Technische Universität München
TUM School of Computation, Information and Technology

# Towards Safe Learning Control under Uncertainty with Guaranteed Performance

**Cong Li**

Vollständiger Abdruck der von der TUM School of Computation, Information and Technology der Technischen Universität München zur Erlangung des akademischen Grades eines

**Doktors der Ingenieurwissenschaften (Dr.-Ing.)**

genehmigten Dissertation.

Vorsitz: Prof. Dr.-Ing. Gerhard Rigoll

Prüfende/-r der Dissertation:

1. Prof. Dr.-Ing. Martin Buss

2. Prof. Dr.-Ing. Klaus Diepold

Die Dissertation wurde am 12.04.2022 bei der Technischen Universität München eingereicht und durch die TUM School of Computation, Information and Technology am 19.09.2022 angenommen.

*To my parents*

献给我亲爱的父母

李夕怀 孙凤梅

# Acknowledgments

This dissertation summarizes my research conducted over the last four years at the Chair of Automatic Control Engineering, LSR, at the Technical University of Munich, Germany. I want to thank all the people who have offered me assistance, encouragement, and support.

First and foremost, I would like to express my utmost gratitude to my supervisor Prof. Martin Buss for his insightful guidance throughout my PhD. Many thanks for offering the opportunity to work in such an excellent team, helping me find a promising research topic and keeping me on the right path. Thanks for the inspiring discussions and talks. Thanks to make all things possible.

Furthermore, I would like to express my gratitude to Dr. Fangzhou Liu for sharing his knowledge about writing papers and rebuttals, and offering me valuable suggestions concerning my academic career. Many thanks for your help, support, and encouragement. Also, I want to thank Dr. Marion Leibold for her constructive feedback and kind help with my research. Many thanks for passing her teaching experience to me.

I thank the whole team at LSR and ITR for the time we shared. Thanks to my colleagues Qingchen Liu, Yongchao Wang, Zengjie Zhang, Zhehua Zhou, Yingwei Du, Tong Liu, Tim Brüdigam, Volker Gabler, Stefan Friedrich, Ni Dang, and Yuhong Chen for making my time in TUM special. Special thanks to Michael Fink to help translate the abstract of my dissertation into German. Also, I want to thank Larissa Schmid for all her kind help during the last four years.

This dissertation is also supported by works from my students Ahmed Nesrin, Hoang Giang Dang, Yuan Meng, Shuo Wang. Thanks for your excellent works. May all of you have a bright future.

Many thanks to Prof.Yong Wang for offering me new perspectives about research and life during my hard time. Thank you for always having my back. Thanks to my friends Haizhen Zhu, Xiaofei Wang, Jinghan Yang, Bin Liu and Xiaolong Wang for helping me survive the difficult time. Thanks to my landlord Hongchang Yang for offering me a great place in Dachau. Thanks my parents Xihuai Li, Fengmei Sun and my sister Wenxiu Li for their endless love and support during the last thirty years. Thanks my girlfriend Yingying Li for her incredible patience and invaluable support. Thanks for helping me being calm to the difficult life.

Munich, April 2022 Cong Li

# Abstract

Autonomous systems either robot manipulators, unmanned aerial vehicles, or self-driving cars are desired to safely accomplish predetermined tasks with guaranteed or optimal performance despite uncertainties. Safety often interpreted as state and input constraints encoding restricted operation ranges and actuation limits should be fully considered for reliable practical applications. Uncertainties arising from sources such as unmodelled dynamics, model uncertainties, and external disturbances need to be appropriately addressed to empower autonomous systems with adaptation capabilities to changing environments. Performance objectives such as closed-loop stability, high tracking accuracy, and minimum energy consumption ought to be satisfied to meet requirements or preferences to complete given tasks. This dissertation presents our efforts applying set-theoretic and reinforcement learning approaches to formulate, analyze, and solve the aforementioned problem termed as safe learning control under uncertainty with guaranteed performance.

The learning-supported set-theoretic methods, specifically the barrier Lyapunov function and the control barrier function, are used in Part I to achieve the desirable robust safety with guaranteed performance for continuous time nonlinear control applications. We first learn uncertain dynamics via concurrent learning to improve the tracking performance safely and gradually using the barrier Lyapunov function based control strategy. In particular, we adopt concurrent learning to ensure that the learned parameters converge to actual values using both realtime and historical data. Besides that, we utilize the barrier Lyapunov function to integrate safety (represented by predetermined tracking error bounds) with stability. Thereby, our designed stable control strategy based on learned dynamics could achieve safe tracking, while reducing uncertainty during the operation process at the same time. However, concurrent learning is restricted to address parametric uncertainties and demands a knowledge of model structure. Regarding the limitations of concurrent learning, we next leverage time-delayed signals to construct incremental systems to facilitate model-free control. These incremental systems are equivalent representations of original controlled plants but without using explicit model knowledge. The utilized time-delayed signals reduce the influence of uncertainties on controlled plants to the effect of a provably bounded time delay estimation error in the afore-mentioned incremental systems. Through an input-to-state stable approach combining with barrier Lyapunov functions and backstepping, we thoroughly analyze the time delay estimation error during the recursive controller design process. This allows us to achieve provably safe control under uncertainty with high-accuracy tracking performance. The preceding approaches follow a common mapping, planning, and control decoupled approach to complete safe execution under uncertainties. Alternatively, we integrate perception with control levels to build safe learning systems resilient to unforeseen environments. This is achieved by the control-level quadratic optimization with the constraints, referred to as instantaneous local control barrier functions and goal-driven control Lyapunov functions, learned from perceptional signals. The integrated approach bypasses gaps among levels in the common map-plan-track decoupled paradigm to facilitate the theoretically guaranteed collision avoidance and convergence to destinations. The instantaneous local sensory data stimulates computationally-cheap safe control strategies with fast adaptation to diverse uncertain environments without building a map.

The reinforcement learning and the control theory are combined in Part II to achieve safe learning and optimization in the presence of uncertainties. The reinforcement learning based optimization framework is embedded with safety and robustness guarantees applying theoretical analysis tools rooted in the control field. We first propose an off-policy risk-sensitive reinforcement learning based control framework to jointly optimize task performance and constraint satisfaction in a disturbed environment. In particular, we design risk-sensitive state penalty terms to construct risk-aware value functions that penalize unsafe behaviours. The above risk-aware value function is approximated by the safety critic employing an off-policy weight update law. During the learning process, the associated approximate optimal control policy is able to satisfy both input and state constraints under disturbances. However, the model-free property of reinforcement learning is traded off for theoretical guarantees in the approach mentioned above. Specifically, prior model information is used to present provable safety and stability under uncertainty. Therefore, we subsequently develop a time-delayed data informed reinforcement learning method, termed as incremental adaptive dynamic programming, to solve the optimal control problem in a model-free way and guarantees rigorous stability. In particular, the time-delayed data informs the value function learning process about one model-free representation of the original controlled plant. Thereby, we could achieve model-free control and also have a mathematical form of dynamics to conduct rigorous theoretical analysis applying rich analysis tools from the control field. Our developed incremental adaptive dynamic programming approach serves as an efficient tool to learn the solutions to both the optimal feedback motion planning and the optimal tracking control problems.

# Zusammenfassung

Autonome Systeme, ob Robotermanipulatoren, unbemannte Luftfahrzeuge oder selbstfahrende Autos, sollen trotz aller Unwägbarkeiten vorgegebene Aufgaben mit garantierter oder optimaler Leistung sicher erledigen. Die Sicherheit wird oft als Zustands- und Eingabebeschränkung, resultierend aus einen eingeschränkten Betriebsbereiche und Stellgrößenbeschränkungen, interpretiert. Diese sollten für zuverlässige praktische Anwendungen umfassend berücksichtigt werden; Unsicherheiten, die sich aus Quellen wie nicht modellierter Dynamik, Modellunsicherheiten und externen Störungen ergeben, müssen angemessen berücksichtigt werden, um autonome Systeme mit Anpassungsfähigkeiten an eine sich verändernde Umgebung auszustatten; Vorgaben wie die Stabilität des geschlossenen Regelkreises, eine hohe Verfolgungsgenauigkeit und ein minimaler Energieverbrauch sollten erfüllt werden, um Voraussetzungen oder Präferenzen bei der Erfüllung bestimmter Aufgaben zu erfüllen. In dieser Dissertation werden unsere Bemühungen vorgestellt, mit Hilfe von mengentheoretischen Methoden und Reinforcement-Learning-Ansätzen das oben genannte Problem zu formulieren, zu analysieren und zu lösen. Das Problem wird als sichere Lernregelung unter Unsicherheit mit garantierter Leistung bezeichnet.

Die lernunterstützten mengentheoretischen Methoden, insbesondere Barriere-Lyapunov-Funktionen und Kontroll-Barriere-Funktionen, werden in Teil I verwendet, um die wünschenswerte robuste Sicherheit für Anwedungen mit garantierter Leistung für zeitkontinuierliche nichtlineare Regelung zu realisieren. Zunächst lernen wir die unsichere Dynamik, um die Verfolgungsleistung schrittweise durch gleichzeitiges Lernen und die auf der Barriere-Lyapunov-Funktion basierende Regelungsstrategie sicher zu verbessern. Insbesondere setzen wir das simultane Lernen ein, um sicherzustellen, dass die erlernten Parameter unter Verwendung von Echtzeit- und historischen Daten zusammen mit den tatsächlichen Werten konvergieren. Außerdem verwenden wir die Barriere-Lyapunov-Funktion, um die Sicherheit (dargestellt durch vorgegebene Grenzen für den Tracking Error) in die Stabilität zu integrieren. Die von uns entwickelte stabile Regelungsstrategie, die auf der erlernten Dynamik basiert, könnte eine sichere Nachführung ermöglichen und gleichzeitig die Unsicherheit während des Betriebs reduzieren. Gleichzeitiges Lernen ist jedoch auf die Bewältigung parametrischer Unsicherheiten beschränkt und erfordert Kenntnisse der Modellstruktur. Bezüglich der Grenzen des gleichzeitigen Lernens, nutzen wir dann zeitverzögerte Signale, um inkrementelle Systeme zu konstruieren, die äquivalente Darstellungen der ursprünglich gesteuerten Anlagen sind, aber ohne explizites Modellwissen auskommen, um die modellfreie Steuerung zu erleichtern. Die verwendeten zeitverzögerten Signale degradieren den Einfluss von Unsicherheiten auf Regelstrecken zum Effekt eines nachweislich begrenzten Zeitverzögerungsschätzfehlers auf die formulierten inkrementellen Systeme. Durch einen Input-to-State Stabilitätsansatz, der mit Barriere-Lyapunov-Funktionen und Backstepping kombiniert wird, analysieren wir rigoros den Zeitverzögerungsschätzungsfehler während des rekursiven Reglerentwurfsprozesses. Dadurch können wir eine nachweislich sichere Regelung unter Unsicherheit mit hoher Genauigkeit bei der Nachführung realisieren. Die vorangegangenen Ansätze verfolgen einen gemeinsamen, von der Planung und Regelung entkoppelten Ansatz, um eine sichere Ausführung unter Unsicherheiten zu erreichen. Alternativ dazu integrieren wir die Wahrnehmung mit den Regelungsebenen, um sichere Lernsysteme zu schaffen, die gegenüber unvorhergesehenen Umgebungen widerstandsfähig sind. Erreicht wird dies durch die quadratische Optimierung auf der Regelungsebene mit den aus

den Wahrnehmungssignalen erlernten Beschränkungen, die als momentane lokale Kontroll-Barriere-Funktionen und zielgetriebene Kontroll-Lyapunov-Funktionen bezeichnet werden. Der integrierte Ansatz umgeht die Lücken zwischen den einzelnen Ebenen des üblichen Paradigmas von entkoppelten Kartographieren, Plannen und Nachverfolgen, um die theoretisch garantierte Kollisionsvermeidung und Konvergenz zum Ziel zu erleichtern. Durch die Verwendung von momentanen lokalen Sensordaten werden rechnerisch günstige und sichere Kontrollstrategien mit schneller Anpassung an verschiedene unsicheren Umgebungen gefördert, ohne eine Karte zu erstellen.

Das Reinforcment Learning und die Regelungstechnik werden in Teil II für sicheres Lernen und Optimierung in Anwesenheit von Ungewissheiten zusammengeführt. Der auf Reinforcement Learning basierende Optimierungsrahmen ist mit Sicherheits- und Robustheitsgarantien durch theoretische Analysewerkzeuge aus dem Bereich der Regelungstechnik ausgestattet. Wir schlagen zunächst ein risikosensitives, auf Reinforcement Learning basierendes Regelungssystem vor, um die Aufgabenerfüllung und die Erfüllung von Bedingungen in einer gestörten Umgebung gemeinsam zu optimieren. Insbesondere entwerfen wir risikosensitive Strafbedingungen, um risikobewusste Wertfunktionen zu konstruieren, die unsichere Verhaltensweisen bestrafen. Die obige risikobewusste Wertfunktion wird durch eine Sicherheitsbewertung unter Verwendung eines off-policy Gewichts-Update-Regel approximiert. Während des Lernprozesses ist die zugehörige approximative optimale Regelstrategie in der Lage, sowohl die Eingangs- als auch die Zustandsbeschränkungen bei Störungen zu erfüllen. Die modellfreie Eigenschaft des Reinforcement Learnings wird bei dem oben erwähnten Ansatz jedoch gegen theoretische Garantien eingetauscht. Insbesondere werden Informationen über frühere Modelle verwendet, um nachweisbare Sicherheit und Stabilität unter Unsicherheit zu präsentieren. Daher entwickeln wir anschließend eine zeitverzögerte, dateninformierte Methode des Reinforcement Learnings, die als inkrementelle adaptive dynamische Programmierung bezeichnet wird, um das optimale Regelungsproblem näherungsweise auf modellfreie Weise zu lösen und dabei strenge Stabilitätsgarantien zu erhalten. Insbesondere liefern die zeitverzögerten Daten den Wertfunktionslernprozess über eine modellfreie Darstellung der ursprünglichen Regelstrecke. Auf diese Weise könnten wir eine modellfreie Regelung realisieren und haben auch eine mathematische Form der Dynamik, um eine strenge theoretische Analyse unter Verwendung umfangreicher Analysewerkzeuge aus dem Bereich der Regelungstechnik durchzuführen. Der von uns entwickelte Ansatz der inkrementellen adaptiven dynamischen Programmierung dient als effizientes Werkzeug zum Erlernen der Lösungen für das Problem der optimalen Bewegungsplanung mit Rückkopplung und das Problem der optimalen Nachführregelung.

# Contents

# Notation

## Acronyms and Abbreviations

| | |
|---|---|
| **ADP** | Adaptive (approximate) dynamic programming |
| **ARE** | Algebraic Riccati equation |
| **ANN** | Artificial neural network |
| **BLF** | Barrier Lyapunov function |
| **CL** | Concurrent learning |
| **CBF** | Control barrier function |
| **CLF** | Control Lyapunov function |
| **E-L** | Euler-Lagrange |
| **GP** | Gaussian process |
| **HJB** | Hamilton-Jacobi-Bellman |
| **HJI** | Hamilton-Jacobi-Issac |
| **ISS** | Input-to-state stable |
| **MPC** | Model predictive control |
| **PE** | Persistence excitation |
| **QP** | Quadratic programming |
| **RBF** | Radial basis function |
| **RL** | Reinforcement learning |
| **SVM** | Support vector machine |
| **TDE** | Time delay estimation |

# Conventions

## Subscripts and Superscripts

| | |
|---|---|
| $\hat{x}$ | Estimate of $x$ |
| $x^*$ | Optimal value of $x$ |
| $x^{-1}$ | Inverse of $x$ |
| $x^\dagger$ | Moore-Penrose pseudoinverse of $x$ |
| $x^\top$ | Transpose of $x$ |
| $x_i$ | $i$-th element of $x$ |
| $x_{ij}$ | $ij$-th element of $x$ |

# Matrix and Vector Norm

The norm definitions of the vector $a \in \mathbb{R}^n$, and the matrix $A \in \mathbb{R}^{n \times m}$ :

| | |
|---|---|
| $\ell_1$ **norm** | $\|a\| = \sqrt{\sum_{i=1}^{n} \lvert a_i \rvert^2}$ |
| $\ell_\infty$ **norm** | $\|a\|_\infty = \max_i \lvert a_i \rvert$ |
| **Frobenius norm** | $\|A\| = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{m} \lvert A_{ij} \rvert^2}$ |

# Number Sets

| | |
|---|---|
| $\mathbb{R}$ | Real numbers |
| $\mathbb{R}^+$ | Positive real numbers |
| $\mathbb{N}^+$ | Positive natural numbers |

# Symbols

| | |
|---|---|
| $\mathrm{diag}(a)$ | Diagonal matrix with the $i$-th diagonal entry as $a_i$ |
| $a \preceq (\prec) b$ | Component-wise comparison of vectors $a$ and $b$, $a_i \leq (<) b_i$ |
| $\lambda_{\min}(A)$ | Minimum eigenvalues of a symmetric real matrix $A$ |

| | |
|---|---|
| $\lambda_{\max}(A)$ | Maximum eigenvalues of a symmetric real matrix $A$ |
| $I_{n\times m}$ | $n \times m$ dimensional identity matrix |
| $\mathrm{Int}(\mathbb{S})$ | Interior of the set $\mathbb{S}$ |
| $\partial\mathbb{S}$ | Boundary of the set $\mathbb{S}$ |
| $\nabla(\cdot)$ | Partial derivative $\partial(\cdot)/\partial x$ |

# 1 Introduction

The provable safe execution of uncertain systems to complete predetermined tasks is required for scenarios such as robot manipulators for public services, and quadrotors for search and rescue in cave environments. Both control and learning communities attempt to build paradigms to achieve provably safe control under uncertainty with guaranteed performance (considered for given missions), although with different focuses. Traditional control methods (set-theoretic methods in particular) are favoured with formal guarantees of indexes such as safety and stability. However, their adaptation ability to unforeseen contexts is limited. Learning approaches (reinforcement learning specifically) allow generalization towards different environments; however, no theoretical guarantees are provided. Therefore, it is natural to bridge learning and control communities to design control schemes that enjoy rigorous theoretical guarantees and generalization towards diverse tasks and environments.

**Set-theoretic methods.** Sets are appropriate tools to specify constraints concerning safety issues and design specifications [1]. Thus, it is attractive to investigate our interested problem in a set-theoretic context. Recently, control barrier function (CBF) has emerged as a promising set-theoretic tool to enforce safety at a control level, see [2], [3] and the references therein. Besides, barrier Lyapunov function (BLF) [4], [5], combined with properties of barrier and Lyapunov functions, is often used with backstepping to stabilize controlled plants while confining certain states into prior-given safe regions. The effectiveness of both CBF and BLF highly relies on accurate dynamics that are not always available. Towards model uncertainties, function approximation based methods are widely utilized. The present function approximation related works could be categorized in terms of different approximation schemes, such as polynomials [6], trigonometric series [7], orthogonal functions [8], splines [9], and neural networks (NNs) [10], etc. Among these approximation schemes, NNs play a vital role in learning-based control methods [11]. Normally, the NN approximation scheme is firstly adopted to learn a model beforehand, and then a control law is designed based on the learned model. However, the guaranteed weight convergence to the actual value is out of consideration in most of NN approximation scheme related works. Furthermore, the influence of unavoidable approximation errors on safety issues remains to be rigorously analyzed and addressed.

**Reinforcement learning approaches.** Reinforcement learning (RL) provides a mathematical formulation for learning-based control strategies [12] and has shown superior performance in multiple scenarios [13]–[15]. Although the distinguishable model-free feature of RL overcomes the difficulty of applying traditional model-based control methods to the unknown (or hardly modelled) plants, the rigorous system stability analysis is not provided in most of related works [16], [17]. However, a system without stability guarantee is potentially dangerous [18]. More recently, adaptive dynamic programming (ADP) [19]–[21] has emerged as a promising control-theoretic RL subfield featured for available system stability proofs. ADP, implemented as an actor-critic or a critic-only NN learning structure [22], [23], forwardly solves the algebraic Riccati equation (ARE) or Hamilton-Jacobi-Bellman (HJB) equation via value function approximations. Although traditional ADP has been widely adopted to investigate stability and robustness issues, input and state constraint satisfaction during the learning process, mainly investigated for safety concerns (e.g., restrictions on torques, joint angels, and angular velocities of robot manipulators), has not yet been efficiently addressed. Violations of any constraints could lead to severe consequences such as damage to physical components. Note that the provided stability proof of ADP compromises the attractive model-free feature of RL since a mathematical form of dynamics is required to present the rigorous closed-loop stability analysis. Even though the required explicit knowledge of dynamics could be avoided by using add-on techniques such as NNs [24]–[26], Gaussian process (GP) [27], or observers [28], the accompanying identification processes further increase analytical complexities, computational loads, and parameter tuning efforts. Thereby, one computationally efficient and easily implemented RL based control strategy, favoured with both a model-free feature and a stability guarantee, is required.

## 1.1 Challenges

The brief introduction illustrated above encourages us to formulate the following challenges revolving around the provable safe control under uncertainty with guaranteed performance from set-theoretic and RL perspectives.

### Provable Safety under Uncertainty

Through convexing safe regions (predetermined or computed [29]) or unsafe spaces (often overly approximated), the safety problem is often investigated on the basis of accurate dynamics via tools such as CBFs [2], BLFs [4], forward (backward) reachable sets [30], model predictive control [31], penalty functions [32], and contraction theory [33], etc. The trustworthy safety checks of the safe control tools mentioned above build upon available perfect dynamics. In the case of only uncertain dynamics accessible to practitioners, current works use parametric or non-parametric methods to provide learned dynamics for safety checks. However, the gap concerning safety check built on dynamics still in learning processes (i.e., potentially inaccurate dynamics) is not fully considered. Assumed available [33] or online/offline estimated uncertainty bounds [34] provide designers with avenues to analyse the influence of uncertainty on safety rigorously. However, the utilized conservative uncertainty bounds might reduce allowable regions into unsafe regions. This results in conservative behaviours that deteriorate performance. Therefore, rigorously quantifying, analysing, and addressing the influence of uncertainty on safety is yet to be determined.

