HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

GRADUATION THESIS

Robust Optimal Control for Nonlinear Systems Based on Adaptive Reinforcement Learning

LE CONG NHAT ANH
anh.lcn173649@sis.hust.edu.vn

NGUYEN XUAN KHAI
khai.nx173970@sis.hust.edu.vn

Field: Control Engineering and Automation
Major: Automatic Control

Supervisor: **Dao Phuong Nam, Ph.D.**

Signature of Supervisor

Department: **Automatic Control**
School: **School of Electrical Engineering**

HANOI, 07/2021

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP

Điều khiển tối tư bền vững cho hệ phi tuyến dựa trên thuật toán học tăng cường thích nghi

LÊ CÔNG NHẬT ANH
anh.lcn173649@sis.hust.edu.vn

NGUYỄN XUÂN KHẢI
khai.nx173970@sis.hust.edu.vn

Ngành KT Điều khiển và Tự động hóa
Chuyên ngành Điều khiển Tự động

Giảng viên hướng dẫn: **TS. Đào Phương Nam**

Chữ ký của GVHD

Bộ môn: **Điều khiển Tự động**
Viện: **Điện**

HÀ NỘI, 07/2021

MINISTRY OF EDUCATION AND TRAINING
HANOI UNIV. OF SCI. AND TECH.

THE SOCIALIST REPUBLIC OF VIETNAM
Independence – Freedom - Happiness

MISSION OF THESIS

Students: Le Cong Nhat Anh
Nguyen Xuan Khai
School of Electrical Engineering
Major: Automatic Control Cohort: 62

1. Thesis title:

   Robust optimal control for nonlinear systems based on adaptive reinforcement learning

2. Thesis content:

   This work studies reinforcement learning (RL) or adaptive/approximate dynamic programming (ADP) methods in optimal feedback control to improve the closed-loop performance of nonlinear systems. The developed adaptive optimal controllers are associated with the solution of the Hamilton–Jacobi–Bellman equation by using policy iteration method, where the learning process occurs through an actor-critic (AC) structure. Integrating the forward-in-time methods with neural networks (NNs), the optimal policy and value function of continuous nonlinear systems are learned online in real time.

   Moreover, this work proposes the combination of RL/ADP-based design and nonlinear methods to develop robust optimal control for continuous nonlinear systems with uncertainties and disturbances. One contribution is the introduction of time-varying robust integral of the sign of the error (RISE) into RL-based control of second-order nonlinear systems. MATLAB simulation results on a 2-DOF robot arm demonstrate the improved performance of the time-varying RISE-based RL scheme in comparison with the original RISE-based RL controller. Another innovation is the disturbance observer-based RL control approach which not only learns the optimal policy but also learns the unknown disturbances. To verify the advantages of the proposed control structure, a comparison with the original RL-based method is made, implementing a surface vessel system simulation.

3. Supervisor: Dao Phuong Nam, Ph.D.

4. From: March 1, 2021

5. To: July 8, 2021

Date: .....

DEPARTMENT **SUPERVISOR** **STUDENTS**

Acknowledgments

This thesis development would not have been possible without the guidance and support which we received from many people. We would like to give our special regards to our supervisor, Dr. Dao Phuong Nam, for all the support and inspiration. We have acquired a lot of scientific research skills and broad knowledge from him when performing this thesis.

We are also indebted to all our lecturers at Hanoi University of Science and Technology, especially at the Department of Automatic Control, for great lessons over the past four years.

Finally, we wish to express our deepest appreciation to our families and friends for their constant help and endless encouragement even during the most challenging time.

Abstract

Coming to the best possible decision based on some predefined set of criteria is never easy, but it is always fascinating. This ability can empower higher degrees of autonomy in addressing problems from a wide range of areas, including artificial intelligence (AI), cybernetics, operations research, economics, and so on. By this drive, many mathematical theories have been proposed in relation to several concepts such as feedback control, optimality, adaptation, and learning. This work investigates the domain of optimal feedback control based on reinforcement learning (RL) and adaptive/approximate dynamic programming (ADP) to improve the closed-loop performance of nonlinear systems. The developed adaptive optimal controllers are associated with the solution of the Hamilton–Jacobi–Bellman equation by using policy iteration method, where the learning process occurs through an actor-critic (AC) structure. Integrating the forward-in-time methods with neural networks (NNs), the optimal policy and value function of continuous nonlinear systems are learned online in real time.

Moreover, this work proposes the combination of RL/ADP-based design and nonlinear methods to develop robust optimal control for continuous nonlinear systems with uncertainties and disturbances. One contribution is the introduction of time-varying robust integral of the sign of the error (RISE) into RL-based control of second-order nonlinear systems. MATLAB simulation results on a 2-DOF robot arm demonstrate the improved performance of the time-varying RISE-based RL scheme in comparison with the original RISE-based RL controller. Another innovation is the disturbance observer-based RL control approach which not only learns the optimal policy but also learns the unknown disturbances. To verify the advantages of the proposed control structure, a comparison with the original RL-based method is made, implementing a surface vessel system simulation.

This work motivates the next focus on partially/completely model-free RL methods such as integral RL and off-policy with explicit optimality and stability proof. Moreover, differential games and multi-agent systems are interesting topics where RL literature can be implemented.

Hanoi, ..., 2021
Students

Le Cong Nhat Anh

Nguyen Xuan Khai

Our Related Papers

1. **C. N. A. Le** and **X. K. Nguyen**, "Adaptive reinforcement learning of nonlinear systems with disturbances based on time-varying RISE method," in *the 38th Student Research Conference*, Hanoi University of Science and Technology (HUST), Hanoi, 2021.

2. D. D. Pham and **X. K. Nguyen**, "Multi-agent control with adaptive reinforcement learning strategy for surface vessels," in *the 38th Student Research Conference*, Hanoi University of Science and Technology (HUST), Hanoi, 2021.

3. P. N. Dao, D. D. Pham, **X. K. Nguyen**, and T. C. Nguyen, "Adaptive reinforcement learning motion/force control of multiple uncertain manipulators", in *2021 International Conference on Intelligent Systems* & Networks, Hanoi, 2021.

4. **X. K. Nguyen** and D. D. Pham, "Tracking control for a range of mechanical systems under input constraints", in *the 37th Student Research Conference*, Hanoi University of Science and Technology (HUST), Hanoi, 2020.

Contents

List of Figures

List of Tables

Acronyms and Abbreviations

$\|\cdot\|$	Euclide Norm.
RL	Reinforcement Learning.
ARL	Adaptive Reinforcement Learning.
IRL	Integral Reinforcement Learning.
HJB	Hamilton–Jacobi–Bellman.
TD	Temporal Difference.
PI	Policy Iteration.
VI	Value Iteration.
ADP	Approximate/Adaptive Dynamic Programming.
NN	Neural Network.
AC	Actor-Critic.
PE	Persistent of Excitation.
RISE	Robust Integral of The Sign of The Error.
DO	Disturbance Observer.
NDO	Nonlinear Disturbance Observer.
DOBC	Disturbance Observer-Based Control.
DOF	Degree of Freedom.
MIMO	Multi-Input Multi-Output.
SV	Surface Vessel.

Chapter 1

Introduction

1.1 Background and motivation

Real world systems are naturally nonlinear at least when considered over wide-ranging operating points. Artificial systems with advanced functionalities in practical processes usually present nonlinearities, time-varying unpredictable parameters, and other complexities from design limitation, operation conditions, unmodeled dynamics, and even internal and external disturbances [1].

As is well known, there are many methods for designing stable control for nonlinear systems. Different control techniques have been proposed for both theoretical interests and practical applications such as robust adaptive control [2, 3], sliding mode control (SMC) [4, 5], backstepping control [6, 7], and intelligent control [8, 9]. Moreover, the problems of input/output constraint, time delay, and finite time control have been considered in [10, 11, 12]. In fact, disturbances can be generalized from unmodeled dynamics, parameter variation, and external disturbances which widely exist in marine engineering (e.g. boats, ships, hovercraft and submarines), aerospace engineering such as missiles, aircrafts and satellites, and also many other engineering systems [13]. Adaptive and robust methods are two prevalent tools to handle uncertain/disturbed systems. Robust techniques, such as SMC [4, 5] and RISE method [14, 15], are also powerful tools to compensate for the uncertainties or disturbances in nonlinear systems. It is also well known that H_∞ control is one of the design methods for handling the disturbance attenuation problem of control systems [16]. However, H_∞ control is in general too conservative to obtain a highly accurate control performance under unknown disturbances. Recently, the successful development of disturbance observer-based control (DOBC) structures for nonlinear systems have been applied in a variety of control systems [17, 18]. A group of controllers known as adaptive controllers learns online to control unknown systems using data measured in real time along the system trajectories [19]. While learning the control solutions, adaptive controllers are able to guarantee stability and system performance [3]. However, stability is only a bare minimum requirement in a system design.

On the other hand, optimal control is an important research area in practice and theory [20]. The objective is to determine a control scheme that optimally drives the dynamics system to equilibrium in terms of a performance index function. Optimal controllers are typically designed offline by solving certain equations, for instance, the Riccati equation, utilizing full information of the system dynamics. Additionally, for nonlinear systems, optimal control can be derived by solving nonlinear HJB

equations that are difficult to find a global analytic solution [21]. The design of optimal controllers that inherits adaptive features (online learning in real time) can be studied by the mechanism of RL and ADP techniques which are the main topic of this thesis.

RL is an important machine learning concept that has been broadly studied in the artificial intelligence community [21]. RL technique involves an actor or agent which interacts with its environment and intends to learn the optimal actions, or control policies, by observing their responses from the environment. In other words, an RL agent performs its actions in order to minimize a long-term performance cost with the environment [22]. A novel stage of RL strategy was started by the introduction of ADP which has been a popular topic in recent times. The advent of RL and especially ADP algorithm bridged the gap between traditional optimal control and adaptive control algorithms [21, 23, 24, 25]. ADP delivers the potential of a new family of adaptive controllers that converge to optimal control schemes directly.

In RL and ADP, there is a substantial and successful body of work that approximates the HJB equations utilizing efficient forward-in-time approaches based on the use of NNs for value functional approximation. Originally, attention has primarily been given to ADP/RL-related control design for Markov decision processes (MDP) [9, 21, 24] and discrete-time feedback control systems [25, 26]. In Q-learning method [24], a Q function depending on both the state variable and control input is handled. Still, in MDP, the considered state spaces are finite or countable, and the stability problem is mostly implicit. Besides, due to the difference between discrete-time (DT) and continuous-time (CT) Bellman equations [25], the existing ADP techniques for DT systems cannot be directly brought to the CT case. Several studies have been conducted on the implementation of RL and ADP for CT systems. In [23], Euler's method was used to discretize the CT Bellman equation. If the equation is not properly discretized, then the DT solutions and the CT solutions are not in agreement. An online AC architecture was used in [27] to solve the continuous-time infinite horizon optimal control problem. In [28] the integral RL (IRL) technique was implemented based on integral reinforcement form. This concept is able to deal with dynamic uncertainties via off-policy method [29]. A different approach of adaptive RL (ARL) structure for unknown dynamics can be regarded using actor-critic-identifier [30]. Expanding this idea, with a special cost function, the authors in [31] design a model-free ARL scheme without any knowledge of the system dynamics. However, it has been said to be clumsy to have a system identification step [25]. Another direction of RL is to address optimal control problems with input saturation using a novel multi-gradient recursive (MGR) reinforcement learning approach [32]. The work in [33] investigated the stability of the ADP-based control combining ideas from reinforcement learning and robust control. Moreover, dealing with system uncertainties and unknown disturbances to achieve robust optimal performance is an interesting concern. In [34], ADP is designed with robust control (sliding variable and RISE method) to cope with uncertain/disturbed nonlinear systems. In general, it is crucial to develop advanced control methods for a wide range of systems and applications (e.g. automobiles, aerospace, industrial machines, and processes, biomedical uses, networks, and power systems) that demonstrate the performance of stability, optimality, adaptability, and robustness.

This work concentrates on the frame of RL/ADP-based control strategy combining with disturbance attenuation schemes for continuous-time nonlinear systems with uncertainties and disturbances.

1.2 Objectives and methodology

Inspired by the reviewed works and study from traditional nonlinear control techniques to adaptive optimal control scheme, the purpose of this thesis is to analyze and design novel ARL-based control structures for uncertain continuous-time nonlinear systems with disturbances. The algorithms have the following characteristics:

- Online and real-time control, avoiding system identification (direct or indirect).
- Guaranteeing robust stability towards disturbances.
- Minimizing performance index function and guaranteeing the convergence of the solution.
- Reducing computing resources to accelerate the speed of convergence.
- Simple and efficient to implement for a wide range of control problems.

These objectives contribute to the improved closed-loop performance of nonlinear systems.

Regarding methodology, this research-based thesis is conducted by

- Studying related references; providing preliminaries.
- Analyzing and designing control systems; computing and proving stability using mathematics.
- Verifying the proposed algorithms via MATLAB simulation.
- Comparing the results with other related works to demonstrate the improvements.

1.3 Contributions

The goal of this work is to develop novel ADP/RL-based control structures for continuous-time nonlinear systems with uncertainties and disturbances. The contributions of Chapter 3 and Chapter 4 are as follows:

Time-varying RISE-based RL control of nonlinear systems: This work proposes a new structure ARL-based robust control scheme for second-order nonlinear MIMO systems. The introduction of time-varying RISE and learning-based control guarantees tracking performance under several assumptions on the nonlinearities and uncertainties of the system. The time-varying RISE is constructed by replacing static feedback gains in the original RISE control law with nonlinear ones as functions of system variables. In addition, adaptive reinforcement learning (ARL) is employed to achieve adaptive optimal tracking performance for a transformed autonomous system via defining a sliding variable. The main contribution of this thesis is the use of time-varying RISE, in conjunction with the AC algorithm to guarantee robust tracking of a nonlinear system subjected to disturbances. Moreover, this work clarifies the initial conditions of the robot manipulator system and presents exploratory signal function, compared to [34]. All explicit variables and functions

in this work contribute to the numerical comparison between the two approaches. MATLAB simulation results on a 2-DOF robot arm demonstrate the improved performance of the time-varying RISE-based RL scheme in comparison with the original RISE-based RL controller.

Disturbance observer-based RL control of nonlinear systems: Considering nonlinear systems with unknown disturbances, this work proposes a disturbance observer-based RL control scheme. To facilitate RL algorithm, the kinematic and feed-forward controller is introduced to transform the original system into an autonomous one. On-policy AC architecture is used to address the optimal control problem for the augmented dynamic subsystem without disturbances and its aim is to stabilize the nonlinear plant and obtain the optimal value function. Additionally, a nonlinear disturbance observer is implemented in this study to attenuate the unknown disturbances and uncertainties of the system. The compensation control, together with the RL core, produces the robust optimal control input. Simulation results in the presence of disturbance on a surface vessel (SV) model show upgrades to the original approach in terms of tracking performance.

1.4 Thesis structure

The main contents are organized as

Chapter 1 introduces the general ideas of the work. The background, motivation, related work, and contributions of the thesis are systematically given.

Chapter 2 reviews the main literature of RL and shows how these techniques can be applied to tackle control problems. Then, disturbance attenuation methods are discussed in detail for later control design. Coordination between RL, optimal control, and disturbance attenuation methods are formed and implementation issues are highlighted.

