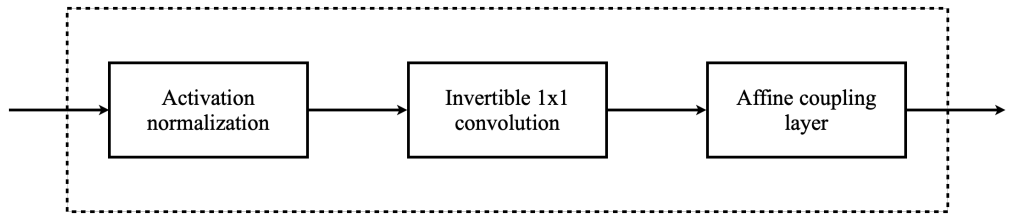

Figure A.2 The architecture of a Flow.

given by

$$x = (y - b) \oslash s, \tag{A.2}$$

where $\oslash$ is the element-wise division. The log-determinant is given by

$$\log \left| \det \left(\frac{\partial y}{\partial x} \right) \right| = \sum_{i=1}^{n} \log |s_i|, \tag{A.3}$$

where $s =: [s_1, \ldots, s_n]^T$.

A.1.2 1×1 Convolution Layers

Given an input $x \in \mathbb{R}^n$, the output $y \in \mathbb{R}^n$ of the 1×1 convolution layer is defined by

$$y = Wx, \tag{A.4}$$

and the inverse is given by

$$x = W^{-1}y. \tag{A.5}$$

Here, the invertible matrix $W \in \mathbb{R}^{n \times n}$ denotes the weight matrix of the convolution network, which is given by its LU decomposition as

$$W = P(L + I_n)(U + \mathrm{diag}(s)), \tag{A.6}$$

where $P \in \mathbb{R}^{n \times n}$ is the permutation matrix, $L \in \mathbb{R}^{n \times n}$ and $U \in \mathbb{R}^{n \times n}$ are the lower and upper triangular matrices with zero diagonal entries, respectively, and $s \in \mathbb{R}^n$. The trainable network parameters are L, U, and s. The log-determinant is given by

$$\log \left| \det \left(\frac{\partial y}{\partial x} \right) \right| = \sum_{i=1}^{n} \log |s_i|, \tag{A.7}$$

where $s =: [s_1, \ldots, s_n]^T$.

A.1.3 Affine Coupling Layers

Given an input $x \in \mathbb{R}^n$, affine coupling layers first partition it with a fixed size $p < n$ as

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \tag{A.8}$$

where $x_1 \in \mathbb{R}^p$ and $x_2 \in \mathbb{R}^{n-p}$. There are two deep neural networks in an affine coupling layer: $s(\cdot; w_s) : \mathbb{R}^p \to \mathbb{R}^p$ and $p(\cdot; w_p) : \mathbb{R}^p \to \mathbb{R}^p$, where w_s and w_p are the trainable network parameters. Then, the output $y \in \mathbb{R}^n$ is given by $y = [y_1^T, y_2^T]^T$, where

$$y_1 = x_1 \tag{A.9}$$

$$y_2 = \exp(s(x_1; w_s)) \odot x_2 + t(x_1; w_t). \tag{A.10}$$

Given y, the inverse of the network can be obtained as

$$x_1 = y_1, \tag{A.11}$$

$$x_2 = (y_2 - t(y_1; w_t)) \oslash \exp(s(y_1; w_s)). \tag{A.12}$$

The log-determinant is given by

$$\log \left| \det \left(\frac{\partial y}{\partial x} \right) \right| = \sum_{i=1}^{p} \log |s_i|, \tag{A.13}$$

where $s(x_1; w_s) =: [s_1, \ldots, s_p]$.

국문초록

본 논문에서는 최적제어 문제를 해결하기 위해 비모델(model-free) 강화학습 알고리듬을 제안하였다. 제어 시스템의 안정성은 제어기 설계 시 필수적으로 고려되어야 할 사항으로 본 논문에서 제안한 알고리듬은 학습되는 제어기가 최적일 뿐만 아니라 안정한 제어기로 수렴하도록 설계되었다. 기존의 근사 동적 프로그래밍 기법들과는 달리, 제안한 알고리듬은 안정한 초기 제어기를 필요로 하지 않는데, 이는 불안정한 평형점을 가지는 시스템의 비모델 학습 관점에서 주요한 장점이다.

논문의 전반부에서는 데이터만을 이용해 선형 시스템의 안정한 최적제어기를 학습할 수 있는 새로운 형태의 Q-학습 알고리듬을 제안한다. 초기 불안정한 제어 입력을 허용하기 위해 성능지수를 평가하기 위한 가치함수를 음함수 형태 재정의하고, 선형 시스템에 대해 존재성과 유일성을 보였다. 가상의 제어 동역학을 상태변수에 추가한 확장된 상태공간에서의 가치 음함수로 Q-함수를 정의하고, 이를 기반으로 하는 정책 반복법 기반의 Q-학습 알고리듬을 제안하였다. 이 알고리듬은 학습 중인 제어기로부터 데이터를 얻을 필요가 없는 off-policy 기법으로, 시스템의 숙련된 운영자나 실험적으로 설계된 PID 제어기를 통해 얻은 데이터를 사용할 수 있다는 장점이 있다. 제안한 Q-러닝 알고리듬을 이용하면 학습되는 제어기가 유한 단계 이내에 안정화 되며, 최종적으로 대수적 리카티(Riccati) 방정식의 안정한 선형 최적해로 수렴함을 행렬 관성 이론을 기반으로 증명하였다.

논문의 후반부는 제안된 강화학습 알고리즘을 비선형 시스템에 적용하는 문제를 다룬다. 이를 위해 비선형 시스템을 무한 차원 공간에서 선형화하는 쿠프만(Koopman) 연산자 이론을 활용한다. 리프팅(lifting)이라 불리는 매핑에 의해 생성되는 쿠프만 연산자의 유한 차원의 불변 부분공간이 존재한다고 가정할 때,

선형화된 시스템의 최적제어를 위해 가제어성과 가관측성을 가지기 위한 조건을 정립한다. 리프팅에 대한 여러 조건을 바탕으로 기존 비선형 시스템 최적제어 문제와 선형화된 시스템의 최적제어 문제 간의 동치성을 증명하고, 앞서 제안한 강화학습 알고리즘을 사용할 수 있는 이론적 근거를 마련한다. 모든 조건을 만족하는 리프팅을 찾기 위해 가역 심층신경망을 활용한 미분동형(diffeomorphic) 리프팅 근사법을 제안한다. 특정 시스템 그룹에 대해 공통된 리프팅이 존재한다면 그룹 내의 불확실한 시스템에 대해 제안한 비모델 강화학습을 활용할 수 있다는 점에 착안하여, 공통 리프팅을 학습하는 메타 러닝(meta learning) 프레임워크를 개발하였다.

마지막으로 이미 알려진 최적 제어기와 비선형 동역학을 갖는 비선형 시스템을 사용하여 수치 시뮬레이션을 수행하고, 제안된 프레임워크의 타당성과 구현 세부사항을 살펴보았다.

주요어: 강화 학습, 데이터 기반 제어, 학습 기반 제어, 자동 제어 시스템, 최적 제어, 대수적 리카티 방정식

학번: 2015-20765