### Guaranteed Weight Convergence

Regrading the performance issue in adaptive control, in general, the weight convergence is not required. It has been illustrated in [35], [36] that weight estimation errors may not converge to zero (indeed it may not converge at all) even though an acceptable tracking performance is achieved. However, it is undesirable to verify safety using a constantly changing model. Thereby, the guaranteed weight convergence, offering an ensured accurate learned dynamics, is required to ensure safety under model uncertainties. Conventionally, the persistence of excitation (PE) condition is used to check the weight convergence. The weight convergence to the actual value is guaranteed if the PE condition is satisfied [36], [37]. Among existing works [20], [38], the PE condition could be satisfied by incorporating external noises to control inputs. However, this method lacks practicability given that the direct incorporation of external noises into control inputs may suffer a degradation of control performance, and a waste of energy, etc. Most importantly, the incorporated external noise, ignored during the theoretical analysis process, would invalidate the provided safety and stability proofs. Hence, a fundamental problem about the guaranteed weight convergence is yet to be discussed, and one practical as well as efficient method to provide the required excitation remains explored.

### RL for Control with Theoretical Guarantees

RL serves as one promising framework for synthesising control policies to satisfy multiple objectives. However, RL is predominately used in simulated environments due to the lack of guaranteed safety and stability. The difficulty of providing stability proofs for RL results from noninterpretable neural network policies, unknown system dynamics, and random explorations. The control-theoretic RL method ADP relieves the stability issue via a linear approximator, a given or an identified model, and an analytical deterministic optimal control policy. However, the guaranteed safety during the learning process has not been efficiently addressed in ADP. Most of previous ADP related works adopt the system transformation technique to deal with state constraints [28], [39], [40]. This method, nonetheless, is limited to simple constraint forms, e.g., restricted working space in a rectangular form. Besides, although general state constraints could be tackled by the well-designed penalty functions [41], [42], which become dominant in the optimization process when possible constraint violation happens and thus punish potential dangerous behaviours, no strict constraint satisfaction proofs are provided. However, in certain cases such as human-robot interaction scenarios, even the violation of safety-related constraints in a small possibility is unacceptable. Through the above analysis, it is meaningful to use theoretical analysis tools from the control community to inform the RL based policy with stability and safety guarantees. These theoretical guarantees form the basis to use RL for broad applications.

### Curse of Dimensionality and Complexity

The optimal control problem is usually solved via the minimum principle of Pontryagin, or dynamic programming [43]. Using dynamic programming to solve the optimal control problem faces the notorious curse of dimensionality problem, i.e., the volume of the state space grows quickly as the number of dimension grows [44]. The RL based ADP mitigates the curse of dimensionality problem by forwardly solving the ARE or HJB in an approximation way. However, the so-called curse of complexity appears. In particular, the number of activation functions required for the accurate value function approximation grows exponentially with the system dimension [45]. Theoretically, practitioners could seek a sufficient large NN to achieve a satisfying approximation of a high-dimensional value function [46]. However, practically, this is nontrivial considering that appropriate activation functions are usually chosen by trial and error. This process is tedious and time-consuming. Even though a suitable set of activation functions and appropriate hyperparameters are found through engineering efforts, the accompanying computation load jeopardises the realtime performance of the associated weight update law and control strategy [47]. Thus, experimental validations of ADP based control strategy on a high-dimensional system is seldom found in existing works. Applying ADP to solve the optimal control problems of high-dimension systems remains to be explored.

## 1.2 Major Contributions and Dissertation Outline

This dissertation presents our efforts, set-theoretic methods in Part I (Chapter 2–Chapter 4) and reinforcement learning approaches in Part II (Chapter 5–Chapter 7), to solve the challenges formulated above to enable autonomous systems to operate safely and meet task requirements even under uncertainties.

### Safe Parameter Learning and Control of Robot Manipulators (Chapter 2)

This chapter online learns the uncertain dynamics of robot manipulators during the operation process, and safely improves the tracking performance gradually. Our developed control strategy deals with the following problems simultaneously: safety issues regarding output constraints, guaranteed performance concerning tracking errors, and parametric uncertainties of robot manipulators. In particular, we first combine the safety objective with the performance requirement. Then, we use BLFs to account for the safety and performance-related constraints simultaneously. Besides, the torque filtering technique is integrated into concurrent learning to avoid using joint acceleration information for the parameter learning. Finally, a novel double regressor matrix technique is developed to enable the combination of the BLF based control and the torque filtering augmented concurrent learning aided online system identification feasible. Numerical and experimental results validate that our proposed strategy drives uncertain robot manipulators to track the desired trajectories with guaranteed safety and performance.

*The results presented in Chapter 2 have been published in IEEE Transactions on System, Man, and Cybernetics [48].*

### Provably Safe Control under Uncertainty via ISS-PS-BLF (Chapter 3)

This chapter bridges the gap between the safe planning and the guaranteed performance control to accomplish provable safe execution of controlled plants suffering uncertainties. This is accomplished by considering the achievable performance bound of the control level into the planning level. In particular, we first utilize time-delayed signals to formulate an uncertain and disturbed dynamics into an equivalent incremental system without using explicit model knowledge. Then, our proposed input-to-state with provable safety barrier Lyapunov function (ISS-PS-BLF) is utilized with backstepping to design a guaranteed performance tracking controller based on the above formulated incremental system. The realizable tracking performance bound of the designed tracking controller is further considered in the planning level to generate safe reference trajectories. The effectiveness of our developed safe planning and guaranteed performance tracking control scheme is numerically validated via the task-space tracking of robot manipulators and the safe flight of quadrotors.

### Constraint Learning for Safe Operation in Unforeseen Region (Chapter 4)

This chapter presents an integrated perception and control approach that provides limited-performance mobile robots with a low-cost solution (regarding hardware requirements and computation loads) to the safe operation problem in uncertain environments. In particular, the instantaneous local control barrier functions (IL-CBFs) reflecting potential collisions and the goal-driven control Lyapunov functions (GD-CLFs) encoding incrementally discovered subgoals are first online learned from perceptual signals. Then, the learned IL-CBFs are united with GD-CLFs in the context of a quadratic programming (QP) to generate safe feedback control strategies. Rather importantly, an optimization over the admissible control space of IL-CBFs is conducted to improve the QP feasibility. Numerical simulations are conducted to reveal the effectiveness of our proposed safe feedback control strategy that drives mobile robots to safely reach the destination incrementally in uncertain environments.

*The contents shown in Chapter 4 have been submitted for possible publication in IEEE Robotics and Automation Letters. The associated arXiv version is [49]*

### Joint Optimization for Task Performance and Safety (Chapter 5)

This chapter proposes an off-policy risk-sensitive RL based control framework to jointly optimize task performance and constraint satisfaction in a disturbed environment. The risk-aware value function, constructed using the pseudo control and the risk-sensitive input and state penalty terms, is introduced to convert the original constrained robust stabilization problem into an equivalent unconstrained optimal control problem. Then, an off-policy single critic reinforcement learning algorithm is developed to learn the approximate solution to the above constructed risk-aware value function. During the learning process, the associated approximate optimal control policy satisfies both input and state constraints under disturbances. By replaying experience data to the off-policy weight update law of the critic neural network, the weight convergence is guaranteed. Moreover, online and offline algorithms are developed to serve as principled ways to record informative experience data to achieve a sufficient excitation required for the weight convergence. The proofs of system stability and weight convergence are provided. Simulation results reveal the validity of our proposed control framework.

*The contents concerning robust constrained optimal stabilization shown in Chapter 5 have been submitted for possible publication in IEEE Transactions on System, Man, and Cybernetics and the associated arXiv version is [50]. The numerical simulations concerning optimal tracking control with prescribed performance come from our arXiv paper [51].*

### Safe Approximate Optimal Control via Barrier Certified RL (Chapter 6)

This chapter presents a new formulation for model-free robust constrained optimal regulation control of continuous time nonlinear systems. The proposed RL based approach, referred to as incremental adaptive dynamic programming (IADP), utilizes measured input-state data to allow the design of the approximate optimal incremental control strategy, stabilizing the controlled system to the target point under model uncertainties, environmental disturbances, and satisfying input saturation. In particular, we first use sensor data to reduce the requirement of a complete dynamics, where input-state data is adopted to construct an incremental dynamics that reflects the system evolution in an incremental form. Then, the resulting incremental dynamics serves to design the approximate optimal incremental control strategy based on RL, which is implemented as a simplified single critic learning structure to get the approximate solution to the value function of the HJB equation. Rather importantly, we incorporate a time delay estimation error bound related term into the cost function, whereby the unintentionally introduced time delay estimation error is attenuated during the optimization process. Finally, one safety filter is introduced to minimally correct the learned approximate optimal control policy to ensure safe execution. The proofs of system stability and weight convergence are provided. Numerical simulations are conducted to validate the effectiveness and superiority of our proposed IADP, especially regarding the enhanced robustness.

*The contents concerning IADP presented in Chapter 6 have been published in International Journal of Robust and Nonlinear Control [52]. The arXiv version is [53].*

### Time-Delayed Data Informed RL for Optimal Tracking Control (Chapter 7)

To achieve safe execution in uncertain environments, the planned or replanned safe reference trajectories should be accurately tracked by a high-accuracy and robust tracking controller. Thus, this chapter investigates the optimal tracking control problem (OTCP) with preferences on tracking accuracy and robustness. This chapter extends the IADP developed in Chapter 6 to learn the approximate solution to the OTCP. Departing from available solutions to the OTCP, our developed tracking control scheme settles the curse of complexity problem in value function approximation from a decoupled way, circumvents the learning inefficiency regarding varying desired trajectories by avoiding introducing a reference trajectory dynamics into the learning process, and requires neither an accurate nor identified dynamics using time-delayed signals to facilitate model-free control. Specifically, we first convert the intractable OTCP of a high-dimensional uncertain system into multiple manageable sub-problems of low-dimensional incremental error subsystems. Then, the resulting sub-problems are approximately solved by a parallel critic learning structure. The proposed tracking control scheme is developed with rigorous theoretical analysis of system stability and weight convergence, and validated experimentally on a 3-DoF robot manipulator.

*The contents shown in Chapter 7 have been submitted for possible publication in IEEE Transactions on Cybernetics. The associated arXiv version is [54].*

# Part I

# Set-Theoretic Methods

# Concurrent Learning-Based Adaptive Control with Guaranteed Safety and Performance

2

This chapter investigates the tracking control problem of an uncertain $n$-link robot manipulator with guaranteed safety and performance. We employ BLFs and backstepping to design a guaranteed tracking performance controller based upon dynamics learned from concurrent learning (CL). Our developed joint tracking controller could combine with joint-level robot motion planning algorithms [55], [56] to achieve safe operation of robot manipulators. This chapter is organized as follows. Section 2.1 firstly introduces the preliminaries and the problem formulation. Then, the torque filtering (TF) technique is illustrated in Section 2.2, which serves to Section 2.3 to construct a TF-CL aided parameter estimation update law for online identification of unknown systems without using joint acceleration information. The parameter convergence is guaranteed by exploiting current and historical data simultaneously. The developed TF-CL technique enjoys practicability compared with common methods that need to incorporate external noises to satisfy the PE condition required for the parameter convergence. Based on the learned model, Section 2.4 elucidates the recursive controller design process using the backstepping technique, and presents stability proofs as well as compact sets of both tracking errors and system outputs. By ensuring the boundness of BLFs, the system outputs and the tracking errors are proved to lie in the safety set and performance set, respectively. Numerical simulations in Section 2.5 and experimental validations in Section 2.6 illustrate the effectiveness of our proposed control strategy. Summaries are finally provided in Section 2.7.

## 2.1 Preliminaries and Problem Formulation

The dynamics of an $n$-link robot manipulator follows the Euler-Lagrange (E-L) equation

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + F\dot{q} = \tau, \tag{2.1}$$

where $M(q) : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ is the symmetric positive definite inertia matrix; $C(q,\dot{q}) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{n\times n}$ is the matrix of centrifugal and Coriolis terms; $G(q) : \mathbb{R}^n \to \mathbb{R}^n$ represents gravitational terms, and $F \in \mathbb{R}^{n\times n}$ denotes values of viscous friction; $q$, $\dot{q}$, and $\ddot{q} \in \mathbb{R}^n$ are the vectors of joint angles, velocities and accelerations, respectively; $\tau \in \mathbb{R}^n$ represents the vector of input torques applied at each joint.

**Property 1.** *[57] The left side of the system equation* (2.1) *can be written as the following linear in parameter (LIP) form*

$$Y(q,\dot{q},\ddot{q})\theta^* = \tau, \tag{2.2}$$

*where $Y(q,\dot{q},\ddot{q}) : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{n\times m}$ is the regressor matrix; $\theta^* \in \mathbb{R}^m$ is the desired coefficient vector of the E-L equation.*

**Remark 2.1.** *Property 1 exploits the known model properties at hands to construct the regression matrix $Y(q,\dot{q},\ddot{q})$. Considering the E-L equation* (2.1)*, model uncertainties include varying masses or lengths of joints, varying friction parameters, and unknown payloads. The aforementioned model uncertainties can all be incorporated into the coefficient vector $\theta^*$ of* (2.2)*. Note that for the model-free NN approximation scheme [10], the known physical structure of the investigated system is abandoned, which usually suffers from the well known sample inefficiency problem.*

Let $x_1 = q$ and $x_2 = \dot{q}$, the E-L equation (2.1) is written in the state-space form as

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= M^{-1}(x_1)(\tau - C(x_1,x_2)x_2 - G(x_1) - Fx_2), \\ y &= x_1, \end{aligned} \tag{2.3}$$

where $y \in \mathbb{R}^n$ is the system output that denotes the joint angles of the $n$-link robot manipulator (2.1), and assuming that it lies in the following set

$$\mathcal{C} = \{y \in \mathbb{R}^n : k_e \prec y \prec k_f\}. \tag{2.4}$$

Here we consider a trajectory tracking control problem where the robot manipulator is driven to track the desired trajectory $y_d \in \mathbb{R}^n$ precisely. Throughout this chapter, we confine ourselves that the desired trajectory $y_d$ satisfies the following assumption.

**Assumption 2.1.** *The desired trajectory $y_d$ satisfies $-\underline{y}_d \preceq y_d \preceq \overline{y}_d$, where $\underline{y}_d$ and $\overline{y}_d$ are positive constant vectors.*

Based on the system output $y$ of (2.3) and the desired trajectory $y_d$, we define the tracking error $e_1 \in \mathbb{R}^n$ as

$$e_1 = y - y_d. \tag{2.5}$$

For the safety issues considered during the trajectory tracking process, following the barrier function definition illustrated in [2], here the safety set regarding the system output $y$ is defined as

$$\mathcal{S} = \{y \in \mathbb{R}^n : h(y) \leq 0\}, \tag{2.6}$$

where $h(y) : \mathbb{R}^n \to \mathbb{R}$ is a continuous function. The explicit form of $h(y)$ is determined by considering various safety issues during the tracking process. As for the investigated $n$-link robot manipulator, considering the human-robot interactions or limited spaces, its safety set is usually defined as an allowable operation region [58], [59] that follows

$$\bar{\mathcal{S}} = \{y \in \mathbb{R}^n : k_c \prec y \prec k_d\}, \tag{2.7}$$

where $k_c = [k_{c_1}, \cdots, k_{c_n}]^\top \in \mathbb{R}^n$ and $k_d = [k_{d_1}, \cdots, k_{d_n}]^\top \in \mathbb{R}^n$ are known constant vectors determined by controller designers. The safety set $\bar{\mathcal{S}}$ in (2.7) is a representative and explicit form of $\mathcal{S}$ in (2.6). Note that $k_c \prec -\underline{y}_d$ and $\overline{y}_d \prec k_d$ hold, i.e., $y_d$ lies in the safety set $\bar{\mathcal{S}}$.

For the performance issues, we demand that the tracking error $e_1$ (2.5) lies in the following performance set

$$\mathcal{P} = \{e_1 \in \mathbb{R}^n : -k_a \prec e_1 \prec k_b\}, \tag{2.8}$$

where $k_a = [k_{a_1}, \cdots, k_{a_n}]^\top \in \mathbb{R}^n$ and $k_b = [k_{b_1}, \cdots, k_{b_n}]^\top \in \mathbb{R}^n$ are predefined constant vectors. According to (2.5), the resulting working space based on the desired tracking error bound (2.8) would be

$$\bar{\mathcal{P}} = \left\{ y \in \mathbb{R}^n : -k_a - \underline{y}_d \prec y \prec k_b + \overline{y}_d \right\}. \tag{2.9}$$

To counter the constraints concerning safety in (2.7) and performance in (2.8), BLF [4], [5] emerges as an efficient tool. To deal with both symmetric and asymmetric constraints, a simple indicator function based BLF is proposed as

$$V(z) = p(z)\frac{z^2}{k_u^2 - z^2} + (1 - p(z))\frac{z^2}{k_l^2 - z^2}, \tag{2.10}$$

where $z \in \mathbb{R}$ is the system state; and $k_l$, $k_u \in \mathbb{R}$ are constraint bounds; When $z \to k_l$ or $z \to k_u$, $V(z) \to \infty$; $p(z)$ is an indicator function that follows

$$p(z) = \begin{cases} 1, & z > 0 \\ 0, & z \leq 0 \end{cases}. \tag{2.11}$$

According to Definition 2 in [4], the proposed BLF in (2.10) is an effective BLF.

**Remark 2.2.** *From the perspective of the guaranteed performance represented by* (2.8)*, prescribed performance control (PPC) [60] is closely related to our work, which exploits the prescribed performance function (PPF) based system transformation technique to guarantee that, the tracking error converges to an explicit residual set, the convergence rate is no less than a predefined value, and a maximum overshoot is less than a prespecified constant. However, although multiple performance criteria could be provided by PPC, we found in practice that its efficient application requires extensive parameter tuning efforts because the adopted PPF is sensitive and easier lead to singularity. Moreover, the system transformation process results in additional complexity. Thus, a simple BLF is chosen here to achieve guaranteed performance and safety. Although no explicit values of the final residual set and convergence rate are provided by our designed BLF based control strategy in Section 2.4, both simulation and experiment results in Section 2.5 and Section 2.6 have shown satisfying performance comparable to the PPC based work [60].*

For the investigated tracking control problem, the priority of safety is over performance. To improve the tracking performance while always guaranteeing safety, $\mathcal{C} \subseteq \bar{\mathcal{S}}$ and $\mathcal{C} \subseteq \bar{\mathcal{P}}$ should be satisfied together. For the purpose of achieving these considerations simultaneously, we could firstly choose the values of $-k_a$ and $k_b$ to meet $\bar{\mathcal{P}} \subseteq \bar{\mathcal{S}}$, and then design a tracking controller to enforce $\mathcal{C} \subseteq \bar{\mathcal{P}}$. For example, consider a scenario that a robot manipulator works close to humans where the pre-planned $y_d$ for given tasks ensures collision free with humans. To guarantee the collision avoidance while accomplishing the predefined tasks, we need to restrict the operation range of the robot manipulator such that $\mathcal{C} \subseteq \bar{\mathcal{S}}$, and enable the robot manipulator to track $y_d$ precisely to satisfy $\mathcal{C} \subseteq \bar{\mathcal{P}}$, respectively. The aforementioned safety and performance requirements could be integrated together by choosing the explicit values (i.e., $-k_a$ and $k_b$) of the guaranteed tracking performance such that $\bar{\mathcal{P}} \subseteq \bar{\mathcal{S}}$ holds. Then, a BLF based controller that drives the robot manipulator to track $y_d$ with the guaranteed tracking performance also enforces the executed trajectory to lie in the restricted operation range at the same time. Although it seems to be a conservative approach, comparing to works that can only consider partial objectives of performance [61] or safety [62], the resulting BLF based controller could drive the robot manipulator to track the desired trajectory while satisfying requirements of both safety and performance together.

Based on the aforementioned settings, the tracking control problem with guaranteed safety and performance is formulated as follows.

**Problem 2.1.** *Given the uncertain robot manipulator* (2.1)*, and the desired trajectory $y_d$ within the prior known safety set $\bar{\mathcal{S}}$* (2.7)*. Choose appropriate bounds for the performance set $\mathcal{P}$* (2.8)*, and design a stable adaptive control strategy based on the proposed BLF* (2.10) *to drive the uncertain robot manipulator to track the desired trajectory $y_d$ while satisfying requirements of safety characterized by $\bar{\mathcal{S}}$ and performance denoted as $\mathcal{P}$ together.*

## 2.2 Torque Filtering Technique

For the LIP form of the E-L equation given in (2.2), measurements of the acceleration $\ddot{q}$ are required to construct the regressor matrix $Y(q, \dot{q}, \ddot{q})$. Since the information of joint acceleration is sensitive to measurement noises, it is not applicable to use it directly to design a controller. To eliminate the need for this information, the torque filtering technique is adopted here to reformulate the original LIP form (2.2) to get a new equivalent LIP form without requirements for the joint acceleration information. Comparing to the common Kalman filter that highly depends on prior knowledge (e.g., noises to be filtered) and requires extensive parameter tuning efforts [63], the adopted torque filtering technique is a simple and easily implemented method for practical applications.

To facilitate the introduction of the torque filtering technique, two auxiliary vectors $h(q, \dot{q}) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ and $g(q, \dot{q}) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ are firstly defined as

$$\begin{aligned} h(q, \dot{q}) &= M(q)\dot{q} = Y_1(q, \dot{q})\theta^*, \\ g(q, \dot{q}) &= -\dot{M}(q)\dot{q} + C(q, \dot{q})\dot{q} + G(q) + F\dot{q} = Y_2(q, \dot{q})\theta^*, \end{aligned} \tag{2.12}$$

where $Y_1(q, \dot{q}) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{n\times m}$ and $Y_2(q, \dot{q}) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{n\times m}$ are two new regressor matrices without incorporating the information of $\ddot{q}$.