Chapter 3 proposes an RL-based controller to obtain optimal tracking of a class of nonlinear systems with uncertainties and disturbances. While the sliding variable helps to achieve the reduced-order system, the novel time-varying RISE method contributes to the robust stability toward unexpected factors.

Chapter 4 proposes a robust optimal control structure for uncertain nonlinear systems with disturbances. The original nonlinear system is transformed into the autonomous form by designing a kinematic and feed-forward control scheme. Using the same AC architecture as the previous design, however, a DO is implemented to estimate the unknown disturbances. A combined control input including RL-based term and compensation term guarantees the robustness and stability of the closed-loop system.

Chapter 5 serves as a conclusion of the key ideas, contributions, and limitations of this work. It also sheds light on research directions and developments in the future.

Chapter 2

Literature Review

2.1 Reinforcement learning

RL, which was first noticed in the learning behavior of humans and other mammals, is defined differently in different works. Generally, an RL problem involves the existence of an agent that can interact with an environment by taking actions and collecting a reward from it. Sutton and Barto described RL as how to map situations to actions in order to maximize a numerical reward signal [21]. Apparently, maximizing a reward is corresponding to minimizing a cost, which is used more commonly on the subject of optimal control [20]. A mapping between situations and actions is called a policy, and the goal of RL is to learn an optimal policy in such a way that a predefined cost is minimized. Instead of involving a supervisor to instruct an agent on how to perform the best possible action, RL focuses on how the agent should adjust its behaviors toward the optimal one through interactions with the unknown environment. A typical RL iteration consists of two main stages. First, the agent interacts with the environment to assess the cost under the present policy. Policy evaluation is the name for this stage. Second, the agent implements a new policy based on the evaluated cost to further reduce the cost. This stage is known as policy improvement. Connections between optimal control and RL are formed and implementation topics are emphasized, which stimulate the approaches developed in this thesis.

2.1.1 Reinforcement learning methods

In normal cases, RL methods determine the performance index function or value function, which indicates how well a given action is applied for a given state. The meaning of value function is the long-term reward/penalty accumulated by the agent and for a deterministic MDP which can be defined as an infinite-horizon return with discount factor as [21]

$$V^u(x_0) = \sum_{k=0}^{\infty} \gamma^k r_{k+1} \tag{2.1}$$

for the discrete-time system,$x_{k+1} = f(x_k, u_k)$, $r_{k+1} \triangleq r(x_k, u_k)$ is the reward/penalty at the k^{th} step, x_k and u_k are the state and action, respectively, and $\gamma \in [0, 1)$ is the discount factor. The general objective of RL is to determine a policy that maximizes the value function. Using Bellman's equation, the unknown value function is

rewritten as [21]

$$V^u(x) = r(x,u) + \gamma V^u(f(x,u)) \tag{2.2}$$

where the index k is suppressed. Define the optimal value function as

$$V^*(x) = \min_u V^u(x) \tag{2.3}$$

Bellman's optimality principle states that "an optimal policy has the property that no matter what the previous control actions have been, the remaining controls constitute an optimal policy with regard to the state resulting from those previous controls." Therefore, (2.3) is rewritten as follows

$$V^*(x) = \min_u \left[r(x,u) + \gamma V^*(f(x,u))\right] \tag{2.4}$$

Supposing at time k, an arbitrary control u is applied and the optimal policy from time $k+1$ is on. The optimal control at time k according to Bellman's optimality principle is given as

$$u^*(x) = \arg\min_u \left[r(x,u) + \gamma V^*(f(x,u))\right] \tag{2.5}$$

The above backward recursion lays the foundation of all DP/RL methods: policy iteration, value iteration, and Q-learning [21]. There is another way of RL method classification: model-based and model-free. In, model-based or DP-based RL algorithms (2.4) and (2.5) are employed offline and complete environment information, as seen from (2.4) and (2.5), is required. In contrast, model-free RL methods are based on the temporal difference (TD), which denotes the difference between temporally successive estimates of the same quantity. These are online methods which do not utilize complete system model, instead, they exploit data collected from the process, that is, they learn by interacting with the environment. Several prevalent RL methods are described as follows.

2.1.1.1 Q-learning

Instead of approximating the cost functions of successive policies $V(x)$, Q-learning updates the Q-factors $Q(x,u)$ associated with an optimal policy, thereby precluding the various policy evaluation steps of PI [21]. The Q-iteration method finds the optimal Q-factor $Q^*(x,u)$ based on TD error as

$$Q(x,u) \leftarrow Q(x,u) + \alpha[r(x,u) + \gamma \min_a Q(\bar{x},a) - Q(x,u)] \tag{2.6}$$

The Q-learning method plays an important role in RL due to the use of the optimal action-value function which is independent of the current policy (also called off-policy), which significantly makes the convergence analysis of the algorithm easier. Moreover, another factor is that Q-learning method may not require the complete knowledge of system dynamics. Besides, for the convergence to Q^*, sufficient exploration is necessary. Then, from doing a greedy search on Q^* the optimal policy can be explicitly obtained as

$$u^*(x) = \arg\min_a Q^*(x,a) \tag{2.7}$$

2.1.1.2 Policy iteration

Policy iteration (PI) methods include two phases: policy evaluation and policy improvement. Beginning with an initial admissible policy, this algorithm estimates the value function (policy evaluation phase) and then uses a greedy search on the estimated value function to improve the policy (policy improvement phase). The PI algorithm, the main method used in this work, is expressed by the following steps.

Algorithm 2.1: Policy iteration (PI) algorithm

1. Initialize: Choose any admissible, i.e. stabilizing, control policy.

   Until convergence

2. Policy evaluation phase: Using the Bellman Equation to determine the value of the current policy

$$V^u(x) \leftarrow r(x,u) + \gamma V^u(f(x,u)) \tag{2.8}$$

3. Policy improvement phase: Improve the current policy using

$$\bar{u}(x) = \arg\min_a \left[r(x,a) + \gamma V^u(f(x,a))\right] \tag{2.9}$$

It can be seen from (2.8) and (2.9) that knowledge of the system dynamics $f(x,u)$ is required to implement PI method. Using an integral approach, the algorithm can be applied without the knowledge of system dynamics, which constructs a model-free method known as integral reinforcement learning (IRL) [28]. In addition, using TD learning, online PI algorithms may simultaneously/synchronously run policy evaluation and policy improvement steps; however, only under very restrictive conditions and sufficient exploration, they may converge to the optimal policy.

2.1.1.3 Value iteration

In value iteration (VI), starting from an arbitrary initial policy, the value function is directly improved by effectively combining the evaluation and the improvement steps into one single update using the following recurrence associations from DP [21]

$$V(x) \leftarrow \min_a \left[r(x,a) + \gamma V(f(x,a))\right] \tag{2.10}$$

The optimal $V^*(x)$ can be obtained with less computing resources than PI, however, PI algorithm typically requires fewer iterations for convergence.

2.1.2 Actor-critic architecture

2.1.2.1 Neural network implementation

In DP, at every iteration, the estimated value function is stored as a look-up table, and all the table records are updated with the entire state space. In fact, they become computationally intractable as the state space expands, which leads to the curse of dimensionality. Considering continuous spaces, the infinite number of states and actions makes the problem become even more complicated. The problem

is solved by value functions representation using function approximators via Stone-Weierstrass Theorem. Using linearly parameterized approximators, the function is fairly represented. It is stated that a single-layer neural network (NN) can simultaneously approximate a function and its derivative with a sufficiently large number of activation functions. A continuously differentiable function could be conveniently represented as

$$V(x) = W^T \psi(x) + \varepsilon(x) \tag{2.11}$$

where W is the unknown parameter vector, and $\psi(x)$ is a user-defined basis function, and $\varepsilon : \mathbb{R}^n \to \mathbb{R}$ denotes the function approximation error. The function approximation error, along with its derivative can be made arbitrarily small by increasing the number of basis functions.

It is crucial to choose appropriate basis functions which represent all the independent characteristics of the value function while solving the RL problem. It is noted that some prior information about the process can be embedded in the NN activation function. Moreover, the NN weight learning process makes use of optimization algorithms such as least squares, gradient descent, etc. Besides, deep NN can also be used as nonlinearly parameterized approximators; though, it is tougher to prove weight convergence in comparison with linearly parameterized network architectures.

2.1.2.2 Policy iteration actor-critic architecture

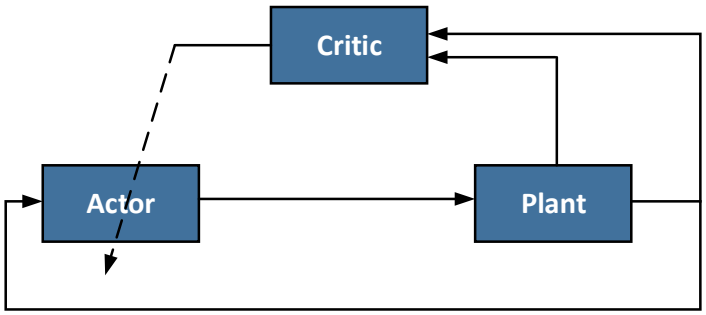


Figure 2.1: Online policy iteration based on AC architecture

The actor-critic architecture (see Figure 2.1) is one of the most widely used architectures to implement online RL algorithms. The actor can learn the control, where the estimate of value function is obtained by the critic. Using an adaptive update law designed as a differential equation, the actor and the critic weights are tuned continuously [30]. The actor can also be adjusted in order to minimize the Bellman error or the TD error. Simultaneously, the critic weights can be tuned by the TD method or using heuristic DP or its variants. Under appropriate PE conditions, they can converge to a neighborhood of the optimal value function and the optimal policy.

2.1.3 Infinite horizon optimal control

RL and optimal control are inextricably linked. This section presents the undiscounted infinite horizon optimal regulation problem for continuous-time nonlinear systems. First, a continuous-time nonlinear system is considered as

$$\dot{x} = F(x, u) \tag{2.12}$$

where $x(t) \in \mathcal{X} \subseteq \mathbb{R}^n$, $u(t) \in \mathcal{U} \subseteq \mathbb{R}^m$ is the control input. If $F : \mathcal{X} \times \mathcal{U} \to \mathbb{R}^n$ is Lipschitz continuous on $\mathcal{X} \times \mathcal{U}$ containing the origin, the solution $x(t)$ of the system in (2.12) is unique for any finite initial condition x_0 and the control $u \in \mathcal{U}$. Moreover, the system is stabilizable, in other words, the closed-loop system can be asymptotically stable by applying an appropriate continuous feedback control law $u(x(t))$.

For the system (2.12), the infinite-horizon scalar performance index can be described as

$$J(x(t)) = \int_t^\infty r(x(s), u(s))ds \tag{2.13}$$

where t is the initial time, $r(x, u) \in \mathbb{R}$ is the utilization function, written as

$$r(x, u) = Q(x) + u^T Ru \tag{2.14}$$

where $R \in \mathbb{R}^{m\times m}$ is a positive-definite symmetric matrix and $Q(x) \in \mathbb{R}$ is positive definite and continuously differentiable.

The objective is to find an admissible control $u^* \in \Psi(\mathcal{X})$, so that the performance index function in (2.13) associated with the system (2.12) is minimized.

It is noted that, an admissible control $u(t)$ refers to a continuous feedback control law $u(x(t)) \in \Psi(\mathcal{X})$, where $\Psi(\cdot)$ implies an admissible control set, which guarantees the asymptotic stability of the system (2.12) on $\mathcal{X}$, $u(0) = 0$, and $J(\cdot)$ in (2.13) is finite.

Then, the optimal value function can be determined as

$$V^*(x(t)) = \min_{\substack{u(\tau)\in\Psi(\mathcal{X}) \\ t\leq\tau<\infty}} \int_t^\infty r(x(s), u(x(s)))ds \tag{2.15}$$

Given the continuously differentiable value function, Bellman's principle of optimality can be used to obtain the subsequent optimality condition

$$0 = \min_{u(t)\in\Psi\ (\mathcal{X})} \Big[r(x, u) + \frac{\partial V^*(x)}{\partial x} F(x, u) \Big] \tag{2.16}$$

This is a partial differential equation for the optimal cost $V^*(x)$. It is called the Hamilton–Jacobi–Bellman (HJB) equation. With the assumption that $V^*(x)$ is continuously differentiable, the HJB in (2.16) provides a way to acquire the optimal control policy $u^*(x)$. Using (2.14) and (2.16), the optimal feedback control can be obtained as

$$u^*(x) = -\frac{1}{2} R^{-1} \frac{\partial F(x, u)}{\partial u}^T \frac{\partial V^*(x)}{\partial x}^T \tag{2.17}$$

For the continuous-time nonlinear affine system

$$\dot{x} = f(x) + g(x)u \tag{2.18}$$

where $f(x) \in \mathbb{R}^n$ and $g(x) \in \mathbb{R}^{n\times m}$, (2.17) can be rewritten as

$$u^*(x) = -\frac{1}{2} R^{-1} g^T(x) \frac{\partial V^*(x)}{\partial x}^T \tag{2.19}$$

The HJB in (2.16) can be rewritten by replacing for the immediate cost in (2.14), the system in (2.18) and the optimal control in (2.19), as

$$\begin{aligned} 0 &= Q(x) + \frac{\partial V^*(x)}{\partial x} f(x) - \frac{1}{4} \frac{\partial V^*(x)}{\partial x} g(x) R^{-1} g^T(x) \frac{\partial V^*(x)^T}{\partial x} \\ 0 &= V^*(0) \end{aligned} \tag{2.20}$$

The optimal policy in (2.19) can be revealed with the understanding of the optimal value function $V^*(x)$, which is the solution of the HJB equation in (2.20). In fact, the nonlinear HJB equation is very difficult to be solved in general and sometimes, does not have an analytical solution.

2.1.4 Reinforcement learning and adaptive optimal control

On the one hand, adaptive control offers methods to design controllers which are able to learn or adapt online to the uncertainties in system dynamics by the way of minimizing the output error (e.g., least squares or gradient descent methods). Nevertheless, traditional adaptive control theories do not target optimality in the sense of maximizing a performance function in the long run.

On the other hand, most optimal control approaches are offline and need explicit model knowledge. The Riccati equation in linear systems is solved offline and requires precise information of the system dynamics. In the context of control engineering, ADP and RL bridge the gap between conventional optimal control and adaptive control algorithms [35]. Adaptive optimal control methods learn the optimal policy and value function for a physical system. Unlike conventional optimal control, RL intends to solve the nonlinear HJB equation online in real time. In contrast, unlike classical adaptive controllers, both stability and optimality properties are the main concern for the closed-loop system. This has fostered the idea of adaptive autonomy in an optimal aspect by developing ADP/RL-based controllers [25].

In MDPs, RL algorithms described in Section 2.1.1 have been successfully implemented to find optimal policies in uncertain environments, for example, Q-learning based on TD is an online model-free RL method. It is argued that RL is a direct adaptive optimal control technique. The nature of RL algorithms is discrete so that numerous methods have been suggested for adaptive optimal control of discrete-time systems. In fact, for continuous-time systems, RL implementation is not as explicit as in the former problem, because while the discrete-time TD error is model-free, the continuous-time TD error formula fundamentally involves full knowledge of the system dynamics (2.16). Based on the model-based TD method, RL techniques for continuous-time systems are introduced. Moreover, RL-based controllers encounter many other issues: closed-loop stability, function approximation, optimal convergence performance, and the tradeoff between exploitation and exploration. A number of studies have thoroughly addressed these difficulties, which play an important role in the effective application of RL-based control techniques. Generally, this thesis is inspired by the need to deliver innovation on RL-based feedback control structures and discover their potential as adaptive optimal control concepts.