Based on the auxiliary vectors in (2.12), the system equation (2.1) is rewritten as

$$\dot{h}(q, \dot{q}) + g(q, \dot{q}) = (\dot{Y}_1(q, \dot{q}) + Y_2(q, \dot{q}))\theta^* = \tau, \tag{2.13}$$

where $\dot{h}(q, \dot{q}) = \dot{M}(q)\dot{q} + M(q)\ddot{q} = \dot{Y}_1(q, \dot{q})\theta^*$.

The advantage of writing the robot manipulator model in the form (2.13) is that this new equivalent form of (2.1) has been separated in a way that allows $\ddot{q}$ to be filtered out. To filter out $\ddot{q}$ existing in $\dot{Y}_1(q, \dot{q})$, a linear stable filter is introduced as

$$f(s) = \frac{1}{ks + 1}, \tag{2.14}$$

where $s$ is the Laplace operator and $k \in \mathbb{R}$ is a time constant. By filtering (2.13) based on (2.14), we get the filtered version of (2.13) as

$$\dot{h}_f(q, \dot{q}) + g_f(q, \dot{q}) = \tau_f, \tag{2.15}$$

where $\dot{h}_f(q,\dot{q}) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ and $g_f(q,\dot{q}) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ are the filtered versions of $\dot{h}(q,\dot{q})$ and $g(q,\dot{q})$, respectively. $\tau_f \in \mathbb{R}^n$ is the filtered version of $\tau$.

Based on (2.13), the corresponding LIP form of the filtered system (2.15) reads

$$(\dot{Y}_{1_f}(q,\dot{q}) + Y_{2_f}(q,\dot{q}))\theta^* = \tau_f, \tag{2.16}$$

where $Y_{1_f}(q,\dot{q}) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{n\times m}$ and $Y_{2_f}(q,\dot{q}) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{n\times m}$ are the filtered versions of the regressor matrices $Y_1(q,\dot{q})$ and $Y_2(q,\dot{q})$, respectively.

For the filter given in (2.14), the filtered variables and their original forms satisfy the following equations

$$\begin{gathered}
k\dot{Y}_{1_f}(q,\dot{q}) + Y_{1_f}(q,\dot{q}) = Y_1(q,\dot{q}), \quad Y_{1_f}(q,\dot{q})|_{t=0} = 0, \\
k\dot{Y}_{2_f}(q,\dot{q}) + Y_{2_f}(q,\dot{q}) = Y_2(q,\dot{q}), \quad Y_{2_f}(q,\dot{q})|_{t=0} = 0, \\
k\dot{\tau}_f + \tau_f = \tau, \quad \tau_f|_{t=0} = 0.
\end{gathered} \tag{2.17}$$

Substituting the first equation of (2.17) into (2.16), finally we get the filtered LIP form of the E-L equation (2.1) as

$$Y_f(q,\dot{q})\theta^* = \tau_f, \tag{2.18}$$

where $Y_f(q,\dot{q}) = (Y_1(q,\dot{q}) - Y_{1_f}(q,\dot{q}))/k + Y_{2_f}(q,\dot{q}) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{n\times m}$ is the new filtered regressor matrix without requirements for the joint acceleration information.

Now the new filtered regressor matrix $Y_f(q,\dot{q})$ and the resulting filtered LIP form (2.18) can be adopted to identify the unknown coefficient vector $\theta^*$ without using the joint acceleration knowledge, which is detailly clarified in the next section.

## 2.3 Concurrent Learning Aided System Identification

Since the ideal coefficient vector $\theta^*$ in (2.18) is unknown, and only the estimated parameter vector $\hat{\theta}$ is available, the identification problem to be addressed here is to obtain $\hat{\theta}$ online based on the system input $u$, the output state $y$, and the filtered regression matrix $Y_f$. The online identification of $\hat{\theta}$ is an adaptive parameter estimation problem where the PE condition needs to be satisfied before the estimated parameters converge to the desired values. Unlike common methods that introduce external noises to satisfy the PE condition, based on the LIP form in (2.18), the TF-CL method is adopted here to guarantee the parameter convergence by utilising both current and historical data.

### 2.3.1 Parameter Estimation Update Law

Denote the parameter estimation error as $\tilde{\theta} = \hat{\theta} - \theta^* \in \mathbb{R}^m$. Then, the corresponding model approximation error follows

$$e_f = Y_f\tilde{\theta}. \tag{2.19}$$

Define the quadratic cost of the approximation error as $V_{e_f} = 1/2e_f^\top e_f$. Following the common gradient descent method to minimize $V_{e_f}$, the adaptive parameter estimation update law is derived as

$$\dot{\hat{\theta}} = -\Gamma Y_f^\top e_f, \tag{2.20}$$

where $\Gamma \in \mathbb{R}^{m \times m}$ is a constant positive definite matrix. It is well known that the estimated $\hat{\theta}$ converges to the desired $\theta^*$, iff the regression matrix $Y_f$ satisfies the PE condition [64]:

$$\int_t^{t+T} Y_f^\top(\tau) Y_f(\tau) d\tau \geq \gamma I, \tag{2.21}$$

where $\gamma$, $T \in \mathbb{R}$ are appropriate positive constants. The PE condition in (2.21) could be interpreted as requirements for a degree of data richness: when the regressor matrix $Y_f$ varies sufficiently enough over the time interval $T$ so that the entire $\gamma$ dimension parameter space is spanned, the estimated parameters are guaranteed to converge to the desired values.

Common methods usually adopt the parameter estimation update law in (2.20) and introduce external noises, e.g. signals in sin or cos form, to satisfy the PE condition shown as (2.21). However, the PE condition in (2.21) is hard to check online whether it is satisfied or not. An online verification condition is desirable to tell practitioners that under this condition, the estimated parameters are guaranteed to converge to the desired values. We observe that the PE condition is in essence a condition about data richness, and only current data contributes to the common parameter estimation update law (2.20). Therefore, to get rich enough data, it is natural to also exploit historical data to construct the parameter estimation update law.

In this section, a parameter estimation update law is proposed by using current and historical data simultaneously. The need of adding external noises to satisfy the PE condition (2.21) is avoided with the benefit of the recorded historical data. Based on the TF-CL method, the parameter estimation update law for the unknown coefficient vector $\theta^*$ is designed as

$$\dot{\hat{\theta}} = -\Gamma k_t Y_f^\top e_f - \sum_{j=1}^{P} \Gamma k_h Y_{f_j}^\top e_{f_j}, \tag{2.22}$$

where $k_t, k_h \in \mathbb{R}^+$ are positive constant gains to trade off the relative importance between current and historical data to the parameter estimation update law. $P \in \mathbb{R}^+$ denotes the volume of the history stacks $\mathcal{H}$ and $\mathcal{E}$. The history stacks $\mathcal{H}$ and $\mathcal{E}$ are collections of historical data, where the filtered regressor matrix $Y_{f_j}$ and the filtered approximation error $e_{f_j}$ denote the $j$-th collected data of the history stacks $\mathcal{H}$ and $\mathcal{E}$, respectively.

The parameter estimation update law (2.22) contains two parts. The first part $-\Gamma k_t Y_f^\top e_f$ relates to current data, which is a common gradient descent update law to minimize the quadratic model approximation error $V_{e_f}$, as like (2.20). However, an update law only with the first part cannot guarantee parameter convergence. Thus, the second part $-\sum_{j=1}^{P} \Gamma k_h Y_{f_j}^\top e_{f_j}$, which is constructed by historical data, is introduced to provide the sufficient excitation required for the parameter convergence. To analyse the parameter convergence problem based on the parameter estimation update law (2.22), a rank condition is firstly clarified in Assumption 2.2.

**Assumption 2.2.** *Given a history stack $\mathcal{H} = [Y_{f_1}^\top, ..., Y_{f_P}^\top] \in \mathbb{R}^{m \times (n \times P)}$, where $Y_{f_j} \in \mathbb{R}^{n \times m}$ is the $j$-th collected data of $\mathcal{H}$, there holds rank$(\mathcal{H}\mathcal{H}^\top) = m$.*

Note that comparing to the traditional PE condition in (2.21), the rank condition of a history stack $\mathcal{H}$ in Assumption 2.2 provides an index about the richness of the historical data that could be checked online. If the rank condition is satisfied, it guarantees that the estimated parameters will converge to the desired values and vice versa. Proofs for the desirable parameter convergence are given as follows.

**Theorem 2.1.** *Given Assumption 2.2 and the parameter estimation update law in* (2.22)*, the parameter estimation error* $\tilde{\theta}$ *converges to zero asymptotically.*

*Proof.* Let $V_{cl} : \mathbb{R}^m \to \mathbb{R}$ be a candidate continuously differential Lyapunov function as

$$V_{cl} = \frac{1}{2}\tilde{\theta}^\top \Gamma^{-1} \tilde{\theta}. \tag{2.23}$$

The bound of the Lyapunov function is

$$\frac{1}{2}\lambda_{\min}(\Gamma^{-1}) \left\|\tilde{\theta}\right\|^2 \leq V_{cl} \leq \frac{1}{2}\lambda_{\max}(\Gamma^{-1}) \left\|\tilde{\theta}\right\|^2. \tag{2.24}$$

Calculating the time derivative of $V_{cl}$ and substituting (2.22) into it yields

$$\begin{aligned}\dot{V}_{cl} &= \tilde{\theta}^\top \Gamma^{-1} \dot{\tilde{\theta}} = \tilde{\theta}^\top \Gamma^{-1} \dot{\hat{\theta}} = -k_t \tilde{\theta}^\top Y_f^\top e_f - \tilde{\theta}^\top \sum_{j=1}^{P} k_h Y_{f_j}^\top e_{f_j} \\ &= -k_t \tilde{\theta}^\top Y_f^\top Y_f \tilde{\theta} - \tilde{\theta}^\top \sum_{j=1}^{P} k_h Y_{f_j}^\top Y_{f_j} \tilde{\theta} \leq -\tilde{\theta}^\top \sum_{j=1}^{P} k_h Y_{f_j}^\top Y_{f_j} \tilde{\theta} = -\tilde{\theta}^\top Q \tilde{\theta},\end{aligned} \tag{2.25}$$

where $Q = \sum_{j=1}^{P} k_h Y_{f_j}^\top Y_{f_j} \in \mathbb{R}^{m \times m}$. According to Assumption 2.2, $Q$ is positive definite and $\lambda_{\min}(Q)$ is a positive constant. Thus, the following inequality holds:

$$\dot{V}_{cl} \leq -\lambda_{\min}(Q) \left\|\tilde{\theta}\right\|^2. \tag{2.26}$$

It is concluded that the parameter estimation error will converge to zero asymptotically. □

### 2.3.2 History Stack Management Algorithm

The parameter estimation update law and the corresponding convergence proof have been provided in Theorem 2.1. The premise of Theorem 2.1 is the satisfaction of the rank condition in Assumption 2.2, i.e., a history stack $\mathcal{H}$ containing sufficiently different data is needed. Besides, according to (2.26) and $Q = \mathcal{H}\mathcal{H}^\top$, the convergence rate of the estimated parameters is related to the minimum eigenvalues of the history stack $\mathcal{H}$, i.e., $\lambda_{\min}(\mathcal{H}\mathcal{H}^\top)$. With the above analysis, we know that the convergence of the estimated parameters to the desired values with a fast speed equals to (a) the satisfaction of the rank condition in Assumption 2.2, and (b) the enlargement of the minimum eigenvalue $\lambda_{\min}(\mathcal{H}\mathcal{H}^\top)$. Thus, to achieve parameter convergence with a fast speed, in our algorithm, the history stack $\mathcal{H}$ and $\mathcal{E}$ are updated with new data points based on two criteria: one is the data threshold $\varepsilon$ that acts as a criterion for data difference, and guides the algorithm to collect different enough data to satisfy the rank condition; the other is the minimum eigenvalue of the history stack $\mathcal{H}$ that relates to the convergence rate of the estimated parameters. Note that for computation simplicity, the minimum singular value $\sigma_{\min}(\mathcal{H}\mathcal{H}^\top)$ replaces with $\lambda_{\min}(\mathcal{H}\mathcal{H}^\top)$ to act as a criterion for data storage given that $\sigma_{\min}(\mathcal{H}\mathcal{H}^\top) = \sqrt{\lambda_{\min}(\mathcal{H}\mathcal{H}^\top)}$.

Details of Algorithm 1 are as follows. Firstly, the hyperparameter data threshold $\varepsilon$ ensures that only new data that is sufficiently different from the latest collected data will be incorporated into the history stacks $\mathcal{H}$ and $\mathcal{E}$. Secondly, to improve the parameter convergence speed, when $\mathcal{H}$ reaches its volume limit $P$, only data points that lead to an increment of the minimum singular values of the history stack $\mathcal{H}$ will be collected. As for the method proposed in [65], the same data might be used multiple times in the history stack $\mathcal{H}$ (data richness deteriorates), and the monotonic increment of the minimum singular values cannot be guaranteed (the convergence rate of the estimated parameter is discouraged). To ensure monotonic increment of the minimum singular values, in our algorithm, the newly coming data always compares with the latest data inserted into the history stack $\mathcal{H}$. Note that the history stack volume $P$ is a hyperparameter that requires careful tuning, which requires $P \geq m$ to satisfy the rank condition in Assumption 2.2, where $m$ is the dimension of the desired coefficient vector $\theta^*$. The pseudocode of the history stack management algorithm is shown as Algorithm 1.

**Algorithm 1** History Stack Management Algorithm

```
Input: Iteration number: $i \geq 1$; Data threshold: $\varepsilon$; Volume: $P$; Auxiliary variables: $T_h$, $T_e$;
    Index: $I = P$; Empty set: $S$; State dimension: $n$.
Output: History stacks $\mathcal{H}$, $\mathcal{E}$.
if $i \leq P$ then
    if $\|Y_f - \mathcal{H}(:, ni - n + 1 : ni)\| / \|Y_f\| \geq \varepsilon$ then
        $\mathcal{H}(:, ni - n + 1 : ni) = Y_f^\top$ in (2.18)
        $\mathcal{E}(:, n) = e_f$ in (2.19)
        $i = i + 1$
    end if
else
    if $\|Y_f - \mathcal{H}(:, nI - n + 1 : nI)\| / \|Y_f\| \geq \varepsilon$ then
        $T_h = \mathcal{H}$; $T_e = \mathcal{E}$; $V = \sigma_{\min}(\mathcal{H}\mathcal{H}^\top)$
        for $l = 1 : P$ do
            $\mathcal{H}(:, nl - n + 1 : nl) = Y_f^\top$ in (2.18)
            $S(l) = \sigma_{\min}(\mathcal{H}\mathcal{H}^\top)$; $\mathcal{H} = T_h$
        end for
       $[V_{\max}, I] = \max(S)$
        if $V_{\max} \geq V$ then
            $\mathcal{H}(:, nI - n + 1 : nI) = Y_f^\top$ in (2.18)
            $\mathcal{E}(:, I) = e_f$ in (2.19)
        else
            $\mathcal{H} = T_h$; $\mathcal{E} = T_e$
        end if
    end if
end if
```

**Remark 2.3.** *In very special cases, it is still possible that the collected historical data from one single trajectory might not be rich enough to satisfy the rank condition in Assumption 2. To counter this potential data deficiency problem, in the initial learning period, a random noise $\Delta$ could be incorporated into the regressor matrix, i.e., $Y_f \leftarrow Y_f + \Delta$, within a short time to enable Algorithm 1 to collect the historical data that the real system does not experience. The random noise $\Delta$ is abandoned once the rank condition in Assumption 2.2 is satisfied.*

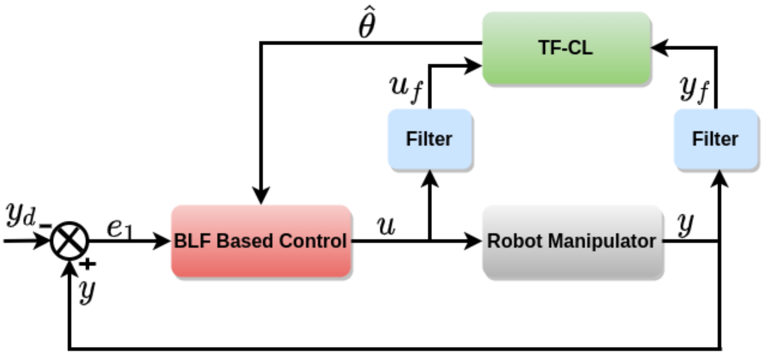


Figure 2.1: Schematic of the proposed method that consists of the TF-CL aided system identification process followed by the BLF based controller design.

## 2.4 Robot Manipulator Controller Design

In this section, based on the identified system from Section 2.3, a recursive controller design process is clarified to yield a stable adaptive control strategy using the backstepping technique and Lyapunov analysis, as shown in Figure 2.1. The resulting control strategy renders the time derivative of the BLF (2.10) to be always negative semi-definite. This guarantees that with a finite initial value of the BLF, the BLF value will always be bounded during the tracking process. The boundness of the BLF implies that the safety set (2.7) and the performance set (2.8) will not be transgressed, i.e., requirements of safety and performance are both satisfied.

Recall the tracking error $e_1 = x_1 - y_d$ in (2.5) and define the error $e_2 = x_2 - \alpha$, where $\alpha \in \mathbb{R}^n$ is a stabilizing function to be designed.

*Step 1.* The following BLF candidate is chosen to design a controller:

$$V_1 = \frac{1}{2}\sum_{i=1}^{n} p(e_{1_i})\frac{e_{1_i}^2}{k_{b_i}^2 - e_{1_i}^2} + (1 - p(e_{1_i}))\frac{e_{1_i}^2}{k_{a_i}^2 - e_{1_i}^2}. \tag{2.27}$$

Taking time derivative of $V_1$ yields

$$\dot{V}_1 = \sum_{i=1}^{n} p(e_{1_i})\frac{k_{b_i}^2 e_{1_i}\dot{e}_{1_i}}{(k_{b_i}^2 - e_{1_i}^2)^2} + (1 - p(e_{1_i}))\frac{k_{a_i}^2 e_{1_i}\dot{e}_{1_i}}{(k_{a_i}^2 - e_{1_i}^2)^2}. \tag{2.28}$$

The time derivative of $e_1$ is

$$\dot{e}_1 = \dot{x}_1 - \dot{y}_d = x_2 - \dot{y}_d = e_2 + \alpha - \dot{y}_d. \tag{2.29}$$

Substituting (2.29) into (2.28) yields

$$\dot{V}_1 = \sum_{i=1}^{n} p(e_{1_i})\frac{k_{b_i}^2 e_{1_i}(e_{2_i} + \alpha_i - \dot{y}_{d_i})}{(k_{b_i}^2 - e_{1_i}^2)^2} + (1 - p(e_{1_i}))\frac{k_{a_i}^2 e_{1_i}(e_{2_i} + \alpha_i - \dot{y}_{d_i})}{(k_{a_i}^2 - e_{1_i}^2)^2}, \tag{2.30}$$

where $\alpha_i$ and $\dot{y}_{d_i}$ are $i$-th dimension of $\alpha$ and $\dot{y}_d$, respectively. In order to make (2.30) be negative semi-definite, the stabilizing function $\alpha$ is designed as

$$\alpha = \dot{y}_d - p(e_1)(k_b^\top k_b - e_1^\top e_1)^2 k_1 e_1 - (1 - p(e_1))(k_a^\top k_a - e_1^\top e_1)^2 k_1 e_1, \tag{2.31}$$

where $k_1 \in \mathbb{R}^{n\times n}$ is a diagonal matrix of positive constants, and its $i$-th diagonal entry is denoted as $k_{1i}$. Since asymmetric constraints are considered in this chapter, the last two terms of (2.31) are designed to characterize the upper and lower constraint boundaries, i.e., $k_b$ and $-k_a$, respectively.

For simplicity, we denote $L = \sum_{i=1}^{n} p(e_{1_i})\frac{k_{b_i}^2 e_{1_i} e_{2_i}}{(k_{b_i}^2 - e_{1_i}^2)^2} + (1-p(e_{1_i}))\frac{k_{a_i}^2 e_{1_i} e_{2_i}}{(k_{a_i}^2 - e_{1_i}^2)^2}$. Then, combining with (2.31), we rewrite (2.30) as

$$\begin{aligned}\dot{V}_1 &= -\sum_{i=1}^{n} p(e_{1_i}) k_{1_i} k_{b_i}^2 e_{1_i}^2 + (1-p(e_{1_i}))k_{1_i} k_{a_i}^2 e_{1_i}^2 + L \\ &= -e_1^\top k_1[p(e)k_b^\top k_b + (1-p(e))k_a^\top k_a]e_1 + L = -e_1^\top K_1 e_1 + L,\end{aligned} \tag{2.32}$$

where $K_1 = k_1[p(e)k_b^\top k_b + (1-p(e))k_a^\top k_a] \in \mathbb{R}^{n\times n}$ is a positive definite matrix.

*Step 2.* We define

$$V_2 = \frac{1}{2} e_2^\top M(x_1) e_2, \tag{2.33}$$

and choose

$$V_{blf} = V_1 + V_2. \tag{2.34}$$

The time derivative of $V_{blf}$ is

$$\dot{V}_{blf} = \dot{V}_1 + \dot{V}_2 = \dot{V}_1 + e_2^\top M(x_1)\dot{e}_2 + \frac{1}{2} e_2^\top \dot{M}(x_1) e_2. \tag{2.35}$$

Combining with (2.3), the time derivative of $e_2$ follows

$$\dot{e}_2 = \dot{x}_2 - \dot{\alpha} = M^{-1}(x_1)(\tau - C(x_1,x_2)x_2 - G(x_1) - Fx_2) - \dot{\alpha}. \tag{2.36}$$

Invoking (2.32), (2.35), and (2.36) yields

$$\dot{V}_{blf} = -e_1^\top K_1 e_1 + L + e_2^\top[\tau - C(x_1,x_2)x_2 - G(x_1) - Fx_2 - M(x_1)\dot{\alpha} + \frac{1}{2}\dot{M}(x_1)e_2]. \tag{2.37}$$

where the stabilizing function $\alpha$ is defined in (2.31).

If an accurate model is available, a stabilizing control law could be directly designed as

$$\tau = M(x_1)\dot{\alpha} + C(x_1,x_2)x_2 + G(x_1) + Fx_2 - k_2 e_2 - (e_2^\top)^\dagger L - \frac{1}{2}\dot{M}(x_1)e_2, \tag{2.38}$$

where $k_2 \in \mathbb{R}^{n\times n}$ is a matrix of positive constants to be designed; $(e_2^\top)^\dagger L$ is a stabilizing term, wherein † stands for the Moore-Penrose inverse.

Accurate model information is required in (2.38) to design a stabilizing control law. However, but it is unavailable in our problem. To provide an approximation of the unknown model information existing in the right side of (2.38), comparing the difference between (2.2) and (2.38), a double regressor matrices technique for the TF-CL is introduced here. In particular, like the regressor matrix $Y(q,\dot{q},\ddot{q})$ that is proposed for approximation of the system (2.2), a new regressor matrix $X(x_1,x_2,\alpha,\dot{\alpha})$ is formulated to approximate the unknown model in (2.38). Based on the newly designed $X(x_1,x_2,\alpha,\dot{\alpha})$, the following approximation equation establishes:

$$X(x_1,x_2,\alpha,\dot{\alpha})\theta^* = M(x_1)\dot{\alpha} + C(x_1,x_2)x_2 + G(x_1) + Fx_2 - \frac{1}{2}\dot{M}(x_1)e_2, \tag{2.39}$$

where $X(x_1, x_2, \alpha, \dot{\alpha}) \in \mathbb{R}^{n\times m}$ is a regressor matrix constructed based on the information of $x_1$, $x_2$, $\alpha$ and $\dot{\alpha}$. We defer a detailed discussion of the relationship between regressor matrices $X(x_1, x_2, \alpha, \dot{\alpha})$ and $Y(q, \dot{q}, \ddot{q})$ in Remark 2.4, and focus now on the design of the parameter estimation update law for the BLF based controller with help of the new regressor matrix $X(x_1, x_2, \alpha, \dot{\alpha})$.

Based on (2.39), the model based control law (2.38) is reformulated as

$$\tau = X(x_1, x_2, \alpha, \dot{\alpha})\theta^* - k_2 e_2 - (e_2^\top)^\dagger L. \tag{2.40}$$

Since $\theta^*$ is unknown, and only $\hat{\theta}$ is available, based on the double regressor matrices technique, a TF-CL aided parameter estimation update law for the BLF based controller (2.40) is designed as

$$\dot{\hat{\theta}} = -\Gamma X^\top e_2 - \Gamma k_t Y_f^\top e_f - \sum_{j=1}^{P} \Gamma k_h Y_{f_j}^\top e_{f_j}. \tag{2.41}$$

Comparing the difference between (2.22) and (2.41), the first term of (2.41) is designed as a stabilizing term, which serves to provide the stability proof in Theorem 2. By adjusting the values of $\Gamma$, $k_t$, and $k_h$, the importance of each part to the parameter estimation update law is traded off. Finally, based on the estimated parameter vector $\hat{\theta}$ from the TF-CL method, the stabilizing control law (2.40) is rewritten as

$$\tau = X(x_1, x_2, \alpha, \dot{\alpha})\hat{\theta} - k_2 e_2 - (e_2^\top)^\dagger L. \tag{2.42}$$

**Remark 2.4.** *Observing* (2.2) *and* (2.39)*, we find that these two equations share the same coefficient vector $\theta^*$ but with different regressor matrices. The double regressor matrices technique illustrated here makes a combination of the TF-CL method and the BLF based control strategy feasible. $Y(q, \dot{q}, \ddot{q})$ is a regressor matrix fully depends on the model structure. $X(x_1, x_2, \alpha, \dot{\alpha})$ is a regressor matrix constructed based on both model properties and the stabilizing function $\alpha$.*

In the reaming part of this section, the main conclusions of this chapter and the corresponding proofs are given based on the parameter estimation update law (2.41) and the stabilizing control strategy (2.42).

**Theorem 2.2.** *Consider an n-link robot manipulator in* (2.1)*, the parameter estimation update law* (2.41) *and the control policy* (2.42)*. Given Assumptions 2.1-2.2, for initial values of the system output and the tracking error lying in the predefined safety set* (2.7) *and performance set* (2.8)*, the following properties hold:*

*(i) The tracking error $e_1$, the error $e_2$, and the parameter estimation error $\tilde{\theta}$ are stable and converge to zero asymptotically.*

*(ii) The tracking error $e_1$ is bounded by $\Omega_{e_1}$ where*

$$\Omega_{e_1} = \left\{ e_1 \in \mathbb{R}^n : -\underline{U}_{e_1} \leq e_1 \leq \overline{U}_{e_1} \right\} \in \mathcal{P}, \tag{2.43}$$

*where $\underline{U}_{e_1} = [\underline{U}_{e_{1_i}}, ..., \underline{U}_{e_{1n}}]^\top \in \mathbb{R}^n$, $\underline{U}_{e_{1_i}} = k_{a_i}\sqrt{\frac{2V(0)}{1+2V(0)}}$; $\overline{U}_{e_1} = [\overline{U}_{e_{1_i}}, ..., \overline{U}_{e_{1n}}]^\top \in \mathbb{R}^n$, $\overline{U}_{e_{1_i}} = k_{b_i}\sqrt{\frac{2V(0)}{1+2V(0)}}$; $V(0)$ is the value of the BLF at $t = 0$.*

*The error $e_2$ remains in the compact set $\Omega_{e_2}$*

$$\Omega_{e_2} = \left\{ e_2 \in \mathbb{R}^n : \|e_2\| \leq \sqrt{\frac{2V(0)}{\lambda_{\min}(M)}} \right\}. \tag{2.44}$$

*(iii) For all $t > 0$, there holds $y(t) \in \Omega_y$, where*

$$\Omega_y = \left\{ y \in \mathbb{R}^n : -\underline{U}_{e_1} - \underline{y}_d \prec y \prec \overline{U}_{e_1} + \overline{y}_d \right\} \in \bar{\mathcal{S}}. \tag{2.45}$$

*Proof. Proof of (i):* For the stability proof, let $Z = [e_1, e_2, \tilde{\theta}]^\top \in \mathbb{R}^{2n+m}$ and consider the following Lyapunov function

$$V(Z) = V_{blf} + V_{cl}. \tag{2.46}$$

Combining with (2.25) and (2.37), the time derivative of (2.46) yields

$$\begin{aligned} \dot{V}(Z) &= \dot{V}_{blf} + \dot{V}_{cl} \\ &= -e_1^\top K_1 e_1 + L + e_2^\top [\tau - C(x_1, x_2)x_2 - G(x_1) \\ &\quad - Fx_2 - M(x_1)\dot{\alpha} + \frac{1}{2}\dot{M}(x_1)e_2] + \tilde{\theta}^\top \Gamma^{-1} \dot{\tilde{\theta}} \end{aligned} \tag{2.47}$$

Substituting (2.39), (2.41) and (2.42) into (2.47) reads

$$\begin{aligned} \dot{V}(Z) &= -e_1^\top K_1 e_1 + L + e_2^\top [X\hat{\theta} - k_2 e_2 - (e_2^\top)^\dagger L - X\theta^*] + \tilde{\theta}^\top \Gamma^{-1} \dot{\tilde{\theta}} \\ &= -e_1^\top K_1 e_1 - e_2^\top k_2 e_2 + e_2^\top X \tilde{\theta} + \tilde{\theta}^\top \Gamma^{-1} [-\Gamma X^\top e_2 - \Gamma k_t Y_f^\top e_f - \sum_{j=1}^{P} \Gamma k_h Y_{f_j}^\top e_{f_j}] \\ &= -e_1^\top K_1 e_1 - e_2^\top k_2 e_2 - k_t \tilde{\theta}^\top Y_f^\top Y_f \tilde{\theta} - \tilde{\theta}^\top \sum_{j=1}^{P} k_h Y_{f_j}^\top (Y_{f_j} \hat{\theta}_j - \tau_{f_j}) \\ &= -e_1^\top K_1 e_1 - e_2^\top k_2 e_2 - k_t \tilde{\theta}^\top Y_f^\top Y_f \tilde{\theta} - \tilde{\theta}^\top \sum_{j=1}^{P} k_h Y_{f_j}^\top (Y_{f_j} \hat{\theta}_j - Y_{f_j} \theta^*) \\ &\leq -e_1^\top K_1 e_1 - e_2^\top k_2 e_2 - \tilde{\theta}^\top \sum_{j=1}^{P} k_h Y_{f_j}^\top Y_{f_j} \tilde{\theta}. \end{aligned} \tag{2.48}$$

Let $N = \operatorname{diag}(K_1, k_2, Q) \in \mathbb{R}^{(2n+m)\times(2n+m)}$, wherein $Q = \sum_{j=1}^{P} k_h Y_{f_j}^\top Y_{f_j} \in \mathbb{R}^{m\times m}$, (2.48) could be rewritten as

$$\dot{V}(Z) \leq -Z^\top N Z \leq -\lambda_{\min}(N) \|Z\|^2, \tag{2.49}$$

where $\lambda_{\min}(N) = \min(\lambda_{\min}(K_1), \lambda_{\min}(k_2), \lambda_{\min}(Q))$. Finally, it is concluded that the tracking error $e_1$, the error $e_2$, and the parameter estimation error $\tilde{\theta}$ converge to zero asymptotically.

*Proof of (ii):* Since $V(Z)$ is positive definite and $\dot{V}(Z) < 0$ according to (2.49), $V(Z) \leq V(Z(0))$ establishes. From $V(Z) = V_1(e_1) + V_2(e_2) + V_{cl}(\tilde{\theta})$ and the fact that $V_2(e_2)$ and $V_{cl}(\tilde{\theta})$ are positive functions, it is concluded that $V_1(e_1) < V(Z(0))$, i.e., $V_1(e_1)$ is bounded. According to the characteristics of the BLF (2.27), when $e_1 \to -k_a$ or $e_1 \to k_b$, we get $V_1(e_1) \to \infty$. Thus, the boundness of $V_1(e_1)$ implies that $e_1 \neq -k_a$ or $e_1 \neq k_b$. Given that $-k_a \prec e_1(0) \prec k_b$, it is concluded that $-k_a \prec e_1(t) \prec k_b, \forall t > 0$. This means that the tracking error always lies in the required performance set (2.8). Besides, from the analysis mentioned above, we know that $V_1(e_1) < V(0)$. To get the bound of $e_1$, firstly we take the $i$-th element of $e_1$ as an example. For $e_{1_i}$, the following inequalities establish

$$V(0) > \begin{cases} \frac{e_{1_i}^2}{2(k_{b_i}^2 - e_{1_i}^2)} & 0 < e_{1_i} < k_{b_i} \\ \frac{e_{1_i}^2}{2(k_{a_i}^2 - e_{1_i}^2)} & -k_{a_i} < e_{1_i} < 0 \end{cases}. \tag{2.50}$$

We represent the above (2.50) as the following equivalent form

$$e_{1_i}^2 < \begin{cases} k_{b_i}^2 \frac{2V(0)}{1+2V(0)} & 0 < e_{1_i} < k_{b_i} \\ k_{a_i}^2 \frac{2V(0)}{1+2V(0)} & -k_{a_i} < e_{1_i} < 0 \end{cases}. \tag{2.51}$$

From above it is concluded that for $e_{1_i} > 0$, $e_{1_i} < k_{b_i}\sqrt{\frac{2V(0)}{1+2V(0)}}$ holds, and $e_{1_i} > -k_{a_i}\sqrt{\frac{2V(0)}{1+2V(0)}}$ establishes when $e_{1_i} < 0$. Furthermore, since $\sqrt{\frac{2V(0)}{1+2V(0)}} < 1$, $-k_{a_i} < -k_{a_i}\sqrt{\frac{2V(0)}{1+2V(0)}} < e_{1_i} < k_{b_i}\sqrt{\frac{2V(0)}{1+2V(0)}} < k_{b_i}$ establishes. Consider all elements of $e_1$ and the performance set $\mathcal{P}$ in (2.8), (2.43) establishes.

Consider the case of $e_2$, since $V_2(e_2) = \frac{1}{2}e_2^\top M e_2 < V(0)$, $\|e_2\| \leq \sqrt{\frac{2V(0)}{\lambda_{\min}(M)}}$ establishes, i.e., $e_2$ remains in the set $\Omega_{e_2}$.

*Proof of (iii):* The output follows $y = x_1 = e_1 + y_d$. According to (2.43), $-\underline{U}_{e_1} \leq e_1 \leq \overline{U}_{e_1}$ establishes. We know that $-\underline{y}_d \leq y_d \leq \overline{y}_d$ from Assumption 2.1. Thus, it is easy to get that $-\underline{U}_{e_1} - \underline{y}_d \leq y \leq \overline{U}_{e_1} + \overline{y}_d$. Since $\underline{U}_{e_1} \prec k_a$ and $\overline{U}_{e_1} \prec k_b$, $-k_a - \underline{y}_d \prec -\underline{U}_{e_1} - \underline{y}_d \prec 0$ and $0 \prec \overline{U}_{e_1} + \overline{y}_d \prec k_b + \overline{y}_d$ establishes, i.e., $\Omega_y \in \bar{\mathcal{P}}$. Since $-k_a$ and $k_b$ are chosen to satisfy $\bar{\mathcal{P}} \subseteq \bar{\mathcal{S}}$, $\Omega_y \in \bar{\mathcal{S}}$ also establishes, i.e., system outputs will not transgress the predefined safety set (2.7). □

## 2.5 Numerical Simulation

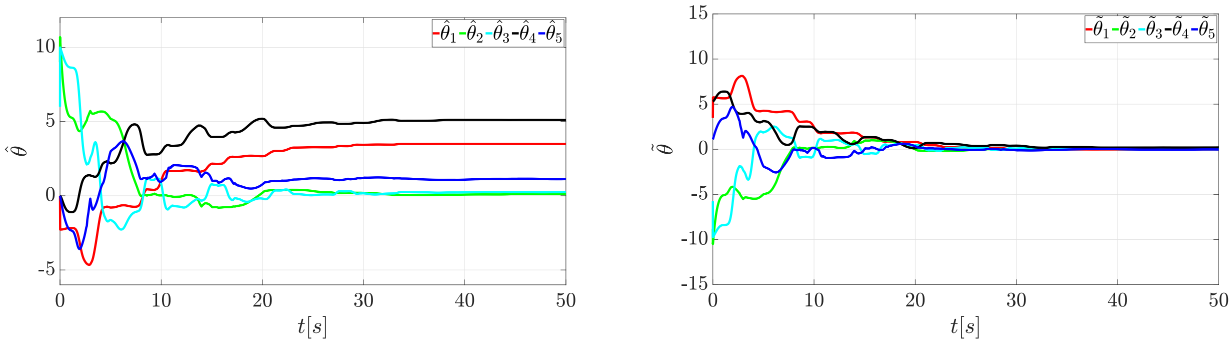
This section implements numerical simulations based on a 2-DoF robot manipulator [20], [66] to show the effectiveness of the parameter estimation update law (2.41) and the BLF based control strategy (2.42). The utilized 2-DoF robot manipulator serves as a benchmark to test whether the TF-CL method could ensure the estimated parameters converge to the actual values.

### 2.5.1 Static Coefficient Vector

The white-box model of the 2-DoF robot manipulator follows

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + F\dot{q} = \tau,$$

where $q = [q_1, q_2]^\top \in \mathbb{R}^2$, $\dot{q} = [\dot{q}_1, \dot{q}_2]^\top \in \mathbb{R}^2$, $M(q) = \begin{bmatrix} p_1 + 2p_3c_2 & p_2 + p_3c_2 \\ p_2 + p_3c_2 & p_2 \end{bmatrix} \in \mathbb{R}^{2\times 2}$, $C(q,\dot{q}) = \begin{bmatrix} -p_3\dot{q}_2 s_2 & -p_3(\dot{q}_1 + \dot{q}_2)s_2 \\ p_3\dot{q}_1 s_2 & 0 \end{bmatrix} \in \mathbb{R}^{2\times 2}$, $F = \begin{bmatrix} f_1 & 0 \\ 0 & f_2 \end{bmatrix} \in \mathbb{R}^{2\times 2}$, $c_2 = \cos q_2$, and $s_2 =$



(a) Trajectory of the estimated parameter $\hat{\theta}$.

(b) Trajectory of the estimation errors $\tilde{\theta}$.

Figure 2.2: The trajectories of $\hat{\theta}$ and $\tilde{\theta}$ using the parameter estimation update law (2.41).

$\sin q_2$. The regressor matrices $Y_1(q,\dot{q}) \in \mathbb{R}^{2\times 5}$, $Y_2(q,\dot{q}) \in \mathbb{R}^{2\times 5}$, and the coefficient vector $\theta^* \in \mathbb{R}^5$ for the TF-CL method are given as

$$Y_1(q,\dot{q}) = \begin{bmatrix} \dot{q}_1 & \dot{q}_2 & 2\dot{q}_1 c_2 + \dot{q}_2 c_2 & 0 & 0 \\ 0 & \dot{q}_1 + \dot{q}_2 & \dot{q}_1 c_2 & 0 & 0 \end{bmatrix},$$

$$Y_2(q,\dot{q}) = \begin{bmatrix} 0 & 0 & 0 & \dot{q}_1 & 0 \\ 0 & 0 & \dot{q}_1\dot{q}_2 s_2 + \dot{q}_1^2 s_2 & 0 & \dot{q}_2 \end{bmatrix},$$

$$\theta^* = \begin{bmatrix} p_1 & p_2 & p_3 & f_1 & f_2 \end{bmatrix}^\top.$$

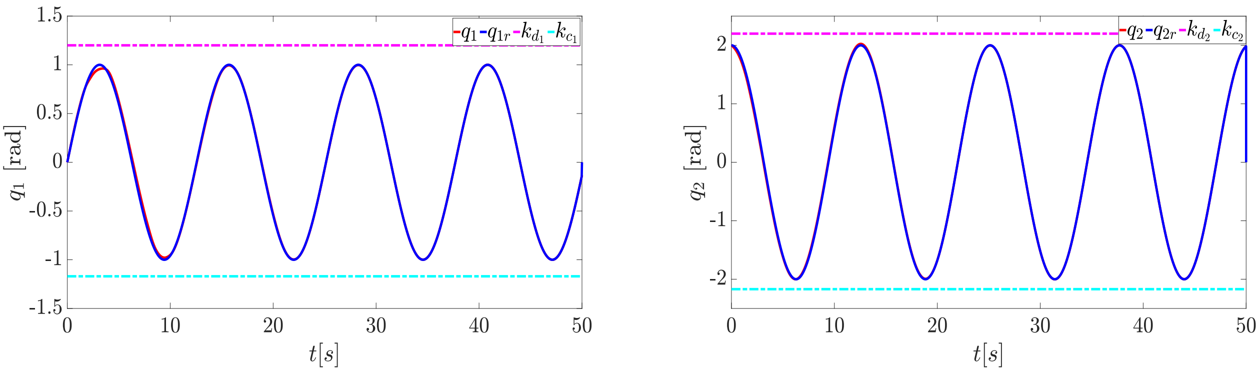
According to [66], the desired values of $\theta^*$ are set as $p_1 = 3.473$, $p_2 = 0.196$, $p_3 = 0.242$, $f_1 = 5.3$, and $f_2 = 1.1$. The regressor matrix $X(q,\dot{q},\alpha,\dot{\alpha}) \in \mathbb{R}^{2\times 5}$ used in (2.42) follows

$$X(q,\dot{q},\alpha,\dot{\alpha}) = \begin{bmatrix} \dot{\alpha}_1 & 0 & X_1 & \dot{q}_1 & 0 \\ 0 & \dot{\alpha}_1 + \alpha_2 & X_2 & 0 & \dot{q}_2 \end{bmatrix},$$

where $\alpha = [\alpha_1, \alpha_2]^\top \in \mathbb{R}^2$, $X_1 = 2\dot{\alpha}_1 c_2 + \dot{\alpha}_2 c_2 - 1/2\dot{q}_2^2 s_2 - \dot{q}_1\dot{q}_2 s_2 + (1/2\alpha_2 - \alpha_1)\dot{q}_2 s_2$, $X_2 = \dot{\alpha}_1 c_2 + \dot{q}_1^2 s_2 + 1/2\dot{q}_1\dot{q}_2 s_2 - 1/2\dot{\alpha}_1$. The hyperparameters for the TF-CL part are set as $\Gamma = \mathrm{diag}(I_{5\times 1})$, $k_t = 200$, $k_h = 0.001$, $P = 7$, and $\varepsilon = 0.1$. The time constant of the filter is set as $k = 0.001$. Initial values of the estimated parameters are set as $\hat{\theta}(0) = [0, 9, 6, 0, 0]^\top$.

As for the BLF based control strategy (2.42), the parameters are set as $k_1 = \mathrm{diag}(40, 30)$, $k_2 = \mathrm{diag}(30, 40)$. The desired trajectory is chosen as $y_d = [\sin 0.5t, 2\cos 0.5t]^\top$. The required safety and performance issues for two joints are set as follows. For joint 1, the safety set is chosen as $\bar{\mathcal{S}}_1 = \{-1.17 < q_1 < 1.2\}$, the performance set follows $\mathcal{P}_1 = \{-0.17 < e_{1_1} < 0.2\}$; For joint 2, the safety set is $\bar{\mathcal{S}}_2 = \{-2.17 < q_2 < 2.2\}$, the performance set is chosen as $\mathcal{P}_2 = \{-0.17 < e_{1_2} < 0.2\}$. To ensure that the 2-DoF robot manipulator tracks the desired trajectory while satisfying the safety and performance criteria illustrated above, we set $k_c = [-1.17, -2.17]^\top$, $k_d = [1.2, 2.2]^\top$, $k_a = [0.17, 0.17]^\top$, and $k_b = [0.2, 0.2]^\top$. In order to ensure initial values lie in the corresponding safety and performance sets, we choose $x_1(0) = [0, 2]^\top$, $x_2(0) = [0.5, 0]^\top$.