2.2 Disturbance attenuation in control systems

In reality, "disturbances" is a generalized concept, which may include external disturbances, unmodeled dynamics, and parameter perturbations. They broadly exist in marine engineering (e.g. boats, ships, hovercraft and submarines), aerospace engineering such as aircrafts, missiles and satellites, and also many other engineering systems [13]. In general, disturbance is key factor that degrades the control system performance. The importance of disturbance attenuation and rejection in control system design is undeniable [3]. Many classical controls with fixed parameters cannot guarantee fast response and high precision in the presence of disturbances. Therefore, different ways have been carried out to deal with disturbances.

On the one hand, adaptive control is a powerful tool for structured uncertainties [1]. Nevertheless, various kinds of disturbances may potentially deteriorate the performance, or even produce instability. Many researchers have looked at robust control as an alternative. Among that, sliding mode control (SMC) can cope with all bounded modeling uncertainties and guarantee asymptotic tracking performance [4]. However, it is difficult for practical application because of the chattering nature of SMC. Addressing this problem prompts many strategies, and high-order SMC is a primary solution [5]. Furthermore, the continuous RISE control strategy developed in [36], which includes a unique integral signum feedback term, can cover sufficiently smooth bounded disturbances. This new control structure contributes to asymptotic stability which can be achieved regardless of existing modeling uncertainties. By this technique, many different problems [14, 37] have been accomplished.

On the other hand, combining control strategies with disturbance estimation methods becomes an attractive strategy, which can deal with greater uncertainties. Among these, the disturbance observer (DO) has been employed with robust control [18], for adaptive robust control [17]. Not only external disturbances but also internal unmodeled dynamics and unknown uncertainties are within the estimation range of DO.

2.2.1 RISE control for nonlinear systems

A novel control structure called robust integral of the sign of the error (RISE) has been introduced in [36] to deal with multi-input multi-output (MIMO) high-order nonlinear systems. This non-model-based continuous control mechanism can guarantee a semi-global asymptotic tracking under some restricted assumptions on the system. Moreover, due to the high robustness and disturbances attenuation, RISE-based control has been implemented in diverse real-time applications. It is noteworthy that RISE and RISE-based control has been modified in several directions. Upgrading this control law with nonlinear parameters may provide the ability to compensate for more degrees of high nonlinearities in most systems [37]. Another way is integrating RISE with other control frameworks like ADP/RL in this thesis, which may bring in wider applications.

2.2.1.1 Background on RISE control

First, we examine a first-order single-input nonlinear system having the general form

$$m(\eta)\dot{\eta} + f(\eta) = u \tag{2.21}$$

where $\eta(t) \in \mathbb{R}$ is the system state, $u(t) \in \mathbb{R}$ is the control input, and $m(\eta)$, $f(\eta) \in \mathbb{R}$ are uncertain nonlinear function. It is assumed that $m(\eta)$ and $f(\eta)$ satisfy the following assumptions

Assumption 2.1. *The positive function $m(\eta)$ is bounded*

$$\underline{m} \leq m(\eta) \leq \bar{m}(\eta) \tag{2.22}$$

where $\bar{m}(\eta) \in \mathbb{R}$ denotes a positive non-decreasing function, and $\underline{m} \in \mathbb{R}$ denotes a positive constant.

Assumption 2.2. *The functions $m(\eta)$ and $f(\eta)$ are second-order differentiable with respect to $\eta(t)$ such that*

$$\begin{aligned} m(\eta), \frac{\partial m(\eta)}{\partial \eta}, \frac{\partial^2 m(\eta)}{\partial \eta^2} \in \mathcal{L}_\infty \quad \text{if} \quad \eta(t) \in \mathcal{L}_\infty \\ f(\eta), \frac{\partial f(\eta)}{\partial \eta}, \frac{\partial^2 f(\eta)}{\partial \eta^2} \in \mathcal{L}_\infty \quad \text{if} \quad \eta(t) \in \mathcal{L}_\infty \end{aligned} \tag{2.23}$$

Let $\eta_d(t) \in \mathbb{R}$ be a desired trajectory that is continuously differentiable up to its third derivative, i.e.

$$\frac{d^i \eta_d(t)}{dt^i} \in \mathcal{L}_\infty, \quad i = 0, 1, 2, 3 \tag{2.24}$$

The tracking error $e(t) \in \mathbb{R}$ is defined as follows

$$e \triangleq \eta_d - \eta \tag{2.25}$$

Our objective is to obtain asymptotic tracking with a continuous control law employing (2.24) and norm-based, inequality bounds on the functions $\frac{\partial^i m(\eta_d)}{\partial \eta_d^i}$ and $\frac{\partial^i f(\eta_d)}{\partial \eta_d^i}$, $i = 0, 1, 2$.

Remark 2.1. *For simple structure, we have assumed $m(\eta)$ and $f(\eta)$ do not depend explicitly on time or on unknown time-varying parameters. Yet, it should be highlighted that the proposed control law can compensate for these problems if the time-varying components satisfy second-order differentiability conditions like in (2.23). Therefore, the functions $m(\eta)$ and $f(\eta)$ could be presented by $m(\eta, \theta_1(t), t)$ and $f(\eta, \theta_2(t), t)$ where $\theta_i(t)$, $i = 1, 2$ refer to unknown time-varying parameter vectors and other time-varying disturbances that may show up nonlinearly in the model.*

RISE control equation that can accomplish the control objective is generally designed as

$$u(t) = (k_s + 1) e(t) - (k_s + 1) e(t_0) + \int_{t_0}^{t} \left[(k_s + 1) \alpha e(\tau) + \beta \operatorname{sgn}(e(\tau)) \right] d\tau \tag{2.26}$$

where k_s, α, $\beta \in \mathbb{R}$ are positive control gains, t_0 is the initial time, and $\operatorname{sgn}(\cdot)$ denotes the standard signum function. The condition for the control law of (2.26) to ensure asymptotic tracking is that the control gains k_s and β are chosen sufficiently large relative to the norm of the initial tracking error and a desired trajectory-based bound, respectively [36, 37]

$$k_s > \frac{1}{4\lambda_3} {\rho_0}^2 \left(\sqrt{\frac{\lambda_2 (\eta(t_0))}{\lambda_1}} \eta_0 \right) \tag{2.27}$$

$$\beta > \|N_d(t)\|_\infty + \frac{1}{\alpha}\left\|\dot{N}_d(t)\right\|_\infty \tag{2.28}$$

Based on the stability analysis introduced in [36] and [37], the signal (2.26) has the ability to learn the unknown system model.

Remark 2.2. *The authors in [36] indicated that this concept is then developed to higher-order, multi-input systems.*

2.2.1.2 Time-varying RISE approach

The beginning structure in (2.26) can be separated into two parts: a linear feedback part depending on the filtered error e, and a nonlinear signum function. The linear part contains proportional and integral terms on the filtered error, which is related to a PI controller but considering the filtered error instead of the position error. The two linear control terms may result in degraded performance when coping with high nonlinearity during critical dynamic operation. Moreover, they are considerably sensitive to disturbances and limited in tuning abilities.

The idea in [37] is to provide the control law with nonlinear time-varying gains instead of the proportional and the integral static feedback ones. Hence, the time-varying feedback RISE controller is obtained as

$$u(t) = (K_s(\cdot) + 1)\, e(t) - (K_s(t_0) + 1)\, e(t_0) + \int_{t_0}^{t} \left[(k_{s0} + 1)\, \alpha(\cdot) e(\tau) + \beta\, \mathrm{sgn}(e(\tau))\right] d\tau \tag{2.29}$$

with $K_s(\cdot)$ and $\alpha(\cdot)$ are two nonlinear feedback functions designed as

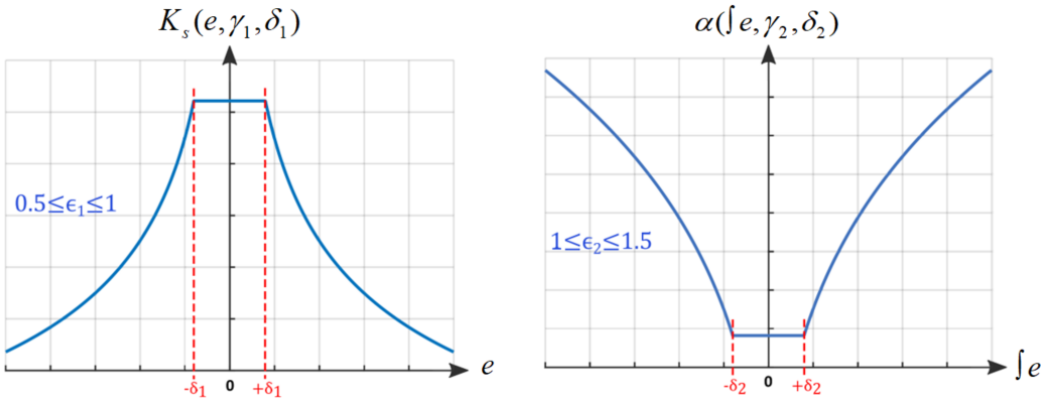
$$K_s(\cdot) \equiv K_s(e, \gamma_1, \delta_1) = \begin{cases} k_{s0}|e|^{\gamma_1 - 1}, |e| > \delta_1 \\ k_{s0}{\delta_1}^{\gamma_1 - 1}, |e| \leq \delta_1 \end{cases} \tag{2.30}$$

$$\alpha(\cdot) \equiv \alpha(e, \gamma_2, \delta_2) = \begin{cases} \alpha_0|\int e|^{\gamma_2 - 1}, |\int e| > \delta_2 \\ \alpha_0{\delta_2}^{\gamma_2 - 1}, |\int e| \leq \delta_2 \end{cases} \tag{2.31}$$

where $k_{s0}, \alpha_0, \gamma_1, \delta_1, \gamma_2, \delta_2$ are positive parameters which need to be designed carefully. Particularly, the two parameters, γ_1 and γ_2, should be selected within the intervals $[0.5, 1]$ and $[1, 1.5]$ respectively to match the desired performance.

The choice of γ_1 within the interval [0.5, 1] can weaken the proportional gain $K_s(\cdot)$ at great filtered error values and boost it at small ones (see Figure 2.2(a)). And, the proportional gain remains constant at a maximum saturated value when the filtered error stays within the narrow region $[-\delta_1, \delta_1]$ around zero. It is worth noting that the combined error provides information on both position and velocity errors. As a result, this gain scheduling might lead to a quick transition of closed-loop system states and beneficial damping.

The nonlinear feedback parameter $\alpha(\cdot)$ changes as a function of the integral of the filtered error (see Figure 2.2(b)), implying that $\alpha(\cdot)$ is more involved in steady-state filtered errors (errors that persist with time). Within the interval $[1, 1.5]$, high integral gain is obtained for significant steady-state filtered errors, while small integral gain is obtained for small ones. When this error stays within the small interval $[-\delta_2, \delta_2]$ around zero, the integral parameter does not vary. Thus, this change may boost the tracking performance towards the reference point and additionally, deal



(a) $K_s(\cdot)$ (b) $\alpha(\cdot)$

Figure 2.2: Functions of the control gains with respect to their arguments

with the integral windup problem as precluding the integral term from accumulating around particular bounds.

By selecting γ_1 and γ_2 in their corresponding intervals, there exists globally bounded nonlinear functions as follows

$$0 < K_{sm} \triangleq k_{s0} \|e\|_\infty^{\gamma_1 - 1} \leq K_s(\cdot) \leq k_{s0} \delta_1^{\epsilon_1 - 1} \triangleq K_{sM} \tag{2.32}$$

$$0 < \alpha_{2m} \triangleq \alpha_{20} \delta_2^{\gamma_2 - 1} \leq \alpha_2(\cdot) \leq \alpha_{20} \left\| \int e \right\|_\infty^{\lambda_2 - 1} \triangleq \alpha_{2M} \tag{2.33}$$

where $\|\cdot\|_\infty$ indicates the infinity-norm.

This time-varying version of RISE may improve the controller's global tracking efficiency and robustness to system uncertainties and parameter variation. It is worth noting that the nonlinear function structure is not sophisticated to implement in real-time experiments.

Theorem 2.1. *The control law in* (2.29) *employed to the second-order nonlinear MIMO system (extended from* (2.21)*) ensures that all the system signals are asymptotically stable with the appropriate choices of the control gains.*

Proof. Please refer to [37] for details. □

2.2.2 Disturbance observer-based control for nonlinear systems

The disturbance observer (DO) is introduced in this section to approximate system's disturbances/uncertainties. Under some situations, the DO error has been shown to be exponentially stable [38]. The disturbance observer-based control (DOBC) approach has two different features compared to other robust control schemes [18]. One feature is that DO-based compensation can be viewed as a "patch" for main controllers that can guarantee good stability and tracking performance but have inconsiderable disturbance rejection and robustness against uncertainties. One benefit is that the primary reputable controller, such as traditional flight control systems, does not need to be changed. After the baseline controller is ideally designed, the DO-based correction is added to improve the robustness and disturbance attenuation. Instead of implementing a completely new and distinct control system that necessitates a new verification and certification procedure, the DOBC verification

may be built on top of the existing verification process to assure safety and reliability. The other feature is that disturbance observer-based control (DOBC) is not designed based on worst cases. Many current methods are worst case-based designs (e.g. robust control) and have been stated as being "over-conservative". In most cases, promised robustness comes at the cost of deteriorated nominal performance. With the DOBC approach, the nominal performance of the baseline controller is maintained while encountering uncertainties and disturbances [39].

2.2.2.1 Nonlinear disturbance observer

A typical MIMO affine nonlinear system with combined disturbances is represented as

$$\dot{X} = F(X) + G_u(X)u + G_d(X)\Delta \tag{2.34}$$

where $X \in \mathbb{R}^n, u \in \mathbb{R}^m, \Delta \in \mathbb{R}^l$ are the state vector, control input, and disturbance vector variables. It is reasonable to assume that $F(X)$, $G_u(X)$, and $G_d(X)$ are smooth functions.