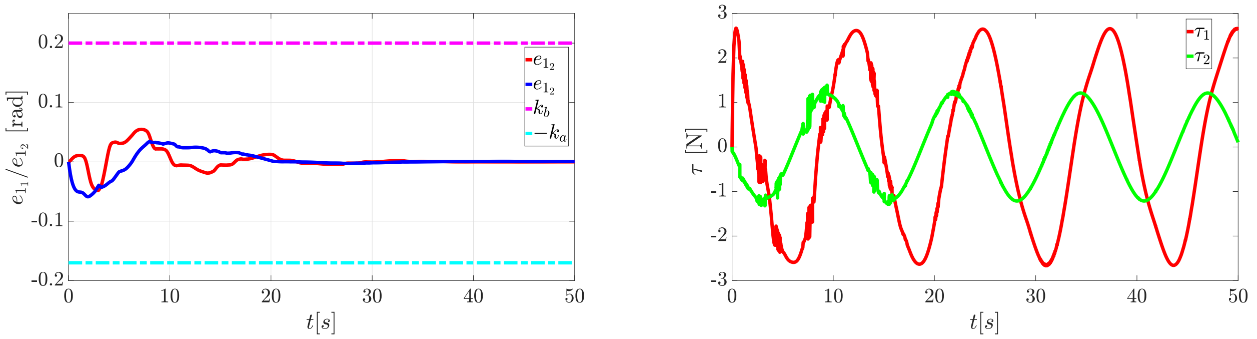
The parameter estimation update law (2.41) is adopted for online estimation of the unknown coefficient vector $\theta^*$. In Figure 2.2a, the estimated parameters converge to their actual



(a) Trajectories of $q_1$ and $q_{1r}$.

(b) Trajectories of $q_2$ and $q_{2r}$.

Figure 2.3: The trajectories of joint angles $q_1$, $q_2$, reference trajectories $q_{1r}$, $q_{2r}$, and associated safety bounds $k_{d_i}$, $k_{c_i}$, $i = 1, 2$ under the proposed control strategy (2.42).



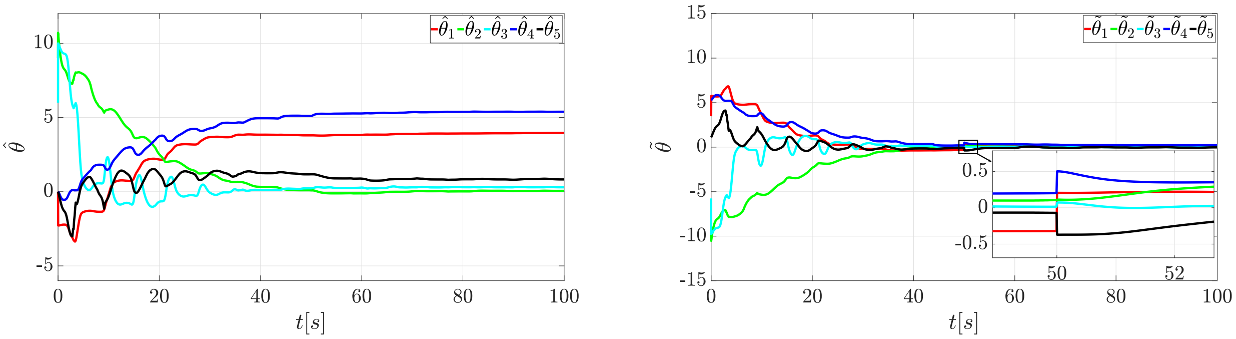
(a) Trajectories of $e_{1_1}$, $e_{1_2}$, and $k_b$,$-k_a$.

(b) Trajectories of the control $\tau$.

Figure 2.4: The trajectories of the joint 1 tracking error $e_{1_1}$, joint 2 tracking error $e_{1_2}$, the associated performance bounds $k_b$,$-k_a$, and the control input $\tau$.

values without incorporating external noises to satisfy the PE condition. The corresponding parameter estimation errors are shown in Figure 2.2b where they finally converge to a small neighbourhood around zero. This means that an estimated model with high quality is gotten. However, even though a good tracking performance achieves in [67], the norm of the estimated weights does not converge.

The proposed control law (2.42) is applied to the 2-DoF robot manipulator. It is displayed in Figure 2.3 that the trajectories of $q_1$ and $q_2$ follow their desired trajectories $q_{1r}$ and $q_{2r}$ precisely. The safety set $\bar{\mathcal{S}}_1$ (the upper bound $k_{d_1}$ and the lower bound $k_{c_1}$) for $q_1$, and the safety set $\bar{\mathcal{S}}_2$ (the upper bound $k_{d_2}$ and the lower bound $k_{c_2}$) for $q_2$ are never be violated during the operation process. The tracking errors of two joints displayed in Figure 2.4a finally converge to zero and always lie in the required performance set $\mathcal{P}_1$ and $\mathcal{P}_2$ respectively. Note that $\mathcal{P}_1$ and $\mathcal{P}_2$ share the same upper bound $k_b$ and lower bound $-k_a$. The associated control trajectory is shown in Figure 2.4b where $\tau$ oscillates when the estimated parameter vector is in the converging process. From the above analysis, it is concluded that the proposed parameter estimation update law (2.41) guarantees that the estimated parameters converge to their desired values. The control strategy given in (2.42) drives the 2-DoF robot manipulator to track the reference trajectory precisely and satisfy



(a) Trajectory of the estimated parameter $\hat{\theta}$.

(b) Trajectory of the estimation error $\tilde{\theta}$.

Figure 2.5: The trajectories of $\hat{\theta}$ and $\tilde{\theta}$ where an disturbance is incorporated at $t = 50\ s$.

the requirements of safety and performance together.

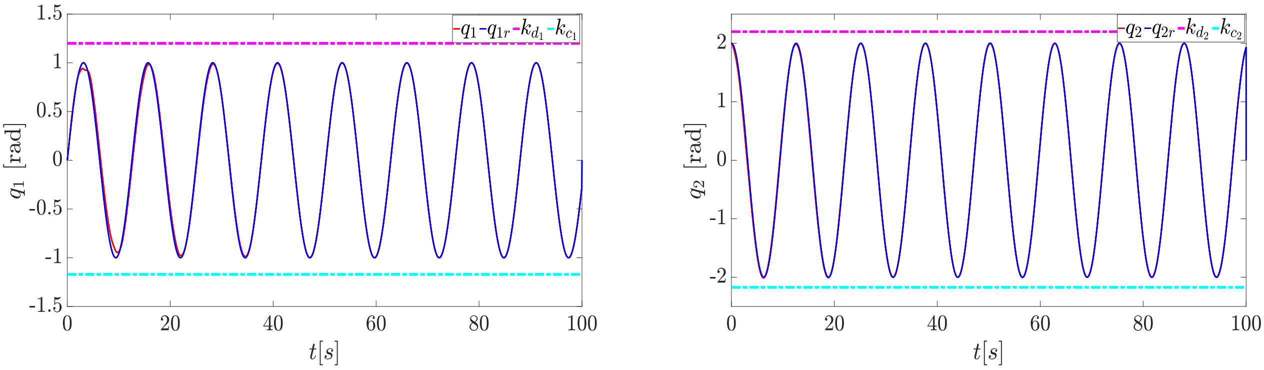
### 2.5.2 Time-Varying Coefficient Vector

This subsection randomly reset the initial coefficient vector $\theta^* = [3.473, 0.196, 0.242, 5.3, 1.1]^\top$ as a new desired parameter vector $\theta_d^* = [4, 0.2, 0.3, 5.6, 0.8]^\top$ at time $t = 50\ s$ to simulate time-varying uncertainties in terms of unknown payloads or friction parameters. In practice, only parts of the coefficient vector will change due to environmental effects on the mass, length or friction parameters. Here a hard disturbance is deliberately chosen to exemplify the effectiveness of the parameter estimation update law (2.41) to counter time-varying uncertainties.

As displayed in Figure 2.5a, the estimated parameters converge to the initial desired values of $\theta^*$ in the first 50 seconds. When an additional disturbance is added at $t = 50\ s$, the TF-CL method collects new data and enables the estimated parameters to finally converge to the new desired values of $\theta_d^*$. As shown in Figure 2.5b, the trajectories of the parameter estimation errors abruptly change at $t = 50\ s$ when an additional disturbance is added. Then, the parameter estimation errors still converge to a small neighbourhood around zero. Figure 2.6a and Figure 2.6b demonstrate that under the proposed control strategy (2.42), two joints track their reference trajectories precisely and will not violate their corresponding predefined safety sets $\bar{\mathcal{S}}_1$ and $\bar{\mathcal{S}}_2$ even when an additional disturbance is added. Besides, it is observed in Figure 2.7a that the tracking errors $e_{1_1}$ and $e_{1_2}$ oscillate when the disturbance is added at time $t = 50\ s$. Then, they finally converge to zero. The tracking errors always lie in the given performance set defined by the lower bound $-k_a$ and the upper bound $k_b$. The control trajectories given in Figure 2.7b provide additional information about the influence of the time-varying uncertainties on the control strategy. When the disturbance is added at $t = 50\ s$, the magnitude of the control $\tau_1$ increases, and the magnitude of the control $\tau_2$ decreases to drive the robot to track the desired trajectory.

## 2.6 Experimental Validation

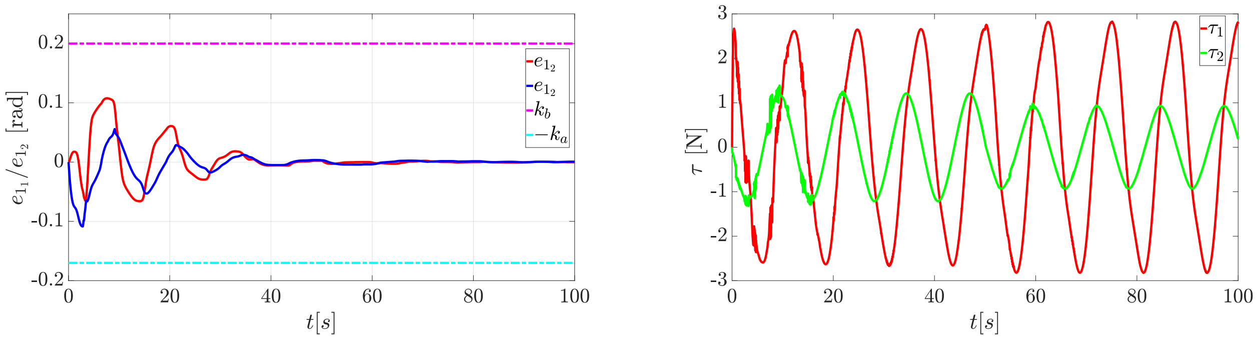
This section experimentally validates the effectiveness of the parameter estimation update law (2.41) and the proposed control strategy (2.42) using the 3-DoF robot manipulator



(a) Trajectories of $q_1$ and reference $q_{1r}$, and the associated safety bounds $k_{d_1}$,$k_{c_1}$.

(b) Trajectories of $q_2$ and reference $q_{2r}$, the associated safety bounds $k_{d_2}$,$k_{c_2}$.

Figure 2.6: The trajectories of joint angles $q_1$, $q_2$ and reference trajectories $q_{1r}$, $q_{2r}$ where an additional disturbance is incorporated at $t = 50$ $s$.



(a) Trajectories of $e_{1_1}$, $e_{1_2}$, and $k_b$,$-k_a$.

(b) Trajectories of $\tau$.

Figure 2.7: The trajectories of the joint 1 tracking error $e_{1_1}$, joint 2 tracking error $e_{1_2}$, the associated performance bounds $k_b$,$-k_a$, and the control input $\tau$ where a disturbance is incorporated at $t = 50$ $s$.

illustrated in Appendix A.1.

According to the detailed model structure provided in Appendix A.1, the unknown coefficient vector $\theta^* \in \mathbb{R}^{11}$ reads

$$\theta^* = [p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10}, f]^\top.$$

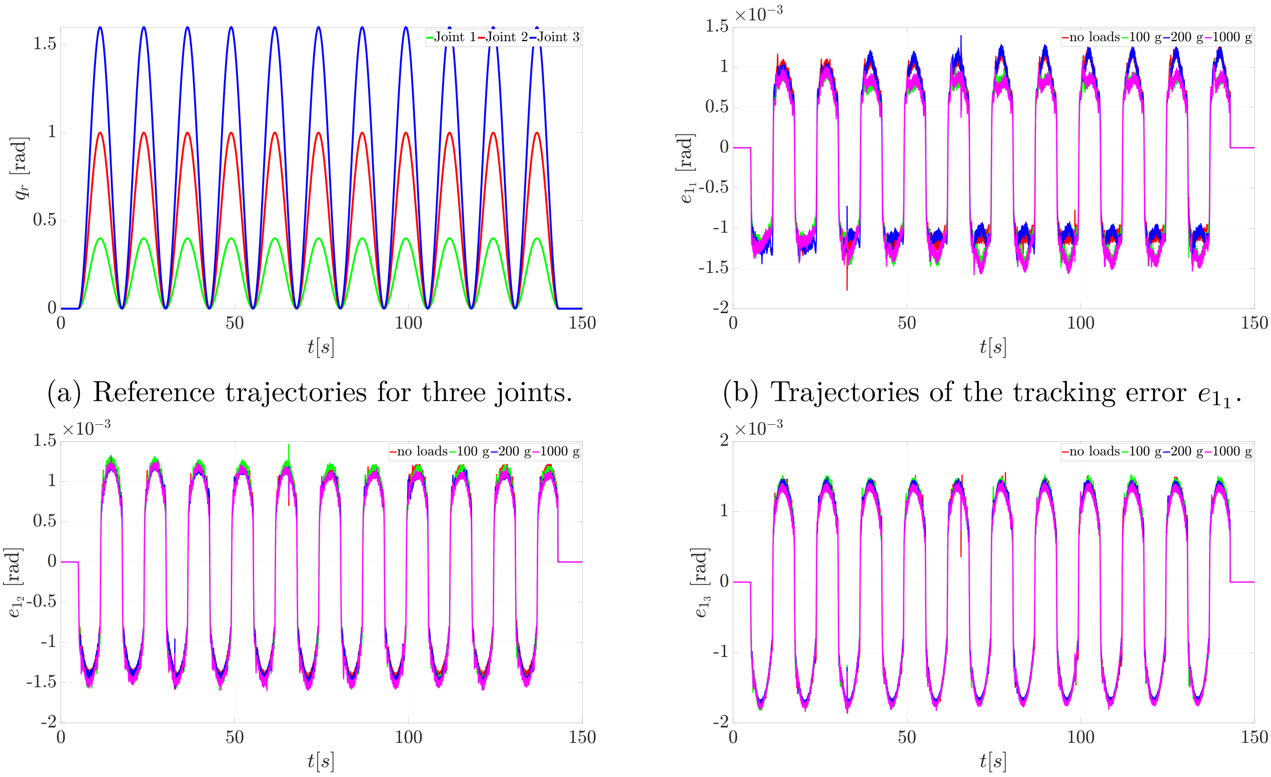
The corresponding regressor matrices $Y_1(q, \dot{q})$, $Y_2(q, \dot{q}) \in \mathbb{R}^{3\times 11}$ are given as

$$Y_1(q, \dot{q}) = [Y_{11}, Y_{12}, Y_{13}]^\top,$$

where $Y_{11}$, $Y_{12}$, $Y_{13} \in \mathbb{R}^{1\times 11}$ are in the following forms,

$$Y_{11} = [\dot{q}_1, c_{23}\dot{q}_1, c_2\dot{q}_1, c_3\dot{q}_1 + c_3\dot{q}_2, \dot{q}_2, c_{23}\dot{q}_2 + c_{23}\dot{q}_3, c_2\dot{q}_2, \dot{q}_3, c_3\dot{q}_3, 0, 0],$$
$$Y_{12} = [0, 0, 0, c_3\dot{q}_1, \dot{q}_1 + \dot{q}_2, c_{23}\dot{q}_1, c_2\dot{q}_1, \dot{q}_3, c_3\dot{q}_3, c_3\dot{q}_2, 0],$$
$$Y_{13} = [0, 0, 0, 0, 0, c_{23}\dot{q}_1, 0, \dot{q}_1 + \dot{q}_2 + \dot{q}_3, c_3\dot{q}_1 + c_3\dot{q}_2, 0, 0].$$

$$Y_2(q, \dot{q}) = [Y_{21}, Y_{22}, Y_{23}]^\top,$$



(a) Reference trajectories for three joints.

(b) Trajectories of the tracking error $e_{1_1}$.

(c) Trajectories of the tracking error $e_{1_2}$.

(d) Trajectories of the tracking error $e_{1_3}$.

Figure 2.8: The trajectories of the tracking error $e$ for three joints under different payloads.

where $Y_{21},\ Y_{22},\ Y_{23} \in \mathbb{R}^{1\times 11}$ follows,

$Y_{21} = [0, s_{23}(\dot{q}_2+\dot{q}_3)\dot{q}_1 - s_{23}\dot{q}_1\dot{q}_2 - s_{23}\dot{q}_1\dot{q}_3 - s_{23}\dot{q}_2\dot{q}_3, 0, 0, 0, s_{23}(\dot{q}_2+\dot{q}_3)\dot{q}_2 + s_{23}(\dot{q}_2+\dot{q}_3)\dot{q}_3 - s_{23}\dot{q}_2^2 - s_{23}\dot{q}_3^2, 0, 0, 0, 0, \dot{q}_1]$,

$Y_{22} = [0, 0, 0, -s_3\dot{q}_2\dot{q}_3, 0, s_{23}(\dot{q}_2+\dot{q}_3)\dot{q}_1 + s_{23}\dot{q}_1^2, s_2\dot{q}_1\dot{q}_2 + s_2\dot{q}_1^2, 0, 0, s_3\dot{q}_2\dot{q}_3, \dot{q}_2]$,

$Y_{23} = [0, 0, 0, s_3\dot{q}_1\dot{q}_2, 0, s_{23}(\dot{q}_2+\dot{q}_3)\dot{q}_1 + s_{23}\dot{q}_1^2, 0, 0, s_3\dot{q}_1\dot{q}_3 + s_3\dot{q}_2\dot{q}_3 + s_3\dot{q}_2^2 + s_3\dot{q}_1^2, 0, \dot{q}_3]$.

The explicit form of the regressor matrix $X(q, \dot{q}, \alpha, \dot{\alpha}) \in \mathbb{R}^{3\times 11}$ used in (2.42) follows

$$X(q, \dot{q}, \alpha, \dot{\alpha}) = [X_1, X_2, X_3]^\top,$$

where $X_1,\ X_2,\ X_3 \in \mathbb{R}^{1\times 11}$ are defined as

$X_1 = [\dot{\alpha}_1, c_{23}\dot{\alpha}_1 - s_{23}\dot{q}_1\dot{q}_2 - s_{23}\dot{q}_1\dot{q}_3 - s_{23}\dot{q}_2\dot{q}_3 + 0.5s_{23}(\dot{q}_2+\dot{q}_3)(\dot{q}_1-\alpha_1), c_2\dot{\alpha}_1 - s_2\dot{q}_1\dot{q}_2 + 0.5s_2\dot{q}_2(\dot{q}_1-\alpha_1), c_3\dot{\alpha}_2 + c_3\dot{\alpha}_1 - s_3\dot{q}_1\dot{q}_3 - s_3\dot{q}_2\dot{q}_3 + 0.5s_3\dot{q}_3(\dot{q}_1-\alpha_1) + 0.5s_3\dot{q}_3(\dot{q}_2-\alpha_2), \dot{\alpha}_2, c_{23}\dot{\alpha}_2 + c_{23}\dot{\alpha}_3 - s_{23}\dot{q}_2^2 - s_{23}\dot{q}_3^2 + 0.5s_{23}(\dot{q}_2+\dot{q}_3)(\dot{q}_2-\alpha_2) + 0.5s_{23}(\dot{q}_2+\dot{q}_3)(\dot{q}_3-\alpha_3), c_2\dot{\alpha}_2 - s_2\dot{q}_2^2 + 0.5s_2\dot{q}_2(\dot{q}_2-\alpha_2), \dot{\alpha}_3, c_3\dot{\alpha}_3 - s_3\dot{q}_3^2 + 0.5s_3\dot{q}_3(\dot{q}_3-\alpha_3), 0, \dot{q}_1]$,

$X_2 = [0, 0, 0, c_3\dot{\alpha}_1 - s_3\dot{q}_1\dot{q}_3 - s_3\dot{q}_2\dot{q}_3 + 0.5s_3\dot{q}_3(\dot{q}_1-\alpha_1), \dot{\alpha}_1 + \dot{\alpha}_2, c_{23}\dot{\alpha}_1 + s_{23}\dot{q}_1^2 + 0.5s_{23}(\dot{q}_2+\dot{q}_3)(\dot{q}_1-\alpha_1), c_2\dot{\alpha}_1 + s_2\dot{q}_1^2 + 0.5s_2\dot{q}_2(\dot{q}_1-\alpha_1), \dot{\alpha}_3, c_3\dot{\alpha}_3 - s_3\dot{q}_3^2 + 0.5s_3\dot{q}_3(\dot{q}_3-\alpha_3), c_3\dot{\alpha}_2 + 0.5s_3\dot{q}_3(\dot{q}_2-\alpha_2), \dot{q}_2]$,

$X_3 = [0, 0, 0, s_3\dot{q}_1\dot{q}_2, 0, s_{23}\dot{q}_1^2 + c_{23}\dot{\alpha}_1 + 0.5s_{23}(\dot{q}_2+\dot{q}_3)(\dot{q}_1-\alpha_1), 0, \dot{\alpha}_1 + \dot{\alpha}_2 + \dot{\alpha}_3, c_3\dot{\alpha}_1 + c_3\dot{\alpha}_2 + s_3\dot{q}_1^2 + s_3\dot{q}_2^2 + 0.5s_3\dot{q}_3(\dot{q}_1-\alpha_1) + 0.5s_3\dot{q}_3(\dot{q}_2-\alpha_2), 0, \dot{q}_3]$.