In [39] and [38], a nonlinear disturbance observer (NDO) was designed to estimate unknown disturbances/uncertainties .

$$\begin{cases} \hat{\Delta} = y + P(X) \\ \dot{y} = -\frac{\partial P(X)}{\partial X}(G_u(X)u + G_d(X)y + G_d(X)P(X) + F(X)) \end{cases} \tag{2.35}$$

where $\hat{\Delta}$ is the estimation of the mismatched disturbances/uncertainties, $y \in \mathbb{R}^l$ is the auxiliary variable which represents the internal dynamics of the nonlinear observer, $P(X) \in \mathbb{R}^l$ is a design nonlinear function for observation efficiency. Define the disturbance estimation error as

$$\tilde{\Delta} = \Delta - \hat{\Delta} \tag{2.36}$$

Assumption 2.3. *The combined disturbance Δ varies slowly over time, i.e., $\dot{\Delta} \simeq 0$.*

Remark 2.3. *Under the assumption that the disturbances fluctuate slowly in relation to the dynamics of the observer (see Assumption 2.3), NDO has been shown to have strict asymptotical convergence. It is worth noting that the observer* (2.35) *can track certain fast time-varying disturbances with a bounded error if the derivative of the disturbances is bounded [17].*

Remark 2.4. *The combined disturbances would depend on the states in the presence of uncertainties, which can be reasonably approximated if the DO dynamics are quicker than the closed-loop dynamics. The state observer-based control methods can be justified on the same grounds.*

Theorem 2.2. *For system* (2.34)*, the disturbance observer is given as* (2.35)*. If $H(X) = (\partial P(X)/\partial X\ )G_d(X)$ is positive definite then the disturbance observer can exponentially track the disturbance, i.e., the disturbance estimation error $\tilde{\Delta}$ is exponentially stable, $\forall X$.*

Proof. A Lyapunov function candidate is chosen as

$$\Phi = \frac{1}{2}\tilde{\Delta}^T\tilde{\Delta} \tag{2.37}$$

Differentiation of the Lyapunov function with respect to time gives

$$\begin{aligned}\dot{\Phi} &= -\tilde{\Delta}^T(\dot{y} + \dot{P}(X)) \\ &= \tilde{\Delta}^T \frac{\partial P}{\partial X}(G_d y + G_d P + F(X) + G_u u) - \tilde{\Delta}^T \frac{\partial P}{\partial X}\dot{X}\end{aligned} \tag{2.38}$$

Combining with (2.34) yields

$$\begin{aligned}\dot{\Phi} &= -\tilde{\Delta}^T(\dot{y} + \dot{P}(X)) \\ &= \tilde{\Delta}^T \frac{\partial P}{\partial X}(G_d y + G_d P + F(X) + G_u u) - \tilde{\Delta}^T \frac{\partial P}{\partial X}(F(X) + G_u u + G_d \Delta) \\ &= \tilde{\Delta}^T \frac{\partial P}{\partial X} G_d(\hat{\Delta} - \Delta) \\ &= -\tilde{\Delta}^T \frac{\partial P}{\partial X} G_d \tilde{\Delta}\end{aligned} \tag{2.39}$$

As $H(X) = \frac{\partial P(X)}{\partial X} G_d(X)$ is positive definite, the (2.39) satisfies $\dot{\Phi} \leq -2\lambda_{\min}(H)\Phi$ which means that $\Phi(t) \leq \Phi(0)e^{-2\lambda_{\min}(H)t}$. Therefore, the disturbance observer error $\tilde{\Delta}$ is exponentially stable, as $t \to \infty$. □

Remark 2.5. *From Theorem 2.2, it is clear that Assumption 2.3 is the guarantee of the exponential stability of disturbance estimation error. If* $\dot{\Delta} \neq 0$*, then* $\dot{\Phi} = \tilde{\Delta}^T \dot{\Delta} - \tilde{\Delta}^T \frac{\partial P}{\partial X} G_d \tilde{\Delta}$*. Therefore, the condition* $\dot{\Delta} = 0$ *is considered in this work.*

2.2.2.2 Disturbance observer-based control

Remark 2.6. *In the case of mismatched disturbances, the NDO* (2.35) *is appropriate. However, because the disturbances may not be in the same channels as the control inputs, the NDO estimates cannot be used to compensate for them directly.*

Generally, the effect of the mismatched disturbances cannot be erased from state variables [38]. In [39], based on the estimated disturbance from NDO (2.35), a combined control input is designed as

$$u = r(X) + d(X)\hat{\Delta} \tag{2.40}$$

By constructing a proper compensation gain $d(X)$, this control law may eliminate the combined disturbance's effect from the output. As a result, the DO-based control (DOBC) strategy's application fields will be significantly expanded.

It is worth noting that the disturbance compensation term $\hat{d}(X)\Delta$ in (2.40) is exclusively meant for disturbances, implying that the NDO only functions if and only if disturbances exist. As a result, it simply acts as a "patch" for the current controller, enhancing its disturbance attenuation and robustness to uncertainties. Then, under disturbances and uncertainties, the closed-loop system performs its nominal performance.

Theorem 2.3. *The closed-loop system consists of a nonlinear system* (2.34)*, combined control law* (2.40) *and NDO* (2.35) *is ISS.*

Proof. For more details, please refer to [39]. □

The following is a general NDO-based robust control design technique for system (2.34) with disturbances:

Algorithm 2.2: DOBC design procedure

1. Design a nonlinear feedback controller as a preliminary step for achieving stability and performance requirements without considering disturbances or uncertainties.

2. Consider together the external disturbances and the impact of the uncertainties and then construct an NDO to estimate the combined disturbances.

3. Implement the general NDO-based robust controller by combining the nonlinear feedback controller and the DO-based compensation term with an appropriate gain to accomplish target performance specification for the nonlinear system with disturbances.

Chapter 3

Time-Varying RISE-Based Reinforcement Learning Control of Nonlinear Systems

This work proposes a new structure ARL-based robust control scheme for second-order nonlinear MIMO systems. RISE algorithm has the ability to learn the unknown model uncertainties and external disturbances. RISE control law with sliding variables guarantees tracking performance under restricted assumptions on the uncertainties and nonlinearities of the system. The time-varying RISE is constructed by replacing static feedback gains in the original RISE control law with nonlinear ones as functions of system variables. The concept is based on that nonlinear time-varying gains improve overall efficiency by compensating for a variety of nonlinearities and additive disturbances. In addition, ARL is employed to obtain optimal tracking performance for the uncertain/disturbed nonlinear robot system. The HJB equation is solved by an iterative method using the online actor-critic ADP technique based on neural networks. MATLAB simulation results on a 2-DOF robot arm demonstrate the improved performance of the time-varying RISE-based RL scheme in comparison with the original RISE-based RL controller.

3.1 Problem formulation

The dynamic model of an n-link robot manipulator can be given in the Lagrange form

$$M(\eta)\ddot{\eta} + C(\eta,\dot{\eta})\dot{\eta} + G(\eta) + F(\dot{\eta}) + d(t) = \tau(t) \tag{3.1}$$

where $\eta \in \mathbb{R}^n$ vector of joint variables, $M(\eta) \in \mathbb{R}^{n\times n}$ is a generalized inertia matrix, $C(\eta,\dot{\eta}) \in \mathbb{R}^{n\times n}$ is a generalized Coriolis/centripetal matrix, $G(\eta) \in \mathbb{R}^n$ is gravity forces, $F(\dot{\eta}) \in \mathbb{R}^n$ is a generalized friction, $d(t)$ is disturbance vector, $\tau(t)$ is the vector of control inputs.

The considered robot manipulator belongs to the class of Euler-Lagrange systems [40], which has the property that the inertia symmetric matrix $M(\eta)$ is positive definite, and satisfies $\forall \xi \in \mathbb{R}^n$

$$\begin{aligned} &\underline{m}\|\xi\|^2 \le \xi^T M(\eta)\xi \le \bar{m}(\eta)\|\xi\|^2 \\ &\xi^T(\dot{M}(\eta) - 2C(\eta,\dot{\eta}))\xi = 0 \end{aligned} \tag{3.2}$$

where $\bar{m}(\eta) \in \mathbb{R}$ is a positive non-decreasing function with respect to η and $\underline{m} \in \mathbb{R}$

is a positive constant. And, $\|\cdot\|$ stands for the classical Euclidean norm. There are several assumptions that will be employed during stability analysis.

Assumption 3.1. *The reference/desired trajectory $\eta_d(t)$ and its first, second, third and fourth time derivatives exist and are bounded.*

Assumption 3.2. *The disturbance vector $d(t)$ and its time derivatives $\dot{d}(t)$ are bounded by known constants.*

Assumption 3.3. *Provided that $\eta(t), \dot{\eta}(t) \in L_\infty$, then $C(\eta, \dot{\eta})$, $F(\dot{\eta})$, $G(\eta)$ and their first, second partial derivatives of with respect to $\eta(t)$ and of the derivatives of $C(\eta, \dot{\eta})$, $F(\dot{\eta})$ with respect to $\dot{\eta}(t)$ exist and are bounded.*

The control objective is that the system precisely follows a desired time-varying trajectory $n_{ref}(t)$ under dynamic uncertainties and disturbances by making use of the background of online ARL-based control design and disturbance attenuation method.

Inheriting the framework of sliding mode control (SMC), the sliding variable is written as

$$s(t) = \dot{e}_1 + \alpha_1 e_1 \tag{3.3}$$

where $e_1(t) = \eta_{ref} - \eta$, $\alpha_1 \in \mathbb{R}^{n\times n} > 0$, and the corresponding sliding surface is

$$S = \{e_1(t) \in \mathbb{R}^n : s(t) = 0\} \tag{3.4}$$

Substituting the sliding variable into the system model (3.1), the dynamics of $s(t)$ is obtained

$$M\dot{s} = -Cs - \tau + f + d \tag{3.5}$$

where the nonlinear function $f(\eta, \dot{\eta}, \eta_{ref}, \dot{\eta}_{ref}, \ddot{\eta}_{ref})$ is defined as follows

$$f = M(\ddot{\eta}_{ref} + \alpha_1 \dot{e}_1) + C(\dot{\eta}_{ref} + \alpha_1 e_1) + G + F \tag{3.6}$$

Remark 3.1. *The sliding variable contributes to the reduction of the order of uncertain/disturbed robot manipulator systems. The achieved first-order continuous-time nonlinear autonomous system facilitates the adaptive reinforcement learning algorithms. Moreover, the nonlinear function $f(\cdot)$ and the external disturbance $d(t)$ will be compensated by the design of time-varying RISE, which is presented in the next section.*

3.2 Adaptive reinforcement learning of nonlinear systems based on time-varying RISE

3.2.1 On-policy actor-critic architecture-based algorithm

With the robot manipulator described in (3.5), design the control input as follows

$$\tau = f + d - u \tag{3.7}$$

where the term u deals with adaptive optimal control problem based on ARL and the remaining term $f + d$ will handled by time-varying RISE framework later. Then, the dynamics in (3.5) is rewritten as

$$M\dot{s} = -Cs + u \tag{3.8}$$

Combining (3.3) and (3.8), the time-varying model of the manipulator is obtained:

$$\dot{x} = \begin{bmatrix} -\alpha_1 e_1 + s \\ -M(\eta_{ref} - e_1)^{-1} C(\eta_{ref} - e_1, \dot{\eta}_{ref} + \alpha_1 e_1 - s)s \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{n\times n} \\ M^{-1} \end{bmatrix} u \tag{3.9}$$

where $x = [e_1^T, s^T]^T$, and the infinite-horizon scalar cost function to be minimized is

$$J(x,u) = \int_0^\infty (\frac{1}{2}x^T Q x + \frac{1}{2}u^T R u)dt \tag{3.10}$$

where $Q \in \mathbb{R}^{2n\times 2n}$ and $R \in \mathbb{R}^{n\times n}$ are positive definite symmetric matrices.

The next step is to define an augmented state $X(t)$ for system transformation, which helps to avoid the time-dependent systems. Therefore, under the important assumption that the reference trajectory $\eta_{ref}(t)$ satisfies $\dot{\eta}_{ref}(t) = f_{ref}(\eta_{ref})$, the ARL is employed to find the optimal policy for the autonomous affine state-space model:

$$\dot{X} = A(X) + B(X)u \tag{3.11}$$

with $\dot{X} = [x^T, {\eta_{ref}}^T, {\dot{\eta}_{ref}}^T, {\ddot{\eta}_{ref}}^T]$ and the matrices:

$$A(X) = \begin{bmatrix} -\alpha_1 e_1 + s \\ -M(\eta_{ref} - e_1)^{-1} C(\eta_{ref} - e_1, \dot{\eta}_{ref} + \alpha_1 e_1 - s)s \\ f_{ref}(\eta_{ref}) \\ \dot{f}_{ref}(\eta_{ref}) \end{bmatrix} \tag{3.12}$$

$$B(X) = \begin{bmatrix} \mathbf{0} \\ M^{-1} \\ \mathbf{0} \end{bmatrix} \tag{3.13}$$

The corresponding infinite horizon scalar cost function to be minimized is defined as follows

$$J(X,u) = \int_t^\infty (\frac{1}{2}X^T Q_T X + \frac{1}{2}u^T R u)d\tau \tag{3.14}$$

where

$$Q_T = \begin{bmatrix} Q & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{3.15}$$

In the first step of RL employing procedure, to guarantee the stability of adaptive optimal control structure, considering the admissible policy $u(X)$ which was described thoroughly in [21, 22, 25].

Now, the optimal control objective is to find an admissible control input $u^*(X)$ so that the infinite horizon cost function (3.14) associated with the affine system (3.11) is minimized.

With the optimal value function $V^*(X)$ defined in (2.15), based on the significant works of [21] and [25] in ADP and RL, the optimal feedback controller $u^*(X)$ is given as (2.17)

$$u^*(X) = -\frac{1}{2}R^{-1}B^T(X)\frac{\partial V^*(X)^T}{\partial X}u^*(X) \tag{3.16}$$

The Hamiltonian of the system in (3.11) is derived as

$$H\Big(X, u, \frac{\partial V}{\partial X}\Big) = \frac{\partial V}{\partial X}\left(A(X) + B(X)u\right) + \frac{1}{2}X^T Q_T X + \frac{1}{2}u^T R u \tag{3.17}$$

The optimal value function $V^*(X)$ in (2.15) and the corresponding optimal policy $u^*(X)$ in (3.16) satisfy the HJB equation $H\left(X, u^*, \frac{\partial V^*}{\partial X}\right) = 0$.

Because it is difficult to explicitly solve the HJB equation, a typical RL/ADP-based method is used, that is actor-critic NNs architecture. The optimal cost function and the optimal control is represented using a NN [30]:

$$V^*(X) = W^T\psi(X) + \varepsilon_v(X) \tag{3.18}$$

$$u^*(X) = -\frac{1}{2}R^{-1}B^T(X)\Big(\big(\frac{\partial\psi}{\partial X}\big)^T W + \big(\frac{\partial\varepsilon_v(X)}{\partial X}\big)^T\Big) \tag{3.19}$$

where $W \in \mathbb{R}^N$ is vector of unknown ideal NN weights, N is the number of neurons, $\psi(X) \in \mathbb{R}^N$ is a smooth basis function vector, and $\varepsilon_v(X) \in \mathbb{R}$ is the function reconstruction error.

The approximate solution of (3.18) is obtained through NN updating laws without the requirement of solving the HJB equation (see [30] for more details). Furthermore, the smooth basis function is chosen depending on the characteristics of the robot manipulator (see Section 3.3.1). In [30], based on the Weierstrass approximation theorem, NNs can uniformly approximate $V^*(X)$ and $\frac{\partial V^*(X)}{\partial X}$ with $\varepsilon_v(X)$, $\frac{\partial\varepsilon_v(X)}{\partial X} \to 0$ as $N \to \infty$.