During the experiment, the 3-DoF robot manipulator is driven to track the desired sinusoidal trajectory $q_r \in \mathbb{R}^3$ designed as

$$q_r = (1 + \sin(\frac{t}{2} - \frac{\pi}{2}))k_{amp}, 5 \leq t \leq 143,$$

where $k_{amp} = [0.2, 0.5, 0.8]^\top$ is the coefficient vector to distribute different amplitudes to each joint. The desired trajectories of three joints are displayed in Figure 2.8a. Considering the required safety and performance issues, for joint 1, the safety set is set as $\bar{\mathcal{S}}_1 = \{-0.1 < q_1 < 0.52\}$, the performance set is chosen as $\mathcal{P}_1 = \{-0.1 < e_{1_1} < 0.12\}$; For joint 2, the safety set is designed as $\bar{\mathcal{S}}_2 = \{-0.1 < q_2 < 1.15\}$, the performance set follows $\mathcal{P}_2 = \{-0.1 < e_{1_2} < 0.15\}$; The safety set and performance set for joint 3 follows $\bar{\mathcal{S}}_3 = \{-0.15 < q_3 < 1.8\}$ and $\mathcal{P}_3 = \{-0.15 < e_{1_3} < 0.2\}$, respectively. To ensure that the 3-DoF robot manipulator track the desired trajectory while satisfying the above safety and performance criteria, parameters are set as $k_a = [0.1, 0.1, 0.15]^\top$, $k_b = [0.12, 0.15, 0.2]^\top$, $k_c = [-0.1, -0.1, -0.15]^\top$, and $k_d = [0.52, 1.15, 1.8]^\top$. The parameters for the TF-CL method are set as: $\Gamma = \mathrm{diag}(0.06 I_{11\times1})$, $k_h = 0.4$, $k_t = 0.8$, $P = 15$, and $\varepsilon = 0.1$. The time constant of the filter is set as $k = 0.001$. Initial values of the estimated parameters are set as $\hat{\theta}(0) = 0_{11\times1}$. For the BLF based control law (2.42), the parameters are set as $k_1 = \mathrm{diag}(20, 20, 20)$, $k_2 = \mathrm{diag}(25, 25, 25)$. Initial values are set as $x_1(0) = [0, 0, 0]^\top$, $x_2(0) = [0, 0, 0]^\top$.

To verify the robustness of our proposed method, the experiment is conducted with different payloads under the same parameter settings mentioned above. The payloads are installed to the end-effector of the manipulator. The tracking errors of three joints with different payloads are displayed in Figure 2.8b, Figure 2.8c and Figure 2.8d, respectively. It is observed that the robot manipulator could track the desired trajectory precisely even under different payloads, and the tracking errors always lie in the required performance set $\mathcal{P}_i, i = 1, 2, 3$.

## 2.7 Summary

This chapter presents a stable adaptive control strategy with guaranteed safety and performance based on BLF, TF-CL, and backstepping. The TF-CL based parameter estimation update law guarantees that the estimated parameters converge to their desired values fast without incorporating external noises to satisfy the PE condition. The joint acceleration information is avoided using the torque filtering technique. Based on the estimated model, our proposed control strategy drives the uncertain $n$-link robot manipulator to track the desired trajectory efficiently, while satisfying the requirements of safety and performance simultaneously. It is proven that the system output always remains in the predefined safety set, the tracking error is bounded by the performance set, and the parameter estimation error finally converges to zero asymptotically. Both simulation and experiment results validate the effectiveness of our proposed approach.

The adopted BLF and TF-CL work together to safely improve the performance. However, the initial system state should lie in the safe set. Otherwise, the singularity problem would happen. To improve the generality and practicability of our method, the future work aims to extend the developed method to provide guaranteed performance and safety on full states even under the consideration of input saturation. Besides, the considered safety issue regarding restricted operation range in this paper allows us to integrate objectives of both safety and performance. The future work aims to extend the proposed method to tackle general safety concepts, e.g., collision avoidance with dangerous regions.

# Input-to-State Stability Meets Barrier Lyapunov Function for Provable Robust Safety 3

The preceding chapter focuses on the design of joint-space tracking controllers. This chapter devotes to task-space tracking control of robot manipulators. The common BLF used in Chapter 2 is extended to formulate the input-to-state stable with provable safety BLF (ISS-PS-BLF), which is utilized with backstepping to design a tracking controller with provable stability and safety under uncertainty. Combining with available safe planning algorithms [68]–[71], our developed tracking controller in this chapter could drive controlled plants to accomplish provable safe execution under uncertainties. The organization of this chapter is as follows. Section 3.1 first presents the preliminaries and the problem formulation. Then, incremental system is developed in Section 3.2 using time-delayed signals. The formulated incremental system allows us to achieve kinematics and dynamics free control of robot manipulators. This departs from Chapter 2 that a known model structure is required for the model learning. The incremental system mentioned above serves as basis for the recursive controller design process illustrated in Section 3.3. Our developed approach is numerically validated in Section 3.4. Finally, Section 3.6 summarizes this chapter

## 3.1 Preliminaries and Problem Formulation

### 3.1.1 Input-to-State Stable with Provable Safety BLF

This subsection clarifies the required preliminaries to develop our approach by focusing on the state evolution model

$$\dot{x} = f(x) + g(x)u(x) + g(x)d, \tag{3.1}$$

where $x \in \mathbb{R}^n$, $u(x) : \mathbb{R}^n \to \mathbb{R}^m$ are the system state and control input, respectively. $f(x) : \mathbb{R}^n \to \mathbb{R}^n$, $g(x) : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ are bounded and locally Lipschitz. $d \in \mathbb{L}_\infty^m$ is the assumed bounded disturbance with the (essential) supremum norm $|d|_\infty := \sup |d(t)|, t \geq 0$.

As stated in [72], iff the system (3.1) admits an input-to-state stable (ISS) Lyapunov function as Definition 3.1, the system (3.1) is ISS as Definition 3.2. Therefore, designers could realize the ISS control by using one ISS-Lyapunov function to perform the controller design.

**Definition 3.1** (ISS-Lyapunov Function [72]). *A smooth function $V(x) : \mathbb{R}^n \to \mathbb{R}_0^+$ is an ISS-Lyapunov function for system* (3.1) *if there exists $\alpha_1, \alpha_2, \alpha_3, \alpha_4 \in \mathcal{K}_\infty$ such that $\forall$ x, d*

$$\alpha_1(|x|) \leq V(x) \leq \alpha_2(|x|) \tag{3.2a}$$

$$\dot{V}(x, d) \leq -\alpha_3(|x|) + \alpha_4(|d|). \tag{3.2b}$$

**Definition 3.2** (ISS [73]). *The system* (3.1) *is ISS if there exists $\lambda \in \mathcal{KL}$ and $\gamma \in \mathcal{K}_\infty$*

$$|x(t, x_0, d)| \leq \lambda(|x_0|, t) + \gamma(|d|_\infty), \forall\, x_0,\, d, \forall\, t \geq 0.$$

The Definitions 3.1-3.2 inspire us to extend the original BLF [74], which is defined on an ideal accurate dynamics $\dot{x} = f(x) + g(x)u(x)$, to the uncertainty scenario. The resulting ISS-PS-BLF formulated in Definition 3.3 is a valid ISS-Lyapunov function in Definition 3.1 given the establishment of the inequalities (3.3a), (3.3b), (3.3c). Furthermore, (3.3d) implies that a bounded ISS-PS-BLF would confine the state $x_1$ into the predetermined safe region $\mathbb{S}$. Thereby, our defined ISS-PS-BLF (3.3) provides designers with an efficient tool to realize the desired input-to-sate stabilization with provable safety.

**Definition 3.3** (ISS-PS-BLF)**.** *A smooth function $V(x) := V_1(x_1) + V_2(x_2) \in \mathbb{R}_0^+$, where $x := [x_1^\top, x_2^\top]^\top \in \mathbb{R}^{n_1+n_2}$, $x_1 \in \mathbb{R}^{n_1}$, $x_2 \in \mathbb{R}^{n_2}$, is an ISS-PS-BLF for the system* (3.1) *on the open region $\mathbb{S} := \{x_1 \in \mathbb{R}^{n_1} : -\underline{\epsilon} \prec x_1 \prec \overline{\epsilon}\}$, where $\underline{\epsilon}_i$, $\overline{\epsilon}_i \in \mathbb{R}^+$, $\forall i \in \{1, \cdots, n_1\}$, if there exist functions $\beta_i \in \mathcal{K}_\infty$, $i = 1, \cdots, 6$, such that $\forall$ $x$, $d$*

$$\beta_1(|x_1|) \le V_1(x_1) \le \beta_2(|x_1|) \tag{3.3a}$$

$$\beta_3(|x_2|) \le V_2(x_2) \le \beta_4(|x_2|) \tag{3.3b}$$

$$\dot{V}(x, d) \le -\beta_5(|x|) + \beta_6(|d|) \tag{3.3c}$$

$$V_1(x_1) \to \infty, \quad x_1 \to \partial\mathbb{S}. \tag{3.3d}$$

Combining with (3.3) and the result in [74], this work utilizes the following candidate ISS-PS-BLF

$$V(x) := \underbrace{\frac{1}{2}\sum_{i=1}^{n_1}\left[\frac{\overline{\epsilon}_i\underline{\epsilon}_i x_{1_i}}{(\overline{\epsilon}_i - x_{1_i})(\underline{\epsilon}_i + x_{1_i})}\right]^2}_{V_1(x_1)} + \underbrace{\frac{1}{2}x_2^\top x_2}_{V_2(x_2)}, \tag{3.4}$$

to conduct the controller design.

### 3.1.2 Problem Formulation

This work attempts to realize the provable safe execution of uncertain autonomous systems in obstacle-filled environments. Our solution to this nontrivial problem is our developed SP-PGC scheme: combination with the *performance-guaranteed control* that explicitly quantifies the control-level performance under uncertainty, and the *safe planning* where the collision-free desired trajectory is planned within the consideration of the attainable performance of the utilized controllers, see Fig. 3.1.

This work adopts the safe planning algorithms that satisfy the requirement presented in Assumption 3.1.

**Assumption 3.1.** *The planning level outputs a collision-free desired trajectory $p_d \in \mathbb{R}^m$ lying in a safe set $\mathbb{C} := \Big\{p(t) \in \mathbb{R}^m : \underline{p}(t) \prec p(t) \prec \overline{p}(t)\Big\}$, where $\underline{p}(t)$, $\overline{p}(t) \in \mathbb{R}^m$.*

Assumption 3.1 easily holds using off-the-self planning algorithms [75] conducted based on buffered obstacles [1], whose buffer size is $\epsilon \in \mathbb{R}^m$, $\epsilon_i \in \mathbb{R}^+$, $\forall i \in \{1, \cdots, m\}$ (see the left below figure in Fig. 3.1). The safe execution region is $\mathbb{C} := \Big\{p \in \mathbb{R}^m : \underline{p} := p_d - \epsilon \prec p \prec \overline{p} := p_d + \epsilon\Big\}$. Regarding this case, the tracking error $e_1 := p - p_d \in \mathbb{R}^m$ should satisfy $e_1 \in \mathbb{E} :=$

[1] This case matches the robot manipulator numerical and experimental validations displayed in Section 3.4 and Section 3.5.

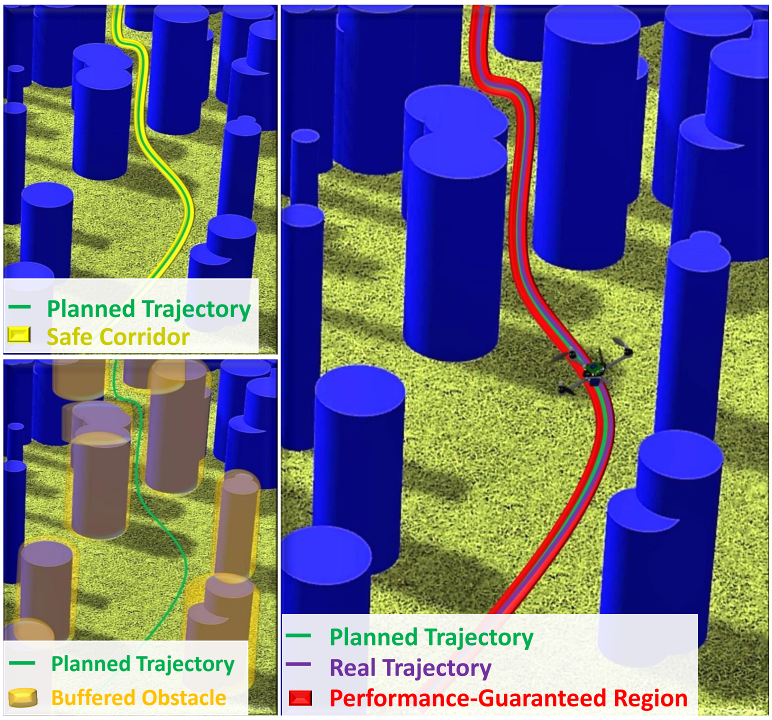


Figure 3.1: Schematic of the SP-PGC scheme. The safe reference trajectory is planned within the consideration of the control-level performance bound $\epsilon$, either establishing safe corridors with radius $\epsilon$ (the left above figure), or floating obstacles via size $\epsilon$ (the left below figure). The control level guarantees tracking performance even under uncertainties and disturbances ignored in the planning level (the right figure).

$\{e_1 \in \mathbb{R}^m : -\underline{\epsilon} := -\epsilon \prec e_1 \prec \epsilon := \overline{\epsilon}\}$ to achieve safety, where $\underline{\epsilon}$, $\overline{\epsilon} \in \mathbb{R}^m$ are lower and upper performance bounds of $e_1$. Alternatively, Assumption 3.1 is easily satisfied by reachable set based algorithms [69], or corridor (funnel) based algorithms [70], [71] (see the left above figure in Fig. 3.1). In this case, $e_1 \in \mathbb{E} := \left\{e_1 \in \mathbb{R}^m : -\underline{\epsilon} := \underline{p} - p_d \prec e_1 \prec \overline{p} - p_d := \overline{\epsilon}\right\}$ should be guaranteed to avoid collision during practical executions [2].

Through the aforementioned analysis, we interpret the provable safe execution under uncertainty problem as a robust performance-guaranteed tracking control problem. This problem is nontrivial given that both state and input constraints are considered under model uncertainties and environmental disturbances. We solve this nontrivial problem via our formulated incremental system in Section 3.2 and the ISS-PS-BLF facilitated controller in Section 3.3.

## 3.2 Data Informed Incremental System

This section utilizes time-delayed data to formulate the incremental system that equivalently describes the movement of the original autonomous system (3.1). By doing so, no explicit model knowledge (kinematics and/or dynamics) is required. The formulated incremental

[2] This case matches the quadrotor numerical simulation in Section 3.4.2.

system serve as the basis for the controller design process presented in Section 3.3.

### 3.2.1 Development of Incremental System

In the following, we focus on the system (3.1) satisfying Assumption 3.2 to clarify the formulation of the associated time-delayed data informed incremental system.

**Assumption 3.2.** *The columns $g_1, g_2, \cdots, g_m \in \mathbb{R}^n$ of the input function $g = [g_1, g_2, \cdots, g_m]$ are linearly independent.*

**Remark 3.1.** *Here $g(x)$ is assumed to be full column rank such that its pseudo inverse $g^\dagger$ could be expressed as a simple algebraic formula (the inverse of $g^\top(x)g(x)$ exists). This property is widely observed in many physical systems, such as the quadrotor presented in Example 1, and the robot manipulator shown in Example 2 fulfill such a property.*

Firstly, introducing a prior-chosen constant matrix $\bar{g} \in \mathbb{R}^{n\times m}$ and multiplying its pseudo inverse $\bar{g}^\dagger$ on (3.1), we obtain

$$\bar{g}^\dagger \dot{x} = h + u, \tag{3.5}$$

where $h := (\bar{g}^\dagger - g^\dagger)\dot{x} + g^\dagger f + d \in \mathbb{R}^n$ embodies the unknown knowledge of the system (3.1).

Then, we use time-delayed data to estimate $h$ as

$$\hat{h} = h_0 = \bar{g}^\dagger \dot{x}_0 - u_0, \tag{3.6}$$

where $(\bullet)_0 = (\bullet)(t - t_s)$ denotes time-delayed data, and $t_s \in \mathbb{R}^+$ is the sampling time.

Finally, substituting (3.6) into (3.5), we get the incremental system:

$$\dot{x} = \dot{x}_0 + \bar{g}\Delta u + \bar{g}\xi, \tag{3.7}$$

where $\Delta u := u - u_0 \in \mathbb{R}^n$, and $\xi := h - \hat{h} \in \mathbb{R}^n$ is the estimation error proved to be bounded and vanishing in Lemma 3.1 under the properly chosen $\bar{g}$.

**Remark 3.2.** *The theoretical derivation processes* (3.5)–(3.7) *mentioned above exploits time-delayed data to transform model uncertainties and external disturbances of* (3.1) *into a provably bounded estimation error $\xi$ of* (3.7). *This is beneficial to achieve provable safety under uncertainty given that the influence of the estimation error $\xi$ on safety could be rigorously analyzed via an ISS approach. To achieve the same goal with our work, however, related works either estimate disturbance bounds explicitly using computation-intensive methods such as GP [34] or directly assume a known bound of uncertainty [76], which often results in conservative behaviours.*

Through the processes (3.5)-(3.7), we get an equivalent form of (3.1) without using explicit model information. In the subsequent Section (3.3), we use the above formulated incremental system (3.7) and our proposed ISS-PS-BLF (3.4) together to design the robust tracking controller with guaranteed performance.

Before proceeding to the controller design process, we provide two explicit examples to clarify how to derive the associated incremental systems from the quadrotor dynamics and the robot manipulator kinematics and dynamics.

**Example 1** (Quadrotor). *The Euler-Lagrange (E-L) equation of a quadrotor follows [77]*

$$m\ddot{\zeta} + mg_c I_z = RT_B + T_d \tag{3.8a}$$

$$J(\eta)\ddot{\eta} + C(\eta, \dot{\eta})\dot{\eta} = \tau_B + \tau_{Bd}, \tag{3.8b}$$

*where $\zeta := [x, y, z]^\top \in \mathbb{R}^3$, and $\eta := [\phi, \theta, \psi]^\top \in \mathbb{R}^3$ represent the absolute linear position and Euler angles defined in the inertial frame, respectively; $m \in \mathbb{R}^+$ denotes the mass of the quadrotor; $g_c \in \mathbb{R}^+$ is the gravity constant; $I_z := [0, 0, 1]^\top$ represents a column vector; $T_B = [0, 0, T]^\top \in \mathbb{R}^3$, where $T \in \mathbb{R}$ is the thrust in the direction of the body z-axis; $\tau_B := [\tau_\phi, \tau_\theta, \tau_\psi]^\top \in \mathbb{R}^3$ denotes the torques in the direction of the corresponding body frame angles; $T_d =\in \mathbb{R}^3$ and $\tau_d \in \mathbb{R}^3$ denote the external disturbance; $R$, $J(\eta)$, $C(\eta, \dot{\eta}) \in \mathbb{R}^{3\times3}$ represent the rotation matrix, Jacobian matrix, and Coriolis term, respectively. We could rewrite the above translation dynamics* (3.8a) *or the attitude dynamics* (3.8b) *as*

$$\dot{x}_1 = x_2 \tag{3.9a}$$

$$\dot{x}_2 = f + gu + gd, \tag{3.9b}$$

*via letting $x_1 := \zeta$ or $\eta \in \mathbb{R}^3$, $x_2 := \dot{\zeta}$ or $\dot{\eta} \in \mathbb{R}^3$, $f := -g_c I_z$ or $-J^{-1}C(\eta, \dot{\eta})\dot{\eta} \in \mathbb{R}^3$, $g := R/m$ or $J^{-1} \in \mathbb{R}^{3\times3}$, $u = T_B$ or $\tau_B \in \mathbb{R}^3$, $d := R^{-1}T_d$ or $\tau_{Bd} \in \mathbb{R}^3$, respectively. Applying the theoretical derivation processes* (3.5)–(3.7) *mentioned above on* (3.9b)*, we get*

$$\dot{x}_1 = x_2 \tag{3.10a}$$

$$\dot{x}_2 = \dot{x}_{2,0} + \bar{g}\Delta u + \bar{g}\xi, \tag{3.10b}$$

*which is an equivalent representation of* (3.8) *but without explicit knowledge of quadrotor dynamics.*

**Example 2** (Robot Manipulator). *The Cartesian-space position $p \in \mathbb{R}^m$ of the robot manipulator end-effector is expressed as*

$$p = h(q), \tag{3.11}$$

*where $q \in \mathbb{R}^n$ is the joint-space angle vector, and $h(q) : \mathbb{R}^n \to \mathbb{R}^m$ is the differential forward kinematics. Note that $m \leq n$ holds. The end-effector velocity and acceleration $\dot{p}$, $\ddot{p} \in \mathbb{R}^m$ are related to the joint velocity and acceleration $\dot{q}$, $\ddot{q} \in \mathbb{R}^n$ as*

$$\dot{p} = J(q)\dot{q} \tag{3.12a}$$

$$\ddot{p} = \dot{J}(q)\dot{q} + J(q)\ddot{q}, \tag{3.12b}$$

*where $J(q) := \partial h(q)/\partial q \in \mathbb{R}^{m\times n}$ is the Jacobian matrix. Besides, the robot manipulator dynamics follows [48]*

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + F_v(\dot{q}) = \tau + \tau_d, \tag{3.13}$$

*where $M(q) : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ is the symmetric positive definite inertia matrix; $C(q, \dot{q}) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{n\times n}$ is the matrix of centrifugal and Coriolis terms; $G(q) : \mathbb{R}^n \to \mathbb{R}^n$ represents the gravitational term; $F_v(\dot{q}) : \mathbb{R}^n \to \mathbb{R}^n$ denotes the viscous friction; $\tau_d \in \mathbb{R}^n$ represents the external disturbance.*

*Substituting* (3.12) *into* (3.13) *yields*

$$M_p(q)\ddot{p} + C_p(q,\dot{q})\dot{p} + G(q) + F_v(\dot{q}) = \tau + \tau_d, \tag{3.14}$$

*where* $M_p(q) := M(q)J^\dagger(q) : \mathbb{R}^n \to \mathbb{R}^{n\times m}$, $C_p(q,\dot{q}) := C(q,\dot{q})J^\dagger(q) - M(q)J^\dagger(q)\dot{J}(q)J^\dagger(q) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{n\times m}$. *The pseudo inverse follows* $J^\dagger(q) := (J^\top(q)J(q))^{-1}J^\top(q) : \mathbb{R}^n \to \mathbb{R}^{n\times m}$. *Then, the integrated kinematics and dynamics form* (3.14) *could be rewritten as the form* (3.10) *by denoting* $x_1 := p \in \mathbb{R}^m$, $x_2 := \dot{p} \in \mathbb{R}^m$, $f := -M_p^\dagger(q)(C_p(q,\dot{q})\dot{p} + G(q) + F_v(\dot{q})) \in \mathbb{R}^m$, $g := M_p^\dagger(q) \in \mathbb{R}^{m\times n}$, $u := \tau \in \mathbb{R}^n$, *and* $d := \tau_d \in \mathbb{R}^n$. *Through the theoretical derivation processes* (3.5)–(3.7)*, we would get one associated incremental system of the robot manipulator (in the same form as* (3.10)*) without using explicit information of kinematics and dynamics.*

**Remark 3.3.** *Examples 1-2 build on the assumption that singularities are always avoided during the whole execution process for the quadrotor and the robot manipulator. The systematic method to avoid singularity is beyond the scope of this paper. Besides, we use the pseudo-inverse of the manipulator Jacobian in* (3.14) *to deal with the redundancy problem of the robot manipulator case.*

**Remark 3.4.** *Note that the formulated* (3.14) *in Example 2 departs from the common method [78] that attempts to write* (3.11)*,* (3.12)*, and* (3.13) *together to formulate an integrated kinematics and dynamics form as* $\bar{M}_p(q)\ddot{p} + \bar{C}_p(q,\dot{q})\dot{p} + J(q)G(q) + J(q)F_v(\dot{q}) = J(q)\tau + J(q)\tau_d$, *where* $\bar{M}_p(q) := J(q)M(q)J^\dagger(q) : \mathbb{R}^n \to \mathbb{R}^{m\times m}$, $\bar{C}_p(q,\dot{q}) := J(q)C(q,\dot{q})J^\dagger(q) - J(q)M(q)J^\dagger(q)\dot{J}(q)J^\dagger(q) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{m\times m}$. *Based on this form, the kinematics free control is impossible following the controller design process illustrated in Section 3.3. In particular, a controller in the form* $\tau_p := J(q)\tau$ *will be firstly designed. Then, an inversion calculation using the explicit kinematic knowledge,* $\tau = J^\dagger(q)\tau_p$ *in particular, is required to recover the torque applied at each joint. Our formulated* (3.14) *directly links the joint-space control input* $\tau$ *with the task-space position* $p$*. This allows us to use time-delayed data to realize kinematics free control later.*

## 3.3 Model-Free Performance-Guaranteed Control

This section utilizes our proposed ISS-PS-BLF (3.4) to develop a model-free performance-guaranteed tracking controller through a recursive controller design process. The ISS-PS-BLF provides explicit quantification of realizable tracking errors. This control-level performance quantification could feedback to the planning level to refine planned trajectories accounting for actual implementation tracking errors. The recursive controller design process based on the incremental system formulated in the previous section is illustrated as follows.

*Step 1:* Focusing on (3.10), the position tracking error follows $e_1 := x_1 - p_d \in \mathbb{R}^m$. To ensure that the tracking error $e_1$ always lies in a predetermined performance bound, $e_1 \in \mathbb{E} := \{e_1(t) \in \mathbb{R}^m : -\underline{\epsilon} \prec e_1(t) \prec \bar{\epsilon}\}$ in particular, we use the following Lyapunov function

$$V_1 := \frac{1}{2}\sum_{i=1}^{m}\left[\frac{\bar{\epsilon}_i\underline{\epsilon}_i e_{1_i}}{(\bar{\epsilon}_i - e_{1_i})(\underline{\epsilon}_i + e_{1_i})}\right]^2, \tag{3.15}$$

to facilitate the controller design. The derivative of (3.15) follows

$$\dot{V}_1 = \sum_{i=1}^{m} e_{1_i} \underbrace{\frac{\bar{\epsilon}_i^3 \underline{\epsilon}_i^3 + \bar{\epsilon}_i^2 \underline{\epsilon}_i^2 e_{1_i}^2}{(\bar{\epsilon}_i - e_{1_i})^3 (\underline{\epsilon}_i + e_{1_i})^3}}_{p_i} \dot{e}_{1_i} = e_1^\top P \dot{e}_1, \tag{3.16}$$

where $P := \mathrm{diag}(p_1, p_2, \cdots, p_m) \in \mathbb{R}^{m \times m}$.

Let $e_2 := x_2 - z \in \mathbb{R}^m$, where $z \in \mathbb{R}^m$ is a stabilizing term designed later. Combining with (3.10a), the explicit form of $\dot{e}_1$ used in (3.16) follows

$$\dot{e}_1 = \dot{x}_1 - \dot{p}_d = x_2 - \dot{p}_d = e_2 + z - \dot{p}_d. \tag{3.17}$$

Then, we design $z := \dot{p}_d - P^{-1} L_1 e_1$, wherein $L_1 := \mathrm{diag}(l_{11}, l_{12}, \cdots, l_{1m}) \in \mathbb{R}^{m \times m}$, $l_{1j} \in \mathbb{R}^+$, $j = 1, \cdots, m$. Substituting (3.17) into (3.16) yields

$$\dot{V}_1 = -e_1^\top L_1 e_1 + e_1^\top P e_2. \tag{3.18}$$

*Step 2:* We choose the ISS-PS-BLF as $V := V_1 + V_2$, wherein the explicit of $V_2$ follows

$$V_2 := \frac{1}{2} e_2^\top e_2. \tag{3.19}$$

Then, combining with (3.10b) and (3.16), we get

$$\begin{aligned} \dot{V} &= \dot{V}_1 + \dot{V}_2 \\ &= -e_1^\top L_1 e_1 + e_1^\top P e_2 + e_2^\top (\dot{x}_{2,0} + \bar{g} \Delta u + \bar{g} \xi - \dot{z}). \end{aligned} \tag{3.20}$$

Finally, we develop the incremental control input as

$$\Delta u = \bar{g}^\dagger (\dot{z} - \dot{x}_{2,0} - L_2 e_2 - P e_1), \tag{3.21}$$

to input-to-state stabilize the tracking errors $e_1$ and $e_2$ to a small neighbourhood around zero as proved in Theorem 3.1, wherein $L_2 := \mathrm{diag}(l_{21}, l_{22}, \cdots, l_{2m}) \in \mathbb{R}^{m \times m}$ is a positive definite matrix, $l_{2j} \in \mathbb{R}^+$, $j = 1, \cdots, m$. Accordingly, the control input applied at the controlled plant is recovered as

$$u = u_0 + \Delta u. \tag{3.22}$$

In the following, we theoretically analyze the properties of our designed performance-guaranteed control strategy (3.22). We firstly present the rigorous proof of the bounded estimation error in Lemma 3.1. Then, the proved bounded estimation error allows us to analyse the desirable provable safety under uncertainty in Theorem 3.1.

**Lemma 3.1.** *Given a sufficiently high sampling rate, there exists a positive constant* $\bar{\xi} \in \mathbb{R}^+$ *such that* $|\xi| \leq \bar{\xi}$.

*Proof.* Combining with (3.5), (3.6) and (3.10), we get

$$\begin{aligned} \xi = h - h_0 = (\bar{g}^\dagger - g^\dagger)(\dot{x}_2 - \dot{x}_{2,0}) + (g_0^\dagger - g^\dagger)\dot{x}_{2,0} \\ + g^\dagger (f - f_0) + (g^\dagger - g_0^\dagger) f_0 + d - d_0. \end{aligned} \tag{3.23}$$

Besides, focusing on (3.10b), the following equation holds

$$\begin{aligned} \dot{x}_2 - \dot{x}_{2,0} &= f + gu + gd - f_0 - g_0u_0 - g_0d_0 \\ &= g\Delta u + (g - g_0)u_0 + f - f_0 + g(d - d_0) + (g - g_0)d_0. \end{aligned} \tag{3.24}$$

Then, substituting (3.24) into (3.23) reads

$$\xi = (\bar{g}^\dagger g - I_{n\times n})\Delta u + \delta_1, \tag{3.25}$$

where $\delta_1 := \bar{g}^\dagger(g - g_0)u_0 + \bar{g}^\dagger(f - f_0) + \bar{g}^\dagger g(d - d_0) + \bar{g}^\dagger(g - g_0)d_0 \in \mathbb{R}^n$. For representation simplicity, let $v := \dot{z} - L_2e_2 - Pe_1$. Accordingly, $v_0 := \dot{z}_0 - L_2e_{2,0} - P_0e_{1,0}$. Then, invoking (3.5), (3.6) and (3.21), we get

$$\begin{aligned} \Delta u = \bar{g}^\dagger(v - \dot{x}_{2,0}) &= \bar{g}^\dagger v - h_0 - u_0 \\ &= \bar{g}^\dagger v - (\bar{g}^\dagger - g_0^\dagger)\dot{x}_{2,0} + g_0^\dagger f_0 - u_0 \\ &= \bar{g}^\dagger v - (\bar{g}^\dagger - g_0^\dagger)(f_0 + g_0u_0) + g_0^\dagger f_0 - u_0 \\ &= \bar{g}^\dagger v - \bar{g}^\dagger(f_0 + g_0u_0) \\ &= \bar{g}^\dagger(v - v_0) - \bar{g}^\dagger(\dot{x}_{2,0} - v_0). \end{aligned} \tag{3.26}$$

Combining (3.10b) with (3.21) yields

$$\dot{x}_2 = v + \bar{g}\xi. \tag{3.27}$$

Besides, according to (3.27), we get

$$\xi = \bar{g}^\dagger(\dot{x}_2 - v),\ \xi_0 = \bar{g}^\dagger(\dot{x}_{2,0} - v_0). \tag{3.28}$$

Substituting (3.28) into (3.26) implies

$$\Delta u = \bar{g}^\dagger(v - v_0) - \xi_0. \tag{3.29}$$

Finally, substituting (3.29) into (3.25), we get

$$\xi = (I_{n\times n} - \bar{g}^\dagger g)\xi_0 + \delta_1 + \delta_2, \tag{3.30}$$

where $\delta_2 := (\bar{g}^\dagger g - I_{n\times n})\bar{g}^\dagger(v - v_0) \in \mathbb{R}^n$.

For theoretical analytical purpose, we rewrite (3.30) into a discrete-time domain as

$$\xi(k) = (I_{n\times n} - \bar{g}^\dagger g(k))\xi(k-1) + \delta_1(k) + \delta_2(k). \tag{3.31}$$

Given a sufficiently high sampling rate, it is reasonable to assume that there exist positive constants $\overline{\delta}_1$, $\overline{\delta}_2 \in \mathbb{R}^+$ such that $|\delta_1| \le \overline{\delta}_1$, and $|\delta_2| \le \overline{\delta}_2$ hold. We choose the value of $\bar{g}$ to satisfy $\left|I_{n\times n} - \bar{g}^\dagger g(k)\right| \le l < 1$, $l \in \mathbb{R}^+$. Then, the following equation holds

$$\begin{aligned} |\xi(k)| &\le l\,|\xi(k-1)| + \bar{\delta}_1 + l\bar{\delta}_2 \\ &\le l^2\,|\xi(k-2)| + (l+1)(\bar{\delta}_1 + l\bar{\delta}_2) \\ &\le \cdots \le l^k\,|\xi(0)| + \frac{\bar{\delta}_1 + l\bar{\delta}_2}{1-l} := \bar{\xi} \end{aligned} \tag{3.32}$$

As $k \to \infty$, $\bar{\xi} \to \frac{\bar{\delta}_1 + l\bar{\delta}_2}{1-l}$. □

**Theorem 3.1.** *Consider the system* (3.10) *with the controller* (3.22)*. Given Assumption 3.1 for initial conditions lying in the safe set* $\mathbb{C}$*, the following properties hold:*

*1) The tracking errors* $e_1$ *and* $e_2$ *are input-to-state stabilizing to a small neighbourhood around zero.*

*2) The Cartesian position tracking error* $e_1$ *satisfies* $e_1 \in \mathbb{E}$*.*

*3) The controlled plant realizes provable safe execution* $p \in \mathbb{C}$ *under model uncertainties and environmental disturbances.*

*Proof. Proof of 1)* Substituting (3.21) into (3.20) yields

$$\begin{aligned}
\dot{V} &= -e_1^\top L_1 e_1 - e_2^\top L_2 e_2 + e_2^\top \bar{g}\xi \\
&= -e_1^\top L_1 e_1 - e_2^\top (L_2 - I_{m\times m}) e_2 - (e_2^\top e_2 - e_2^\top \bar{g}\xi) \\
&= -e_1^\top L_1 e_1 - e_2^\top (L_2 - I_{m\times m}) e_2 \\
&\quad - \left| e_2 - \frac{1}{2}\bar{g}\xi \right|^2 + \frac{1}{4}\left|\bar{g}\xi\right|^2 \\
&\leq -e_1^\top L_1 e_1 - e_2^\top (L_2 - I_{m\times m}) e_2 + \frac{1}{4}\left|\bar{g}\right|^2 \left|\xi\right|^2 \\
&= -e^\top L e + \frac{\left|\bar{g}\right|^2}{4}\left|\xi\right|^2 \leq -\eta_{\min}(L)\left|e\right|^2 + \frac{\left|\bar{g}\right|^2}{4}\left|\xi\right|^2 \\
&\leq -(\eta_{\min}(L) + \frac{\left|\bar{g}\right|^2}{4})\left|e\right|^2, \quad \forall \left|e\right| > \left|\xi\right|,
\end{aligned} \tag{3.33}$$

where $e := [e_1^\top, e_2^\top]^\top \in \mathbb{R}^{2m}$, $L := \mathrm{diag}(L_1, L_2 - I_{m\times m}) \in \mathbb{R}^{2m\times 2m}$, and the minimum eigenvalue of $L$ is $\eta_{\min}(L) := \min\{\eta_{\min}(L_1), \eta_{\min}(L_2 - I_{m\times m})\}$. Note that $L_2 - I_{m\times m} > 0$ is required to make $L$ as one positive definite matrix. This requirement provides practitioners with guidelines to choose suitable values of $L_2$. It is concluded that the tracking errors $e_1$ and $e_2$ are ISS with $\alpha_3(\bullet) = -\eta_{\min}(L)\left|\bullet\right|^2$, $\alpha_4(\bullet) = \frac{\left|\bar{g}\right|^2}{4}\left|\bullet\right|^2$ based on Definition 3.1. Then, $\left|e(t)\right| \leq \lambda(e(t_0), t) + \gamma(\left|\xi(t)\right|_\infty)$ holds according to Definition 3.2, i.e., the tracking error $e$ remains in a ball with radius $\lambda(e(t_0), t) + \gamma(\left|\xi(t)\right|_\infty)$. Besides, as time $t$ increases, the tracking error $e$ approaches to a smaller ball of radius $\gamma(\left|\xi(t)\right|_\infty)$ given that for fixed $e(t_0)$, the $\mathcal{KL}$ function $\lambda$ decreases to zero as $t \to \infty$.

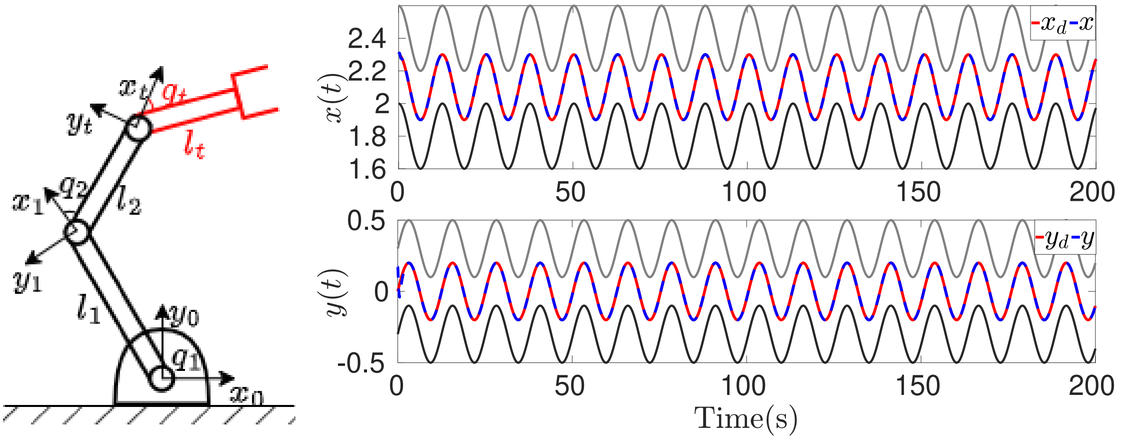
*Proof of 2)* The establishment of (3.33) implies that $V$ is bounded. Thereby, $V_1$ is bounded. Given that $e_1 \to -\underline{\epsilon}$ or $e_1 \to \bar{\epsilon}$ leads to $V_1 \to \infty$ according to (3.3d). Thus, the bounded $V_1$ proves that the tracking error $e_1$ lies in the set $\mathbb{E}$.

*Proof of 3)* The actual execution position of the controlled plant is $p = p_d + e_1$. Based on the fact that $e_1 \in \mathbb{E}$, the possible trajectory lies in the set $\bar{\mathbb{C}} := \{p(t) \in \mathbb{R}^{n_1} : p_d - \underline{\epsilon} \prec p(t) \prec p_d + \bar{\epsilon}\}$. By choosing $-\underline{\epsilon} > \underline{p}(t) - p_d$ and $\bar{\epsilon} \prec \bar{p}(t) - p_d$ and combining with Assumption 3.1, $\underline{p}(t) \prec p_d - \underline{\epsilon}$ and $p_d + \bar{\epsilon} \prec \bar{p}(t)$ hold. Thus, it is proved that $\bar{\mathbb{C}} \in \mathbb{C}$, i.e., the actual execution trajectory $p(t)$ always lies in the safe region $\mathbb{C}$ even the controlled plant (3.1) suffers from model uncertainties and environmental disturbances. □

## 3.4 Numerical Simulation

### 3.4.1 Safe Operation of Robot Manipulator

This subsection concentrates on a 2-DoF robot manipulator Cartesian-space tracking task under varying kinematics settings to exemplify the kinematics free property of our method.



(a) 2-DoF robot and tool. (b) Trajectories of $x(t)$, $y(t)$ and safe boundary.

Figure 3.2: The robot and the end-effector Cartesian-space position under varying kinematics settings (Task 1 case).

The explicit kinematic and dynamic knowledge of the robot manipulator used for simulation purposes is referred to in [79].

The robot manipulator in one restricted environment is required to grasp diverse tools (tools in different lengths $l_t$, and grasping angles $q_t$) to complete different tracking tasks in a provable safe way, see Fig. 3.2a. In particular, the working space of the end-effector should be always confined to one specific region that is treated as a prohibited area for humans or other robots. Note that the information of kinematics, dynamics, and tool (i.e., values of $l_t$ and $q_t$) are unavailable to practitioners to perform the controller design. To accomplish the above task, the collision-free desired trajectory $p_d \in \mathbb{R}^2$ (a circle with center $c := (c_x, c_y)$ and radius $r$) is firstly planned under buffered obstacles with buffer size $\epsilon = 0.3$. This buffer size then serves as the performance bound of our designed performance-guaranteed control strategy (3.22) to ensure that the tool end track the collision-free desired trajectory $p_d$ with the predetermined tracking accuracy $\underline{\epsilon} = [-0.3, -0.3]^\top$, and $\bar{\epsilon} = [0.3, 0.3]^\top$.

The initial conditions are set as $q(0) = [0, 0]^\top$, $\tau(0) = [0, 0]^\top$. The parameters required for the incremental control input (3.21) are set as: $\bar{g} = \mathrm{diag}(10, 10)$, $L_1 = \mathrm{diag}(1, 1)$, and $L_2 = \mathrm{diag}(2, 2)$. The sampling rate is 1kHz. Note that we always keep the same parameter setting to conduct the following different numerical simulations. This exemplifies the robustness of our developed method.