With a fixed number of neurons N, separate the critic $\hat{V}(X)$ and the actor $\hat{u}(X)$ approximation as

$$\hat{V}(X) = \hat{W}_c^T\psi(X) \tag{3.20}$$

$$\hat{u}(X) = -\frac{1}{2}R^{-1}B^T(X)\Big(\frac{\partial\psi}{\partial X}\Big)^T\hat{W}_a \tag{3.21}$$

The adaptation of critic $\hat{W}_c$ and actor $\hat{W}_a$ weights are simultaneous and proposed to minimize the Bellman error defined as

$$\begin{aligned}\delta_{hjb} &= \hat{H}\Big(X, \hat{u}, \frac{\partial\hat{V}}{\partial X}\Big) - H^*\Big(X, u^*, \frac{\partial V^*}{\partial X}\Big)\\ &= \hat{W}_c^T\sigma + \frac{1}{2}X^TQ_TX + \frac{1}{2}\hat{u}^TR\hat{u}\end{aligned} \tag{3.22}$$

with $\sigma(X, \hat{u}) = \frac{\partial\psi}{\partial X}(A + B\hat{u})$ is the regression vector of the critic. Equivalent to the work in [30], the least-squares update of critic weights is given as

$$\frac{d}{dt}\hat{W}_c = -k_c\lambda\frac{\sigma}{1 + \upsilon\sigma^T\lambda\sigma}\delta_{hjb} \tag{3.23}$$

where k_c and υ are constant positive gains, and $\lambda \in \mathbb{R}^{N\times N}$ is a symmetric estimation gain matrix satisfying

$$\frac{d}{dt}\lambda = -k_c\lambda\frac{\lambda\sigma^T}{1 + \upsilon\sigma^T\psi\sigma}\lambda \tag{3.24}$$

It is important to make sure $\lambda(t)$ is positive definite, which prevents the covariance wind-up problem [30].

$$\varphi_1 I \le \lambda(t) \le \varphi_0 I \tag{3.25}$$

Unlike the critic, the actor adaptation law is based on gradient descent method:

$$\frac{d}{dt}\hat{W}_a = -\frac{k_{a1}}{\sqrt{1 + \sigma^T\sigma}}\frac{\partial\psi}{\partial X}BR^{-1}B^T\frac{\partial\psi^T}{\partial X}\big(\hat{W}_a - \hat{W}_c\big)\delta_{hjb} - k_{a2}\big(\hat{W}_a - \hat{W}_c\big) \tag{3.26}$$

PE conditions of the regression vector of the critic are crucial for the adaptive control to converge to the optimal solution. Unlike linear systems, where PE conditions refer to the external input's adequate richness, there has been no reliable technique to assure PE conditions in nonlinear problems until now. In the initial stage of the learning process, an exploratory signal $n(t)$ consisting of sinusoids of different frequencies is introduced to the control to guarantee PE quality.

Remark 3.2. *Unlike the work in [30], the identifier design is not employed in this structure which concentrates on the robot manipulator control design. In addition, the learning technique of synchronous/simultaneous AC architecture* (3.23) *and* (3.26) *is different from data-driven online IRL in [28] and [41]. It is worth noting that a clear functionalized exploratory signal, as well as clear initial conditions of the system, is described in this work instead of random variables, which clarifies the learning process and contributes to the comparison of different approaches. These will be described in Section 3.3.1. It is important to approach robot manipulator dynamics as an affine system* (3.11) *in order to facilitate the ARL algorithm for the system in the tracking control problem.*

3.2.2 Time-varying RISE-based optimal control

The last step is to complete the control design in (3.7) by integrating the estimation of $\varepsilon = f + d$ based on the time-varying RISE framework in [37]. The proposed time-varying RISE structure is presented as in Section 2.2.1.2

$$\varepsilon(t) = (K_s(\cdot) + 1)s(t) - (K_s(t_0) + 1)s(0) + \rho(t) \tag{3.27}$$

$$\frac{d}{dt}\rho = (k_{s0} + 1)\alpha(\cdot)s(t) + \beta\,\mathrm{sgn}(s(t)) \tag{3.28}$$

In summary, the control input is described as

$$\tau = \varepsilon - u + n \tag{3.29}$$

Remark 3.3. *Compared with the proposed controller in [37], this work considers the adaptive optimal control problem of nonlinear systems and makes use of RISE to learn the disturbances/uncertainties. Additionally, due to the time-dependent property, it is not able to directly apply ARL strategy in the model* (3.9)*. Therefore, it is proposed to employ a transformation method to obtain the augmented autonomous system* (3.11) *which facilitates the ARL algorithm. The authors in [30] applied an online ARL-based technique for first-order continuous-time nonlinear autonomous system with no external disturbance. In this work, a disturbed/uncertain manipulator is introduced by second-order continuous-time nonlinear systems* (3.1) *and the sliding variable helps to achieve the reduced-order system model.*

Remark 3.4. *Other important improvements of this work in comparison with [34] are the time-varying RISE utilization and clear presentation of initial conditions and exploratory signals. This thesis improves the standard RISE framework by implementing time-varying nonlinear functions, which generalizes the control problems. Including the above-mentioned time-varying control gains in the classic equation of a RISE controller may boost the controller's global tracking efficiency and robustness to system uncertainties and disturbances. It is also significant that the nonlinear functions' structure is easy enough to incorporate in real-time experiments. Moreover, the random values in [34] are replaced by clear ones (see Section 3.3.1), which facilitates the comparison with the work in [34].*

3.3 Simulation results

3.3.1 Simulation setup

This section describes the evaluation of the performance of the proposed controllers through simulation tests. Both the original RISE and the proposed time-varying RISE methods with ARL were implemented on the two-link robot manipulator. A comparison between the two employed controllers is studied in the next sessions.

Consider a 2-DOF planar robot manipulator system, which is modeled by Euler-Lagrange formula (3.1). With $n = 2$, the described matrices in (3.1) can be given as

$$\begin{aligned} M(\eta) &= \begin{bmatrix} \zeta_1 + 2\zeta_2 \cos\eta_2 & \zeta_3 + \zeta_2 \cos\eta_2 \\ \zeta_3 + \zeta_2 \cos\eta_2 & \zeta_3 \end{bmatrix} \\ G(\eta) &= \begin{bmatrix} \zeta_4 \cos\eta_1 + \zeta_5 \cos(\eta_1 + \eta_2) \\ \zeta_5 \cos(\eta_1 + \eta_2) \end{bmatrix} \\ C(\eta, \dot{\eta}) &= \begin{bmatrix} -\zeta_2 \sin\eta_2 \dot{\eta}_2 & -\zeta_2 \sin\eta_2 (\dot{\eta}_1 + \dot{\eta}_2) \\ \zeta_2 \sin\eta_2 \dot{\eta}_1 & 0 \end{bmatrix} \end{aligned} \tag{3.30}$$

where ζ_i, $i = 1...5$ are system parameters depending on gravitational acceleration and mechanical description. These constant parameters can be selected as

$$\zeta_1 = 5, \quad \zeta_2 = 1, \quad \zeta_3 = 1, \quad \zeta_4 = 1.2, \quad \zeta_5 = g\,. \tag{3.31}$$

The simulation mission is to verify the improved performance of the proposed tracking controllers and to give clear comparison between the two approaches. The desired trajectory is set as $\eta_{ref} = \begin{bmatrix} 3sin(t) & 3cos(t) \end{bmatrix}^T$ with the vector of disturbances is introduced as $d(t) = \begin{bmatrix} 50sin(t) & 50cos(t) \end{bmatrix}^T$.

The optimal control problem is set by constructing the general performance index (3.14) with the positive-definite symmetric matrices in as

$$Q = \begin{bmatrix} 40 & 2 & -4 & 4 \\ 2 & 40 & 4 & -6 \\ -4 & 4 & 4 & 0 \\ 4 & -6 & 0 & 4 \end{bmatrix}, \quad R = \begin{bmatrix} 0.25 & 0 \\ 0 & 0.25 \end{bmatrix} \tag{3.32}$$

In sliding variable $s(t) = \dot{e}_1 + \alpha_1 e_1$, the control parameter $\alpha_1 \in \mathbb{R}^{n \times n}$ are selected to be a constant positive definite matrix:

$$\alpha_1 = \begin{bmatrix} 15.6 & 10.6 \\ 10.6 & 10.6 \end{bmatrix} \tag{3.33}$$

The feedback gains in the standard RISE law (2.26) are designed as

$$k_s = \begin{bmatrix} 210 & 0 \\ 0 & 210 \end{bmatrix}, \quad \alpha = \begin{bmatrix} 9.3 & 0 \\ 0 & 149 \end{bmatrix}, \quad \beta = 5\,. \tag{3.34}$$

and the time-varying RISE parameters as in (2.29), (2.30), and (2.31)

$$\begin{aligned} &k_{s0} = \begin{bmatrix} 200 & 0 \\ 0 & 200 \end{bmatrix}, \quad \gamma_1 = 0.96, \quad \delta_1 = 0.05, \\ &\alpha_0 = \begin{bmatrix} 5 & 0 \\ 0 & 80 \end{bmatrix}, \quad \gamma_2 = 1.6, \quad \delta_2 = 2.81, \\ &\beta = 5\,. \end{aligned} \tag{3.35}$$

The learning parameters in actor-critic architecture are chosen guaranteeing (3.23)–(3.26) as

$$k_c = 800, \quad \upsilon = 1, \quad k_{a1} = 0.02, \quad k_{a2} = 2\,. \tag{3.36}$$

On the other hand, according to [34], the value function V in (3.18) can be solved precisely as

$$V = 2x_1^2 - 4x_1x_2 + 3x_2^2 + 2.5x_3^2 + x_3^2\cos(\eta_2) + x_3x_4 + x_3x_4\cos(\eta_2) + 0.5x_4^2 \tag{3.37}$$

The choice of $\psi(X)$ in (3.18) can be arbitrary. However, for the comparison between approximate result from learning process and the exact result in (3.37), $\psi(X)$ should be chosen as

$$\psi(X) = \left[x_1^2, x_1x_2, x_2^2, x_3^2, x_3^2\cos(\eta_2), x_3x_4, x_3x_4\cos(\eta_2), x_4^2\right]^T \tag{3.38}$$

Considering (3.37), the exact value of $\hat{W}_c$ in (3.20) and $\hat{W}_a$ in (3.21) are

$$W = \begin{bmatrix} 2 & -4 & 3 & 2.5 & 1 & 1 & 1 & 0.5 \end{bmatrix}^T \tag{3.39}$$

In the simulation, the initial covariance matrix is selected as

$$\lambda(0) = \text{diag}\begin{pmatrix} 100 & 300 & 300 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \tag{3.40}$$

All the NN weights $\hat{W}_c,\ \hat{W}_a$ are initialized as

$$\begin{aligned} \hat{W}_c(0) &= \begin{bmatrix}0.6 & 0.1 & 0.7 & 0.5 & 0.5 & 0.5 & 0.7 & 0.6\end{bmatrix}^T \\ \hat{W}_a(0) &= \begin{bmatrix}0.6 & 0.2 & 0.2 & 0.9 & 1 & 1 & 0.4 & 0.2\end{bmatrix}^T \end{aligned} \tag{3.41}$$

and the states and their first-time derivative are initialized as

$$\begin{aligned} q(0) &= \begin{bmatrix}0.5 & 0\end{bmatrix}^T \\ \dot{q}(0) &= \begin{bmatrix}0.9 & 0.8\end{bmatrix}^T \end{aligned} \tag{3.42}$$

To ensure PE qualitatively, an exploratory signal consisting of sinusoids of varying frequencies is added to the control for the first 25 seconds after 35 seconds of simulation time.

$$\begin{aligned} n(t) &= \begin{bmatrix}n_1(t) & n_2(t)\end{bmatrix}^T \\ n_1(t) &= 75\left(\sin(-29t)^2\cos(28t) + \sin(-19t)^2\cos(22t) + \sin(20t)\cos(16t)\right), \\ n_2(t) &= 75\left(\sin(29t)^2\cos(27t) + \sin(23t)^2\cos(19t) + \sin(16t)\cos(15t)\right). \end{aligned} \tag{3.43}$$

In order to quantify the performance of the two control algorithm, it is important to define a certain performance index. Main objective is to improve tracking accuracy of the robot manipulator using the proposed controller. Hence, the Root-Mean-Square Error (RMSE) criterion is an accuracy evaluation tool mainly used to evaluate differences between the desired trajectory and the actual one.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(e_{1,1}^2(i) + e_{1,2}^2(i)\right)} \tag{3.44}$$

where $e_{1,1}$, $e_{1,2}$ refers to the joints tracking errors and N is the number of the collected samples through the whole process.

To estimate the energy consumption for each controller, the input-torques-based criterion is defined as

$$E_T = \sum_{i=1}^{2} \sum_{j=1}^{N} |\tau_i(j)| \tag{3.45}$$

where the control effort E_T is the total summation of the absolute value of the input signal produced by the two actuators.

To determine the convergence error of the training process, we calculate the differences between trained weights and precise weights.

$$CE = \left| \hat{W} - W \right| \tag{3.46}$$

The next session will quantitatively and visually demonstrate the simulation results and comparison between the two methods.

3.3.2 Result analysis

Table 3.1: Control performance evaluation for both controllers

	Original Optimal RISE		Optimal Varying RISE		Comments
Weights	1.9550	2.0000	1.9687	2.0000	41.63% better
	-4.0293	-4.0000	-4.0166	-4.0000	
	2.9487	3.0000	3.0277	3.0000	
	2.4950	2.5000	2.4987	2.5000	
	0.9946	1.0000	0.9996	1.0000	
	0.9915	1.0000	0.9998	1.0000	
	0.9828	1.0000	0.9996	1.0000	
	0.4999	0.5000	0.4991	0.5000	
CE	0.0771		0.0450		
RMSE	0.2073		0.2072		Similar
E_T	2.4767e+06		2.4637e+06		0.52% better

Table 3.1 notes some explicit information to compare the original RISE-based ARL (left sub-figures) and the proposed time-varying RISE methods with ARL (right sub-figures) were implemented on the two-link robot manipulator. The learned weights and ideal weights of each controller are shown together for convenient comparison. Some criteria indexes are calculated as defined in the previous section.

Regarding tracking errors RMSE, both static and varying RISE-based controllers present almost the same performance (0.2073 and 0.2072 respectively). Following the reference trajectory shown in Figure 3.1, the joint tracking errors for both controllers are calculated and depicted in Figure 3.2. The system rapidly follows the desired trajectories without overshoot, which demonstrates excellent tracking performance.

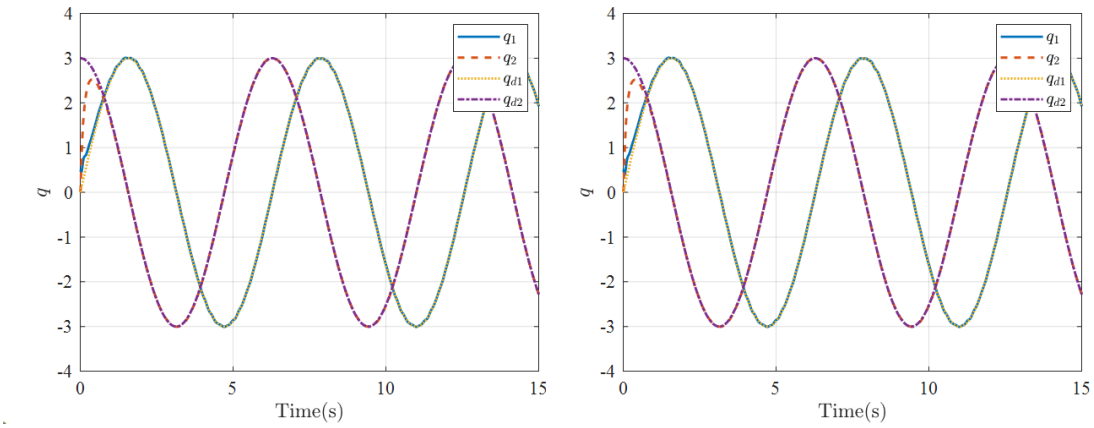


Figure 3.1: Tracking trajectories of the two controllers

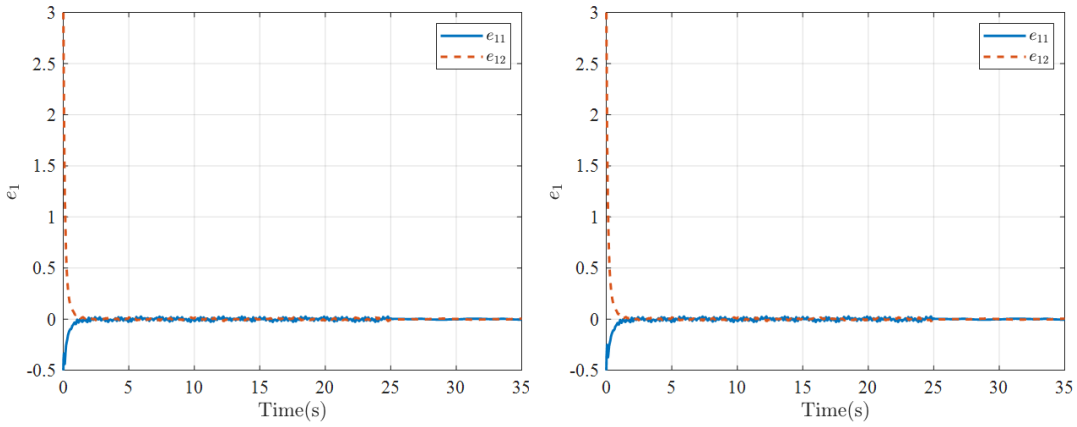


Figure 3.2: Tracking errors of the two controllers

It is worth noting that there are some minor variations in tracking errors of both controllers in the first 25 seconds because sinusoidal probing noises are injected directly to control input during the exploration stage.

An important note is an improvement in terms of energy consumption which is a 0.52% reduction with the novel method.

Because of the extended nonlinear feedback gains and their different behavior, the proposed time-varying RISE-based ARL control improves the original RISE control in terms of precision and efficiency.

Figure 3.3 shows that both methods work similarly when it comes to estimating uncertainties and disturbances. Because of the initial exploration process, it is difficult to include complicated probing signals in the estimation of the RISE algorithm. Following that, both methods' calculation of uncertainties and disturbances is brilliant.

In general, for both methods, the weight convergences in approximating optimal value function using NN are all excellent (see Table 3.1). The weights within the varying RISE ARL-based approach converges precisely to the solution in (3.39) while the standard method shows 41.63% less precise but acceptable final weight values. The convergence processes of actor and critic weights in training NNs are shown from Figure 3.4 and Figure 3.5. It is clear that the proposed time-varying RISE-based control method brings in faster and smoother convergences, compared

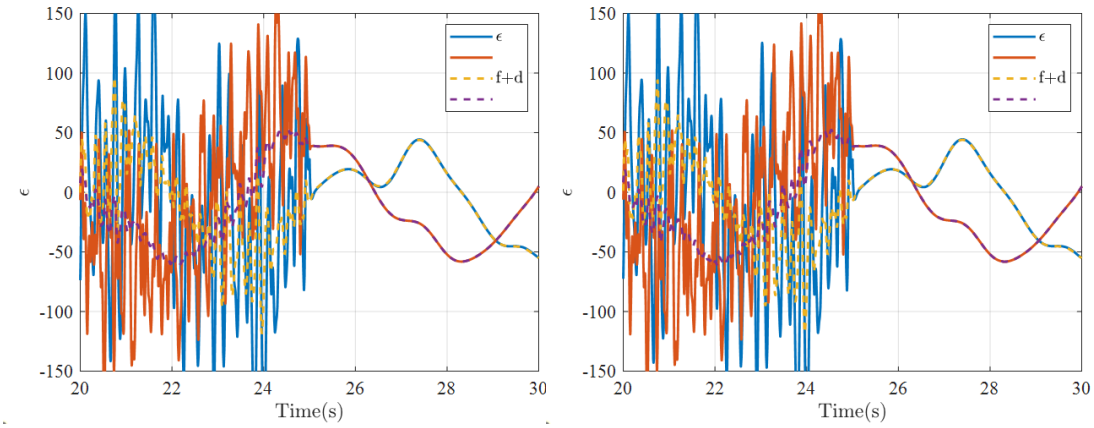


Figure 3.3: Estimation of the two controllers

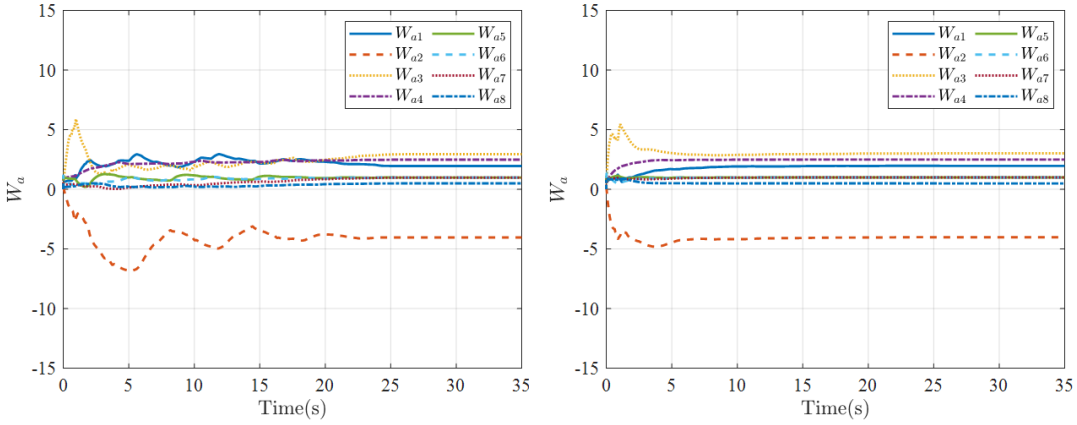


Figure 3.4: Actor weights of the two controllers

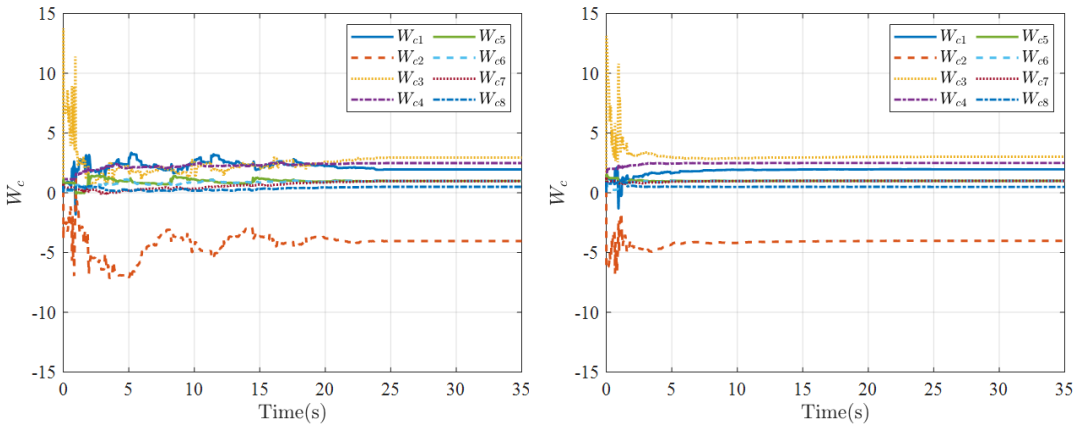


Figure 3.5: Critic weights of the two controllers

to the oscillatory ones presented by the standard RISE method.

3.4 Summary

This study addresses a robust optimal control problem for a class of uncertain nonlinear systems with unknown disturbances. In this work, after defining the sliding variable, an online ARL is presented to achieve optimality for the autonomous system. AC NNs are considered to approximate the HJB equation. Based on the RISE method, uncertain/disturbed components of the systems are estimated, which guarantees the trajectory tracking objective. Moreover, this work proposes a new structure where the time-varying RISE is combined with the ARL method to obtain closed-loop performance improvements. MATLAB simulation results on a 2-DOF robot arm demonstrate the improved performance of the time-varying RISE-based RL scheme in comparison with the original RISE-based RL controller.

Chapter 4

Disturbance Observer-Based Reinforcement Learning Control of Nonlinear Systems

This chapter presents a self-learning control method for nonlinear systems with unknown disturbances and uncertainties. Firstly, kinematic and feed-forward structures are employed to achieve an autonomous affine system from the original surface vessel model. The optimality of the transformed system is guaranteed by the ARL technique. Additionally, a nonlinear disturbance observer is implemented in this study to estimate the unknown disturbances and uncertainties of the system. Adaptive optimal control combining with disturbance compensation ability improves the performance of the closed-loop system.

4.1 Problem formulation

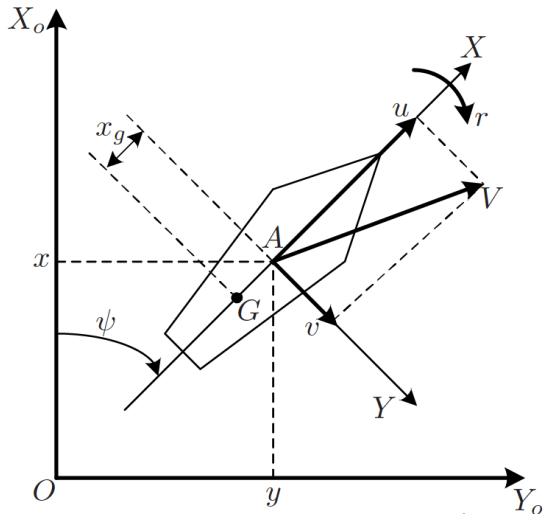


Figure 4.1: Coordinate frames of an SV

From Figure 4.1, $\eta = [x, y, \psi]^T$ denotes the 3-DOF position (x, y) and heading angle (ψ) of the SV in the earth-fixed inertial frame, and $\nu = [u, v, r]^T$ denotes the corresponding linear velocities (u, v) (surge and sway velocities), and yaw angular rate (r), in the body-fixed frame. The general model of an SV sailing in a planar

space neglecting the motions in heave, pitch and roll can be described as follows [42]

$$\begin{cases} \dot{\eta} = R(\eta)\nu(t) \\ M\dot{\nu} = \tau + \Delta(\eta, \nu) - f(\eta, \nu) \end{cases} \tag{4.1}$$

with dynamics $f(\eta, \nu)$ is modeled by

$$f(\eta, \nu) = C(\nu)\nu + D(\nu)\nu + g(\eta, \nu) \tag{4.2}$$

where $\tau \in \mathbb{R}^3$ is control input. The other input $\Delta(\eta, \nu)$ implies model uncertainties and disturbances, which is slowly time varying. The term $g(\eta, \nu)$ denote the restoring forces and moments due to gravitation/buoyancy. The authors in [42] compute and explain in detail the matrices M, $C(\nu)$, and $D(\nu)$.

A surface vessel with three degrees of freedom can be considered without the restoring forces and moments due to gravitation/buoyancy, i.e., $g(\eta, \nu) = 0$. However, noise from the environment can be affected to tilt the ship, then force and thrust will appear to bring the ship back into position balance. Therefore, there is no loss of generality while formula (4.1) remains with component $g(\eta, \nu)$.

The term $R(\eta)$ is a rotation matrix given by

$$R(\eta) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.3}$$

with the following property:

$$\|R(\eta)\| = 1 \quad \text{and} \quad R^T(\eta)R(\eta) = I \tag{4.4}$$

Assumption 4.1. *Assuming that η_d is the bounded desired trajectory, and there exists a Lipschitz continuous trajectory planning function $h_d(\cdot)$ such that*

$$\dot{\eta}_d = h_d(\eta_d) \tag{4.5}$$

The objective of this work is to propose an intelligent control scheme for the SV models (4.1) with unknown combined disturbances including uncertainties and external disturbances such that the reference positions and attitude can be tracked accurately and optimally.

4.2 Kinematic and feed-forward control structure

In this section, the outer loop including kinematic and feed-forward control structure is introduced, which contributes to the transformation from the original SV system to an affine one with autonomous property, facilitating ARL algorithm and DO design.

To quantify the tracking objective, one defines a position tracking error as

$$e_\eta = \eta - \eta_d \tag{4.6}$$

In a SV model (4.1), the kinematic subsystem is known as $\dot{\eta} = R(\eta)\nu$. From (4.6), the kinematic error dynamics is

$$\dot{e}_\eta = R(\eta)\nu - \dot{\eta}_d \tag{4.7}$$

Therefore, design the kinematic control law ν_d to guarantee asymptotic stability of (4.7) as follows

$$\nu_d = R^{-1}(\eta)(\dot{\eta}_d - \beta_\eta e_\eta) \tag{4.8}$$

where β_η is a positive definite matrix. Then, define the body-fixed velocity error:

$$e_\nu = \nu - \nu_d \tag{4.9}$$

Then, the dynamics of e_ν can be given as:

$$\dot{e}_\nu = -M^{-1} f(\eta, \nu) - \dot{\nu}_d + M^{-1}\tau + M^{-1}\Delta \tag{4.10}$$

It is important to obtain the autonomous systems in order to utilize ARL structure. Therefore, a feed-forward term τ_{ff} for stationary operation is added into the control input

$$\tau = u + \tau_{ff} \tag{4.11}$$

$$\tau_{ff} = M\dot{\nu}_d + f(\eta_d, \nu_d) \tag{4.12}$$

The robust optimal control law u will be designed later employing ARL algorithms and disturbance observer.

Considering systems (4.7) and (4.10) under the control law (4.11) and (4.12) without $g(\eta, \nu)$ gives a dynamic equation:

$$\dot{X} = \begin{bmatrix} M^{-1} f\big(\nu_d(e_\eta, n_d)\big) - M^{-1} f\big(e_\nu + \nu_d(e_\eta, n_d)\big) \\ R(e_\eta + \eta_d)e_\nu - \beta_\eta e_\eta \\ \dot{\eta}_d \end{bmatrix} + \begin{bmatrix} M^{-1} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} (u + \Delta) \tag{4.13}$$

where $X = [e_\nu^T \quad e_\eta^T \quad \eta_d^T]^T$ is an augmented state for the dynamic subsystem of SV. Combining with (4.5), the autonomous system can be obtained as

$$\dot{X} = \begin{bmatrix} M^{-1} f\big(\nu_d(e_\eta, n_d)\big) - M^{-1} f\big(e_\nu + \nu_d(e_\eta, n_d)\big) \\ R(e_\eta + \eta_d)e_\nu - \beta_\eta e_\eta \\ h_d(\eta_d) \end{bmatrix} + \begin{bmatrix} M^{-1} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} (u + \Delta) \tag{4.14}$$

The system (4.14) can be represented concisely as

$$\dot{X} = F(X) + G_u(X)u + G_d(X)\Delta \tag{4.15}$$

where

$$\begin{aligned} F(X) &= \begin{bmatrix} M^{-1} f\big(\nu_d(e_\eta, n_d)\big) - M^{-1} f\big(e_\nu + \nu_d(e_\eta, n_d)\big) \\ R(e_\eta + \eta_d)e_\nu - \beta_\eta e_\eta \\ h_d(\eta_d) \end{bmatrix} \\ G_u(X) &= G_d(X) = \begin{bmatrix} M^{-1} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \end{aligned} \tag{4.16}$$

and in general, $X \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $\Delta \in \mathbb{R}^l$, $F(X) \in \mathbb{R}^n$, $G_u(X) \in \mathbb{R}^{n\times m}$, $G_d(X) \in \mathbb{R}^{n\times l}$.

Remark 4.1. *The kinematic and feed-forward control structure contribute to the system transformation, which satisfies the situation of ARL algorithms. It is worth noting that* (4.15) *is a general MIMO affine nonlinear system with combined disturbances. In fact, most of the problems in automobile, robotics, aerospace, and other engineering systems can be described by the nonlinear control-affine equations. Therefore, the work in the next sections can be implemented for a class of nonlinear systems.*

For the nonlinear systems (4.15), the control input is expected to be designed as

$$u = d(X)\hat{\Delta} + u_r(X) \tag{4.17}$$

where $d(X)$ is the disturbance compensation gain to be designed, $\hat{\Delta}$ is the estimation of disturbances/uncertainties based on disturbance observer, and $u_r(X)$ is the optimal control policy with the absence of disturbances. The two parts of the control input (4.17) will be analyzed in the subsequent sections.

Remark 4.2. *The feed-forward design* (4.12) *compensated the effects of inertia, gravity, Coriolis, and friction which cannot be modified by the robust optimal design stage [43]. Additionally, the disturbance observer rejects the effects of unknown disturbances and uncertainties, which results in a nominal system. Therefore, it is reasonable that the cost function is only considered for the auxiliary control* $u_r(X)$ *.*

4.3 Adaptive reinforcement learning of nonlinear systems based on disturbance observer

In this section, an on-policy iterative optimal control method is applied to the system to obtain $u_r(X)$. The benefit of this on-policy iterative algorithm is not only that each iterative control policy makes the closed-loop system without disturbances asymptotically stable and UUB stable in the case the system is affected by disturbance Δ, but the control policy will also convergence to the optimal control policy of this system with the quadratic performance index $J \in \mathbb{R}$ to be minimized and constrained by (4.15).

4.3.1 On-policy actor-critic architecture-based algorithm

At first, let the system input be $u = u_r(X)$, the performance index function for system (4.15) without disturbances is generally defined as follows

$$J(X, u_r) = \int_0^\infty (X^T Q_T X + u_r^T R u_r) d\tau \tag{4.18}$$

where

$$Q_T = \begin{bmatrix} Q & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{4.19}$$

along with Q and R are positive definite symmetric matrices. Given the performance index $J(X, u_r)$, the control objective is to find the auxiliary control input $u_r(X)$ that minimizes (4.18) with system (4.15), which is known as as $u_r(X)$ optimal control.

Remark 4.3. *This is the same problem encountered in the previous analysis. The details of online policy iteration algorithm and AC structure are presented in Section 2.1.1 and Section 3.2.1. Let the iterative control be obtained by* $u_r^k(X)$*,* V^k *be the unique positive-definite function satisfying the Bellman equation (BE) for nonlinear systems* (4.15) *with* $\Delta = 0$*. If the input control for system* (4.15) *with* $\Delta = 0$ *is* $u_r^k(X)$*, then the iterative control* $u_r^k(X)$*,* $k = 1, 2, ...$ *makes the closed-loop system asymptotically stable. Moreover, the optimal control* $u_r^*(X)$ *can be achieved by the iteration, that is* $u_r^k(X) = u_r^*(X)$ *as* $k \to \infty$ *[44].*

The on-policy actor-critic algorithm results in the neural network-based estimation of $V(X)$ and $u_r(X)$ as follows:

$$\hat{V}(X) = \hat{W}_c{}^T \Psi(X) \tag{4.20}$$

$$\hat{u}_r(X) = -\frac{1}{2} R^{-1} G^T(X) \left(\frac{\partial \Psi}{\partial X}\right)^T \hat{W}_a \tag{4.21}$$

The actor and critic networks are both tuned based on the minimization of the TD error which can be written as

$$\delta_{hjb} = \hat{W}_c^T \sigma(X, \hat{u}_r) + X^T Q_T X + \hat{u}_r^T R \hat{u}_r \tag{4.22}$$

4.3.2 Disturbance observer-based robust optimal control

In this section, the disturbance observer analyzed in Section 2.2.2 is implemented for the system (4.15). According to (2.35), we have

$$\begin{cases} \hat{\Delta} = y + P(X) \\ \dot{y} = -\frac{\partial P(X)}{\partial X}(G_u(X)u + G_d(X)y + G_d(X)P(X) + F(X)) \end{cases} \tag{4.23}$$

Based on Theorem 2.2, the disturbance observer can identify the disturbances. Furthermore, $\hat{u}_r(X)$ is the approximate optimal control policy $u_r^*(X)$. Then the system (4.15) with the control input (4.17) is

$$\begin{aligned} \dot{X} &= F(X) + G_u(d\hat{\Delta} + u_r) + G_d\Delta \\ &= F(X) + G_u\hat{u}_r + G_u d\hat{\Delta} + G_d(\tilde{\Delta} + \hat{\Delta}) \end{aligned} \tag{4.24}$$

Theorem 4.1. *Let the closed-loop system be as* (4.24)*, the approximate optimal control be as* (4.21)*. Let* $d(X) = -G_u^+ G_d$ *then the observation error* $\tilde{\Delta}$ *are asymptotically stable, the closed-loop system state* X *and the weight errors are UUB.*

Proof. For more details, please see [44]. □

4.4 Simulation results

4.4.1 Simulation setup

In this section, a simulation example is implemented to verify the effectiveness of the developed procedures on a surface vessel system with a scale factor of 1 : 75. The mass of the model ship is 21 kg, its length and width are 1.2 m and 0.3 m, respectively. The parameters of each surface vessel are chosen as with the following

inertia, Coriolis, damping matrices:

$$\begin{aligned}
M &= \begin{bmatrix} 20 & 0 & 0 \\ 0 & 19 & 0.72 \\ 0 & 0.72 & 2.7 \end{bmatrix} \\
C(\nu) &= \begin{bmatrix} 0 & 0 & -19v - 0.72r \\ 0 & 0 & 20u \\ 19v + 0.72r & -20u & 0 \end{bmatrix} \\
D(v) &= \begin{bmatrix} d_{11} & 0 & 0 \\ 0 & d_{22} & d_{23} \\ 0 & d_{32} & d_{33} \end{bmatrix} \\
d_{11} &= 0.72 + 1.3\,|u| + 5.8u^2 \\
d_{22} &= 0.86 + 36\,|v| + 3\,|r| \\
d_{23} &= -0.1 - 2\,|v| + 2\,|r| \\
d_{32} &= -0.1 - 5\,|v| + 3\,|r| \\
d_{33} &= 6 + 4\,|v| + 4\,|r|
\end{aligned} \tag{4.25}$$

The control objective is to drive the surface vessel to track the desired time-varying trajectory generated as

$$\eta_d(t) = \begin{bmatrix} 12\sin(0.2t) & -12\cos(0.2t) & 0.2t \end{bmatrix}^T \tag{4.26}$$

while guaranteeing the optimal performance (4.18) in the presence of unknown disturbances/uncertainties deployed as

$$\Delta = 30 \begin{bmatrix} 4\sin(t+1.2) + 0.5\dot{x} \\ 0.7\sin(t+0.5) + 0.5\dot{y} \\ 0.25\cos t + 0.5\dot{\psi} \end{bmatrix} \tag{4.27}$$

Let the initial conditions be

$$\begin{aligned}
\eta(0) &= \begin{bmatrix} 1 & -9 & 0.5 \end{bmatrix}^T \\
\nu(0) &= \begin{bmatrix} 3 & 3 & 3 \end{bmatrix}^T \\
\hat{W}_c(0) &= 0.03 \times \mathbf{1}_{12\times 1} \\
\hat{W}_a(0) &= 0.03 \times \mathbf{1}_{12\times 1}
\end{aligned} \tag{4.28}$$

The design parameters consist of kinematic control law, feed-forward, ARL algorithm being chosen as

$$\begin{array}{llll} \beta_\eta = 0.5 & k_c = 1 & k_{a1} = 1 & k_{a2} = 2 \\ \upsilon = 5 & \varphi_0 = 20 & \varphi_1 = 12 & Q = I_3 \quad R = I_3 \end{array} \tag{4.29}$$

For the training of actor-critic architecture to achieve ARL-based optimal control, the dual NNs are designed with 12 nodes. The smooth basis/activation function vector $\Psi(X) \in \mathbb{R}^{12}$ is chosen as

$$\begin{aligned}
\Psi(X) = [&X_1^2, X_1X_2, X_1X_3, X_2^2, X_2X_3, X_3^2, \\
&X_1^2X_7^2,\ X_2^2X_8^2,\ X_3^2X_9^2,\ X_1^2X_4^2,\ X_2^2X_5^2,\ X_3^2X_6^2]^T
\end{aligned} \tag{4.30}$$

For constructing the disturbance observer, let

$$P(X) = 390 \begin{bmatrix} 100X_1 \\ 57X_2 + 2.16X_3 \\ 1.44X_2 + 5.4X_3 \end{bmatrix} \tag{4.31}$$

then $\partial P/\partial X = 390\begin{bmatrix}100 & 0 & 0; 0 & 57 & 2.16; 0 & 1.44 & 5.4\end{bmatrix}^T$,
and $H = (\partial P/\partial X)G_d = 390\begin{bmatrix}5 & 0 & 0; 0 & 3 & 0; 0 & 0 & 2\end{bmatrix}^T$ is positive.

4.4.2 Result analysis

This part presents remarkable simulation outcomes of a robust optimal control algorithm for a surface vessel via online policy iteration and disturbance observer. Results of the ARL method for the same vessel without disturbance observer are introduced in the left column of each figure for a reasonable comparison.

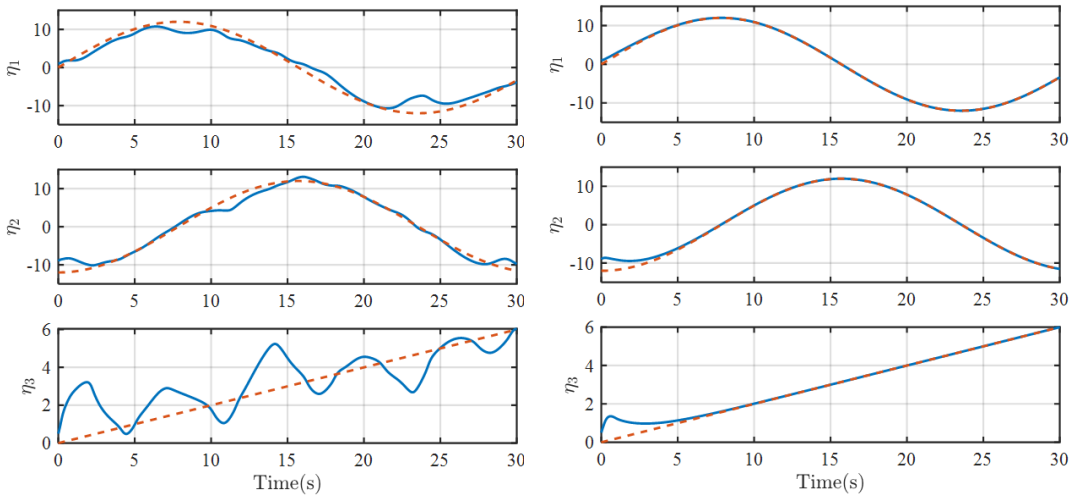


Figure 4.2: Tracking trajectories of the two approaches

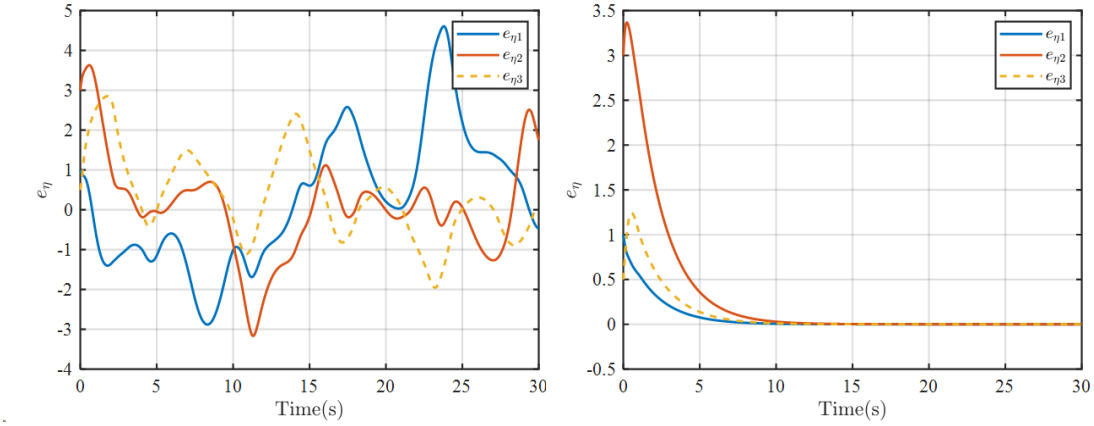


Figure 4.3: Tracking trajectories of the two approaches

The trajectory tracking performance of closed-loop control systems is shown in Figure 4.2 and Figure 4.3. In addition, the reference and actual trajectories in the planar space are shown in Figure 4.4. In the presence of disturbances, the performance of the initial ARL controller (without disturbance observer) is significantly degraded. With the proposed method controlled trajectory can exactly track the desired one within a short time despite suffering from unmodeled dynamics and unknown disturbances. While, When it comes to tracking errors RMSE, the disturbance observer contributes to a significant reduction of 66% (from 2.38 to 0.81).

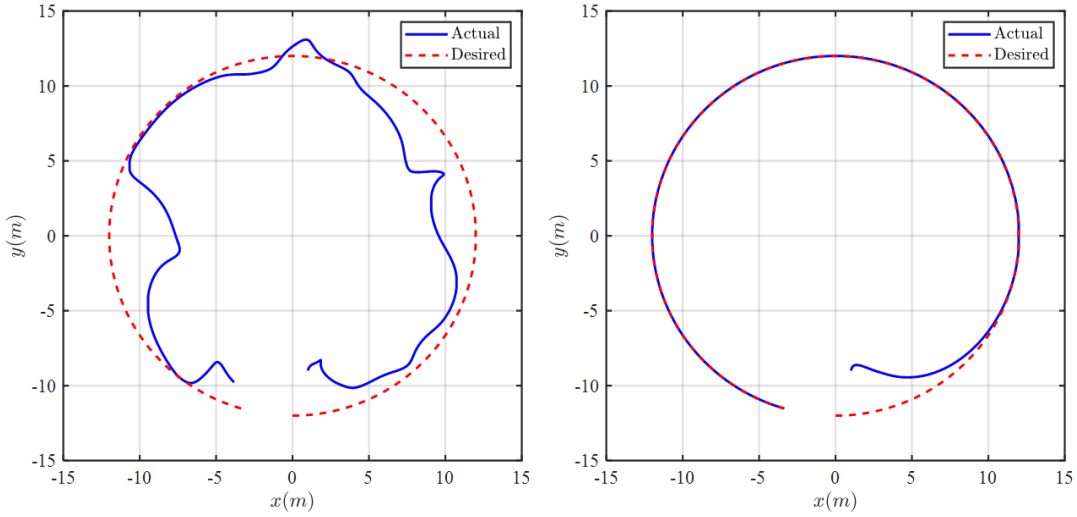


Figure 4.4: The trajectories of surface vessel in the planar space

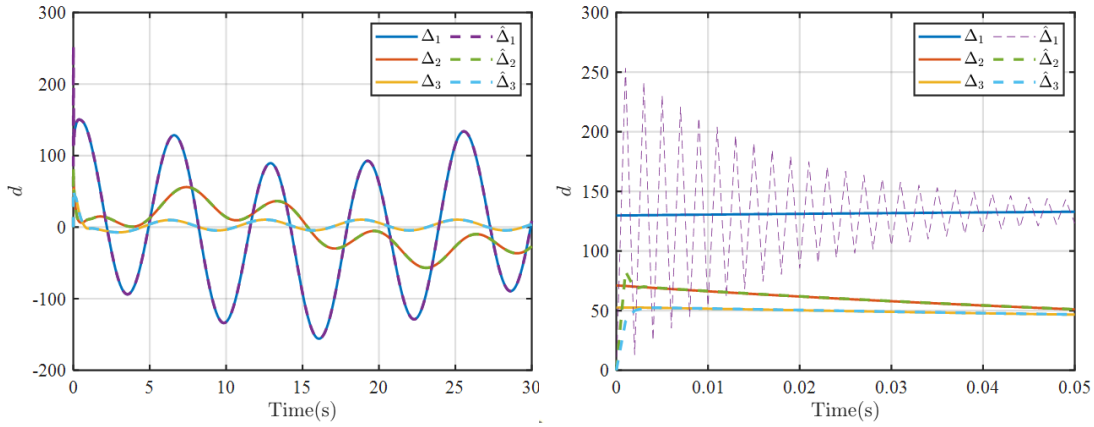


Figure 4.5: The performance of disturbance observer

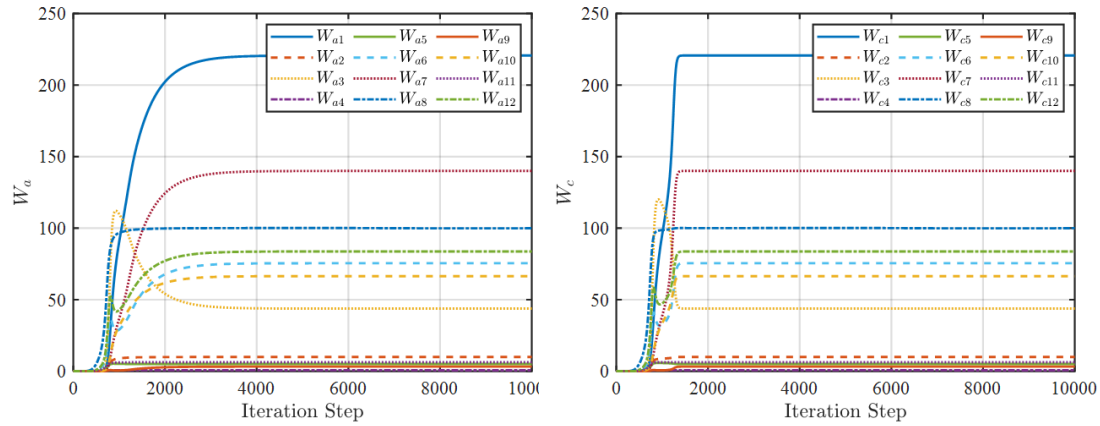


Figure 4.6: The convergence of NN weights of the proposed control system

From Figure 4.5, it is obvious that the observer effectively estimates the disturbances throughout the whole process. It also shows the performance of the disturbance observer during the transient procedure. All the disturbances are accurately observed just after about 0.5 seconds which is 10 times faster than the tracking performance.

The convergences of actor-critic weights of the improved method are shown in Figure 4.6.

4.5 Summary

Considering nonlinear systems with unknown disturbances, this work proposes a DOB RL control approach. On-policy AC architecture is used to address the optimal control problem for a transformed autonomous system without disturbances and it aims to stabilize the nonlinear plant and get the optimal value function. Additionally, a nonlinear disturbance observer is implemented in this study to attenuate the unknown disturbances and uncertainties of the system, which improves the performance of the closed-loop system. The compensation control, together with the RL core, produces the robust optimal control input. To verify the advantages of the proposed control structure, a comparison with the original RL-based method is made, implementing a surface vessel system simulation.

Chapter 5

Conclusion

5.1 Conclusion

This thesis concentrates on reproducing the accomplishment of RL techniques in machine learning to control problems of continuous-time nonlinear systems. To achieve optimality, ADP/RL-based designs are implemented to obtain an approximate solution to the HJB equation. By making use of the PI method together with NN-based function approximators, the optimal policy and value function of continuous nonlinear systems are learned online in real time. Moreover, dealing with system uncertainties and unknown disturbances to achieve robust optimal performance is an interesting concern of this work. While in Chapter 3, a novel time-varying RISE method is used to improve the RL-based control structure for uncertain systems with disturbances, DOBC is used in Chapter 4 to develop robust optimal controllers. Both proposed methods are able to improve the tracking performance of the closed-loop system as desired at the beginning of the thesis. The practical promise of online RL techniques, where the controllers can learn the best policy by interacting with the environment, is illustrated through MATLAB simulations. This work realizes higher degrees of autonomy in addressing problems from a wide range of areas, including artificial intelligence, cybernetics, operations research, economics, and so on.

One drawback of the methods, still, is the requirement of the knowledge of drift matrix and input gain matrix. Besides, due to the time limit, some of closed-loop stability and optimality has not been explicitly proven yet.

5.2 Future work

This thesis contributes to confirming that RL methods can be productively applied to feedback control. The developed techniques are relatively wide-ranging and implementable, however, research in this field is still at a growing stage, and there are a number of interesting unresolved challenges.

This result spurs further research into partially/completely model-free RL approaches such as integral RL and off-policy with explicit optimality and stability demonstration. Furthermore, differential games and multi-agent systems are intriguing areas in which RL research should be deeply investigated.

References

[1] J. Iqbal, M. Ullah, S. Khan, K. Baizid, and S. Cukovic, "Nonlinear control systems – a brief overview of historical and recent advances," *Nonlinear Engineering*, vol. 6, 08 2017.

[2] K. Sachan and R. Padhi, "Output-constrained robust adaptive control for uncertain nonlinear mimo systems with unknown control directions," *IEEE Control Systems Letters*, vol. 3, no. 4, pp. 823–828, 2019.

[3] P. A. Ioannou and J. Sun, *Robust Adaptive Control*. USA: Prentice-Hall, Inc., 1995.

[4] G. Bartolini, L. Fridman, A. Pisano, and E. Usai, *Modern Sliding Mode Control Theory: New Perspectives and Applications*, vol. 375. Springer, 01 2008.

[5] V. Utkin, "Discussion aspects of high-order sliding mode control," *IEEE Transactions on Automatic Control*, vol. 61, no. 3, pp. 829–833, 2016.

[6] J. Davila, "Exact tracking using backstepping control design and high-order sliding modes," *IEEE Transactions on Automatic Control*, vol. 58, no. 8, pp. 2077–2081, 2013.

[7] F. Mazenc and P.-A. Bliman, "Backstepping design for time-delay nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 51, no. 1, pp. 149–154, 2006.

[8] N. Vu, N. Tran, and N. Nguyen, "Adaptive neuro-fuzzy inference system based path planning for excavator arm," *Journal of Robotics*, vol. 2018, pp. 1–7, 12 2018.

[9] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-dynamic programming*. Athena Scientific, 1996.

[10] Z. Wu and T. Zhang, "Adaptive finite-time tracking control for parameterized nonlinear systems with full state constraints," *International Journal of Adaptive Control and Signal Processing*, 06 2021.

[11] D. Nam, P. Loc, N. Huong, and T. Tan, "A finite-time sliding mode controller design for flexible joint manipulator systems based on disturbance observer," *International Journal of Mechanical Engineering and Robotics Research*, pp. 619–625, 01 2019.

[12] X. Luo, X. Zhang, and S. Wang, "On-line squaring of non-square hard constraints of input variable by coordinate alternating in model predictive control," in *Proceedings of the 10th World Congress on Intelligent Control and Automation*, pp. 2529–2536, 2012.

[13] K. G. Vamvoudakis and F. Lewis, "Online solution of nonlinear two-player zero-sum games using synchronous policy iteration," in *49th IEEE Conference on Decision and Control (CDC)*, pp. 3040–3047, 2010.

[14] Y. Guo, J. Zhou, and Y. Liu, "Distributed rise control for spacecraft formation reconfiguration with collision avoidance," *Journal of the Franklin Institute*, vol. 356, 05 2019.

[15] S. Han, "Non-transformed prescribing performance function and finite-time rise-based tracking control for euler-lagrange systems," *IEEE Access*, vol. 8, pp. 136872–136883, 2020.

[16] I. Ponce Monarrez, Y. Orlov, L. Aguilar, and J. Alvarez, "Robust tracking control of servo systems with backlash: Nonsmooth H_∞ control vs. linear H_∞ control," *Proceedings of the American Control Conference*, vol. 2015, pp. 2051–2056, 07 2015.

[17] Z. Chen, Y.-J. Pan, and J. Gu, "Disturbance observer based adaptive robust control of bilateral teleoperation systems under time delays," in *2013 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 2649–2654, 2013.

[18] E. Sariyildiz, R. Oboe, and K. Ohnishi, "Disturbance observer-based robust control and its applications: 35th anniversary overview," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 3, pp. 2042–2053, 2020.

[19] P. J. Werbos, "Building and understanding adaptive systems: A statistical/numerical approach to factory automation and brain research," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 17, no. 1, pp. 7–20, 1987.

[20] F. L. Lewis, D. Vrabie, and V. L. Syrmos, *Optimal control*. John Wiley & Sons, 2012.

[21] R. S. Sutton and A. G. Barto, *Reinforcement learning: an introduction*. The MIT Press, 2018.

[22] R. Kamalapurkar, P. Walters, J. Rosenfeld, and W. Dixon, *Reinforcement Learning for Optimal Feedback Control: A Lyapunov-Based Approach*. Springer Publishing Company, Incorporated, 1st ed., 2018.

[23] L. Baird, "Reinforcement learning in continuous time: advantage updating," in *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*, vol. 4, pp. 2448–2453, 1994.

[24] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3, pp. 279–292, 1992.

[25] F. L. Lewis and D. Vrabie, "Reinforcement learning and adaptive dynamic programming for feedback control," *IEEE Circuits and Systems Magazine*, vol. 9, no. 3, pp. 32–50, 2009.

[26] D. Wang, D. Liu, Q. Wei, D. Zhao, and N. Jin, "Optimal control of unknown nonaffine nonlinear discrete-time systems based on adaptive dynamic programming," *Automatica*, vol. 48, no. 8, pp. 1825–1832, 2012.

[27] K. G. Vamvoudakis and F. L. Lewis, "Online actor–critic algorithm to solve the continuous-time infinite horizon optimal control problem," *Automatica*, vol. 46, no. 5, pp. 878–888, 2010.

[28] J. Y. Lee, J. B. Park, and Y. H. Choi, "Integral reinforcement learning for continuous-time input-affine nonlinear systems with simultaneous invariant explorations," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 5, pp. 916–932, 2015.

[29] X. Yang, H. He, D. Liu, and Y. Zhu, "Adaptive dynamic programming for robust neural control of unknown continuous-time nonlinear systems," *IET Control Theory & Applications*, vol. 11, 05 2017.

[30] S. Bhasin, R. Kamalapurkar, M. Johnson, K. G. Vamvoudakis, F. L. Lewis, and W. E. Dixon, "A novel actor-critic-identifier architecture for approximate optimal control of uncertain nonlinear systems," *Automatica*, vol. 49, no. 1, p. 82–92, 2013.

[31] J. Na, Y. Lv, K. Zhang, and J. Zhao, "Adaptive identifier-critic-based optimal tracking control for nonlinear systems with experimental validation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–14, 2020.

[32] W. Bai, Q. Zhou, T. Li, and H. Li, "Adaptive reinforcement learning neural network control for uncertain nonlinear system with input saturation," *IEEE Transactions on Cybernetics*, vol. 50, no. 8, pp. 3433–3443, 2020.

[33] R. M. Kretchmar, P. Young, C. Anderson, D. Hittle, M. Anderson, and C. C. Delnero, "Robust reinforcement learning control with static and dynamic stability," *International Journal of Robust and Nonlinear Control*, vol. 11, pp. 1469–1500, 2001.

[34] P. N. Dao, P. T. Loc, and T. Q. Huy, "Sliding variable-based online adaptive reinforcement learning of uncertain/disturbed nonlinear mechanical systems," *Journal of Control, Automation and Electrical Systems*, vol. 32, pp. 281–290, 2021.

[35] Y. Jiang and Z.-P. Jiang, *Robust Adaptive Dynamic Programming*. Wiley-IEEE Press, 1st ed., 2017.

[36] B. Xian, D. Dawson, M. de Queiroz, and J. Chen, "A continuous asymptotic tracking control strategy for uncertain nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 49, no. 7, pp. 1206–1211, 2004.

[37] H. Saied, A. Chemori, M. Bouri, M. el Rafei, F. Clovis, and F. Pierrot, "A new time-varying feedback rise control for 2nd-order nonlinear mimo systems: Theory and experiments," *International Journal of Control*, 12 2019.

[38] W.-H. Chen, "Disturbance observer based control for nonlinear systems," *IEEE/ASME Transactions on Mechatronics*, vol. 9, no. 4, pp. 706–710, 2004.

[39] J. Yang, W.-H. Chen, and S. Li, "Non-linear disturbance observer-based robust control for systems with mismatched disturbances/uncertainties," *Control Theory & Applications, IET*, vol. 5, pp. 2053–2062, 12 2011.

[40] Y. Guo, B. Huang, A. Li, and C. Wang, "Integral sliding mode control for euler-lagrange systems with input saturation," *International Journal of Robust and Nonlinear Control*, vol. 29, 12 2018.

[41] K. G. Vamvoudakis, D. Vrabie, and F. L. Lewis, "Online adaptive learning of optimal control solutions using integral reinforcement learning," in *2011 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, pp. 250–257, 2011.

[42] N. Wang, S. Lv, M. J. Er, and W.-H. Chen, "Fast and accurate trajectory tracking control of an autonomous surface vehicle with unmodeled dynamics and disturbances," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 3, pp. 230–243, 2016.

[43] Y. Kim, F. Lewis, and D. Dawson, "Intelligent optimal control of robotic manipulators using neural networks," *Automatica*, vol. 36, pp. 1355–1364, 09 2000.

[44] R. Song and F. L. Lewis, "Robust optimal control for a class of nonlinear systems with unknown disturbances based on disturbance observer and policy iteration," *Neurocomputing*, vol. 390, pp. 185–195, 2020.