The robot manipulator uses different tools (different initial lengths $l_{t_0}$) installed with different initial angles $q_{t_0}$ to complete the following four Cartesian-space tracking tasks. Task 1: $c_1 = (2.1, 0)$, $r_1 = 0.2$ m, $l_{t_0} = 0.2$ m, $q_{t_0} = \pi/6$; Task 2: $c_2 = (2.3, 0.1)$, $r_2 = 0.2$ m, $l_{t_0} = 0.4$ m, $q_{t_0} = \pi/4$; Task 3: $c_3 = (2.3, 0.6)$, $r_3 = 0.2$ m, $l_{t_0} = 0.6$ m, $q_{t_0} = \pi/3$; and Task 4: $c_4 = (2, 0.9)$, $r_4 = 0.2$ m, $l_{t_0} = 0.8$ m, $q_{t_0} = \pi/2$. To fully exemplify the kinematics free property of our method, we additionally consider the non-trivial varying kinematics setting here. In particular, we purposely set the tool length as $l_t = l_{t_0} - 0.0002\ t$ and the grasping angle as $q_t = q_{t_0} - 0.002\ t$ during the working process, where $t$ denotes the current time. The above varying tool length $l_t$ and grasping angle $q_t$ might be caused by wear and tear or poor fixation in industrial productions. This varying kinematic setting invalidates common approaches that require the inverse kinematics calculation.

The Cartesian-space position trajectories displayed in Fig. 3.2b illustrate that the tool

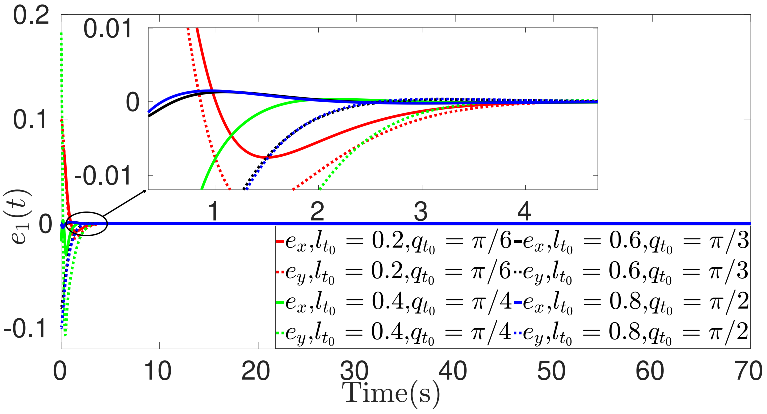


Figure 3.3: The trajectories of the Cartesian-space tracking error $e_1 := [e_x, e_y]^\top$ under varying kinematics and different tasks.

end always lies in the predetermined safe region. Furthermore, the high-accuracy tracking performance shown in Fig. 3.3 validates the flexibility and adaptability of our developed approach towards different tasks under the varying kinematic settings mentioned above.

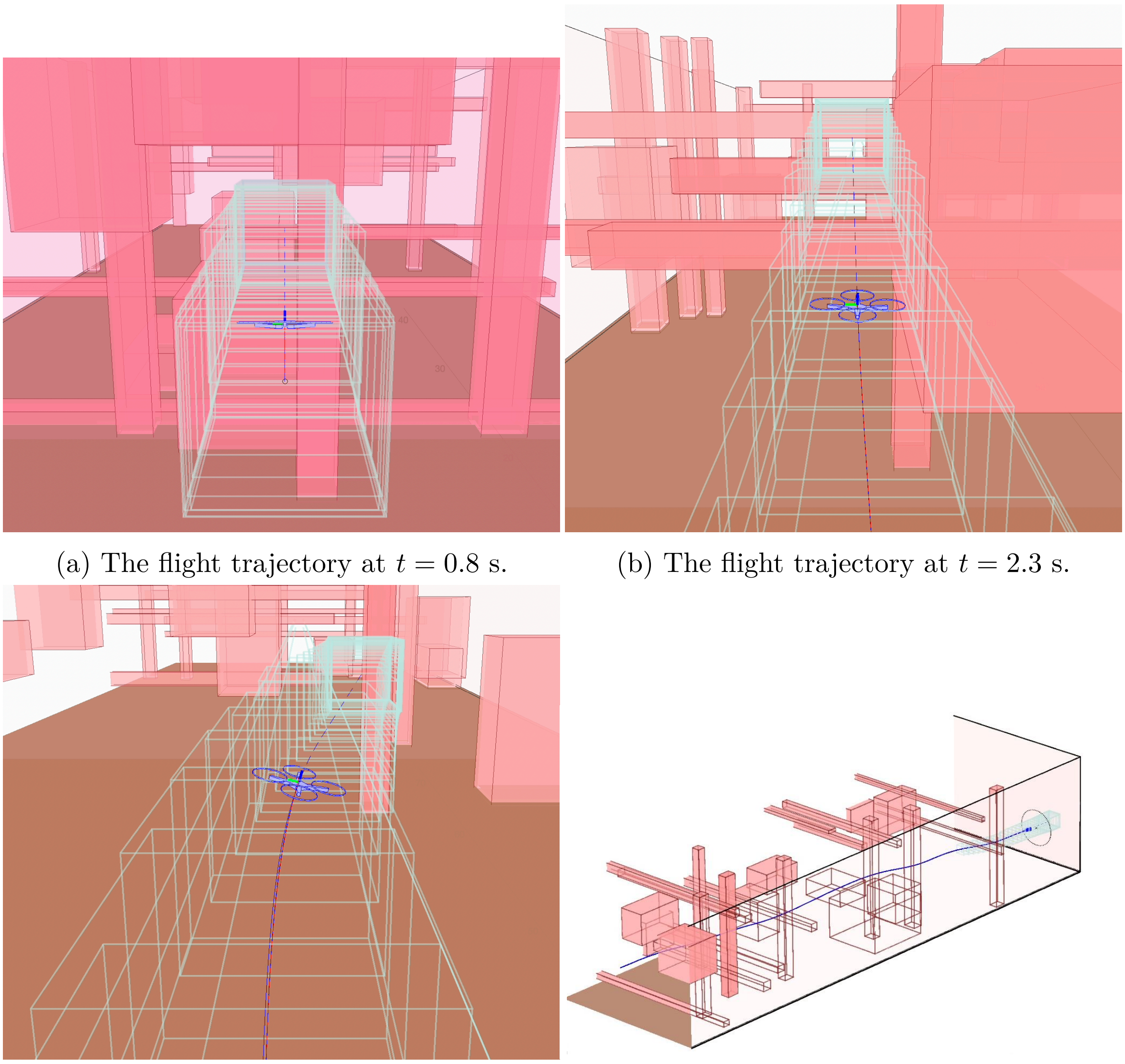
### 3.4.2 Safe Flight of Quadrotor

This subsection numerically validates the generality of our proposed SP-PGC scheme under one safety-critical task of a 6-DoF quadrotor. The quadrotor is required to safely execute in an obstacle-filled environment and finally reach the target position, see Fig. 3.4.

To realize this goal, we firstly use the reachable set based planning algorithm [69] to generate a collision-free desired trajectory $p_d := [x_d, y_d, z_d] \in \mathbb{R}^3$ inside a tube considering the distance from the quadrotor center of mass to the rotor center $d_q = 0.27$ m. Thereby, Assumption 3.1 is satisfied. Then, we follow the procedures illustrated in Section 3.2 and Section 3.3 to design an online performance-guaranteed position tracking controller (3.22) to ensure that the quadrotor always flies in the planned safe tube. The incremental dynamic inversion method (a reformulation of the dynamic inversion method [80] based on the incremental system formulated in Section 3.2) is adopted to design the attitude controller given that we have no specific performance requirements for the attitude control.

Denoting the boundary of the reachable set as $b(t) \in \mathbb{R}^3$. Given that the desired trajectory points $p_d(t)$ are on the center line of the reachable sets, the allowable control-level tracking error (performance bound) to ensure safety follows $k(t) = b(t) - p_d(t) - d_q$. Note that rather than using this varying $k(t)$ to construct ISS-PS-BLFs, we use the minimum value of $k(t)$ ($\underline{\epsilon} = [-0.05, -0.05, -0.05]^\top$, and $\overline{\epsilon} = [0.05, 0.05, 0.05]^\top$ in particular) to exemplify the realizable high-accuracy tracking performance of our designed control strategy (3.22). The simulation parameters for the position controller are set as: $p_0(t) = [1, 0, 5]^\top$, $\bar{g} = \mathrm{diag}(6.5, 6.5, 6.5)$, $L_1 = \mathrm{diag}(0.25, 0.25, 0.25)$, and $L_2 = \mathrm{diag}(8, 8, 8)$.

We validate the effectiveness of our approach in six randomly generated environments. The associated videos are referred to `https://youtu.be/VKlaqWJBxus`. The interactions between the quadrotor and the environment at specific time instants are displayed in Fig. 3.4. Our designed controller enables the quadrotor to always fly inside the safe tunnels (see Fig. 3.4a-3.4c) and finally reach the target position (see Fig. 3.4d).



(a) The flight trajectory at $t = 0.8$ s.

(b) The flight trajectory at $t = 2.3$ s.

(c) The flight trajectory at $t = 6.9$ s.

(d) The global view of flight trajectory.

Figure 3.4: The illustration of the safe execution of quadrotor in one safety-critical environment (red line: planned trajectory; blue line: real trajectory).

## 3.5 Experimental Validation

This section experimentally validates the robustness enhancement brought by the dynamics free property of our method via the task-space tracking task of the 3-DoF robot manipulator (see Fig. 3.5) in the Chair of Automatic Control Engineering (LSR), Technical University of Munich (TUM). More details about the hardware are referred to in our previous work [48].

An industrial welding and cutting task is considered here. We choose $\underline{\epsilon} = [-0.01, -0.01]^\top$, $\overline{\epsilon} = [0.01, 0.01]^\top$ for our designed performance-guaranteed tracking controller (3.22) to drive the robot manipulator end-effector to realize precision machining. To ensure robust safety, the above determined performance bounds ($\underline{\epsilon}$ and $\overline{\epsilon}$ in particular) are considered in the planning level to inflate obstacles by a $\epsilon = 0.01$ margin before generating circle reference signals $p_d := [x_d, y_d] \in \mathbb{R}^2$ (encoding the industrial task). The remaining parameters required

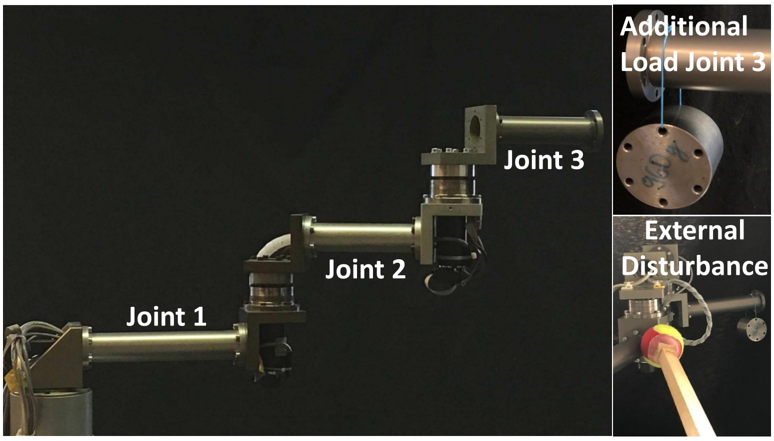
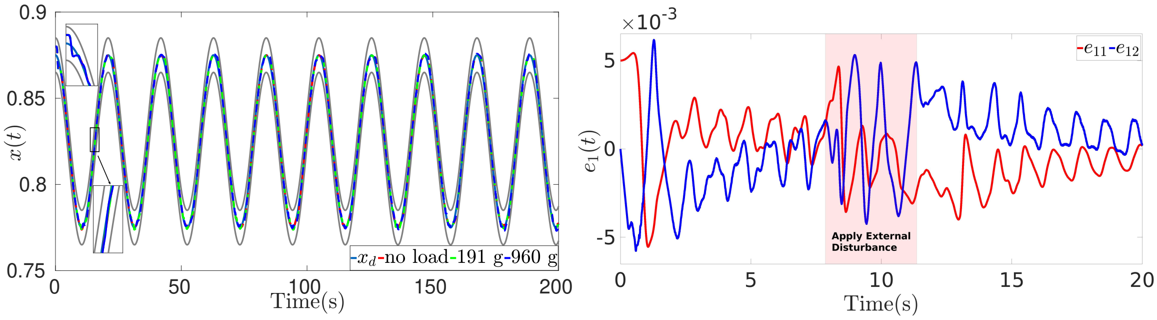


Figure 3.5: The 3-DoF robot manipulator, load and external torque.



(a) Trajectories of $x$, $x_d$, safe bound.

(b) Tracking error $e_1$ (960 g case).

Figure 3.6: The experimental validation of the SP-PGC scheme under different loads and external disturbances.

for the tracking controller (3.22) are set as $\bar{g} = \begin{bmatrix} 120 & 0 & 0 \\ 0 & 120 & 80 \end{bmatrix}$, and $L_1 = \text{diag}(0.1, 0.1)$, $L_2 = \text{diag}(20, 20)$. The initial position is set as $x_1(0) = [0.88, 0]^\top$, $x_2(0) = [0, 0]^\top$, $u = [0, 0, 0]^\top$. The sampling rate is 1kHz.

The Cartesian-space trajectory $x(t)$ displayed in Fig. 3.6a (the trajectory $y(t)$ is similar to Fig. 3.6a, thus omitted for page limit) show that the robot manipulator driven by our developed control strategy (3.22) efficiently tracks the desired trajectory precisely without crossing the predefined safe boundary even under different loads. To further demonstrate the robustness property of our method, we use one stick to apply additional torque to the robot manipulator. As shown in Fig.3.6b, the trajectories of the tracking error $e_1(t)$ firstly oscillate due to the external disturbance and then converge to a small value around zero.

## 3.6 Summary

This work realizes the safe control under uncertainty via our formulated ISS-PS-BLF and incremental system. The utilized time-delayed data reformulates kinematic and dynamic uncertainties as well as environmental disturbances into a provably bounded estimation error, which allows us to rigorously analyze the robustness of safety via an input-to-state stable approach. The safe planning algorithm and the ISS-PS-BLF facilitated tracking controller work together to ensure that autonomous systems realize the safe execution under uncertainty with guaranteed performance. Experimental and numerical validations are conducted to show the efficiency of our proposed SP-PGC scheme.

The time-delayed data informed incremental dynamics serve as one easily implement approach to realize the model-free control. The additional consideration of the control-level performance at the planning level realizes the practical safe operation under uncertainty from a systematic perspective. The proposed SP-GPC scheme requires the initial states of autonomous systems belonging to the safe region. However, this requirement might not be satisfied in practice. The performance bound needs to be adjusted automatically to avoid singularity in case the initial states are outside of the safe region. Besides, the influence of noisy measurements on our developed approach remains to be investigated to improve the practicability of the proposed method . To show the superiority of the method, future works aim to extend the kinematics free and dynamics free control strategy to soft robot manipulators.

# Online Learned Instantaneous Local Control Barrier Function for Collision Avoidance

# 4

The previous Chapter 2 and Chapter 3 follow a decoupled mapping, planning and tracking control paradigm to realise safe execution under uncertainties. This decoupled paradigm forgoes theoretical guarantees for efficient and practical applications. Specifically, each level in the above decoupled paradigm is designed assuming perfect operations of its connected levels. However, it is often the case that the actual performance of each level in real-word environments deviates from the desired one. These gaps (deviations) between different levels mentioned above are difficult to be systematically characterized and addressed. Besides, the decoupled paradigm is computationally intensive and poses high hardware requirements for efficient operation of each level.

Towards the above analyzed deficiencies of the decoupled approach, this chapter proposes an integrated perception and control approach to achieve safe execution in unforeseen environments and accomplish given tasks. The integrated approach avoids gaps among levels and utilizes control-theoretical tools to design feedback control strategies with theoretical guarantees of safety (collision avoidance) and task fulfilment (convergence to goal positions). We exploit instantaneous local sensory data in the control-level to stimulate safe feedback control strategies in prior unknown environments, rather than firstly conduct a computationally intensive mapping process and then planning on the constructed map. The organization of this chapter is as follows. Section 4.1 presents the preliminaries and the problem formulation. Then, instantaneous local control barrier functions (IL-CBFs) and goal-driven control Lyapunov functions (GD-CLFs) constraints are learned from sensory data to encode safety and task requirements, which are clarified in Section 4.2 and Section 4.3, respectively. Thereafter, the learned IL-CBFs and GD-CLFs are united through QP in Section 4.4. Moreover, an optimization over the volume of the shared control space among IL-CBFs, GD-CLFs, and input constraints is developed in Section 4.5 to improve the QP feasibility. The safe feedback control strategy is numerically validated in Section 4.6. Finally, Section 4.7 summarizes this chapter.

## 4.1 Problem Formulation and Preliminaries

### 4.1.1 Problem Formulation

This work investigates the safe operation problem of a mobile robot in previously unforeseen environments. We model the investigated mobile robot as

$$\underbrace{\begin{bmatrix}\dot{p}\\ \dot{v}\end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix}0_{2\times2} & I_{2\times2}\\ 0_{2\times2} & 0_{2\times2}\end{bmatrix}\begin{bmatrix}p\\ v\end{bmatrix}}_{f(x)} + \underbrace{\begin{bmatrix}0_{2\times2}\\ I_{2\times2}\end{bmatrix}}_{g(x)} u, \tag{4.1}$$

where $p := [p_x, p_y]^\top$, $v := [v_x, v_y]^\top$, and $u := [u_x, u_y]^\top \in \mathbb{R}^2$ are the positions, velocities, and control inputs, respectively. For simplicity, we assume that the robot localization is perfect, i.e., the accurate vehicle state is available. The localization is realizable by the low-cost dead reckoning method. Dealing with its cumulative error is a different research direction, which is beyond the scope of this chapter.

Assume that there exist multiple prior unknown obstacles $\mathcal{O}_l$ in an environment $\mathcal{E}$, where $l \in \mathcal{L} := \{l | l = 1, 2, \cdots, L\}$ and $L \in \mathbb{N}^+$ is an uncertain value. The objective is to design a feedback controller $u$ to drive the mobile robot (4.1) to operate safely in an uncertain environment $\mathcal{E}$ and finally reach the predetermined target position $p_d := [p_{d_x}, p_{d_y}]^\top \in \mathbb{R}^2$. We formulate the safe operation problem mentioned above as a constrained optimization problem stated as

$$\min_u J := \int_{t_0}^{t_f} u^\top u \, dt \tag{4.2a}$$

$$\text{s.t. } (4.1)$$

$$p(t_0) = p_0; v(t_0) = v_0 \tag{4.2b}$$

$$u(t) \in \mathcal{U}, \forall t \in [t_0, t_f] \tag{4.2c}$$

$$p(t) \cap \bigcup_{l=1}^{L} \mathcal{O}_l = \emptyset, \forall t \in [t_0, t_f] \tag{4.2d}$$

$$\|p(t_f) - p_d\| \leq \delta. \tag{4.2e}$$

where $\mathcal{U} \subseteq \mathbb{R}^2$ in (4.2c) denotes the bounded input space of the considered dynamics (4.1). $\delta \in \mathbb{R}^+$ in (4.2e) is a prior set threshold to check whether the reach task is completed. A quadratic control energy function is adopted in (4.2a) to reflect designers' preference on the control effort minimization.

The aforementioned safe operation problem (4.2) is nontrivial given the constraints indicating different (might conflicting) objectives of safety and performance maximization; and the requirement of constraint satisfaction under uncertainty (limited knowledge of the environment $\mathcal{E}$). This work seeks for an integrated perception and control approach to solve (4.2), whose mechanism is illustrated in Figure 4.1. In particular, we directly use perceptual inputs to learn IL-CBFs and GD-CLFs that are used in the control level to achieve collision avoidance and accomplish given tasks.

### 4.1.2 Preliminaries

Before proceeding to the development of IL-CBFs and GD-CLFs, we first present the definitions of classic High-order CBF (HO-CBF) and CLF focusing on (4.1). The introduced HO-CBF and CLF here serve as theoretical basis to develop our IL-CBF and GD-CLF later.

**Definition 4.1** (HO-CBF). *[81, Definition 1] Given the control system* (4.1)*, a $C^r$ function $h(t,x) \in \mathbb{R}$ with a relative degree $r$ is called a (zeroing) control barrier function (of order $r$) if there exists a column vector $\alpha := [\alpha_1, \cdots, \alpha_r]^\top \in \mathbb{R}^r$ such that $\forall x \in \mathbb{R}^n$, $t \geq 0$,*

$$\sup_{u \in U} \left[ L_g \bar{L}_f^{r-1} h(t,x) u + \bar{L}_f^r h(t,x) + \alpha^\top \xi(t,x) \right] \geq 0, \tag{4.3}$$

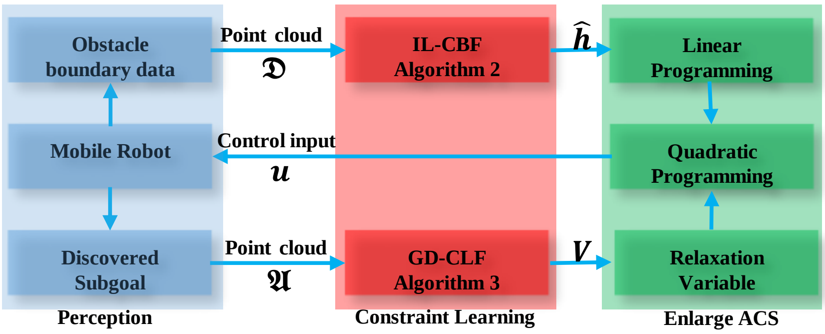


Figure 4.1: Schematic of the integrated approach that maps sensory data to control inputs. The IL-CBFs learned from sensory data in Section 4.2 characterize the obstacle boundaries; The decomposed short-horizon subtasks are encoded by GD-CLFs clarified in Section 4.3; The LP optimization is conducted to enlarge the ACSs in Section 4.5 to improve the feasibility of the QP formulated in Section 4.4.

*where* $\bar{L}_f^r h := \left(\frac{\partial}{\partial t} + L_f\right)^r h$ *is the modified Lie derivative of* $h(t,x)$ *along* $f$ *and* $r \in \mathbb{N}^+$*, and the roots of the polynomial*

$$\mathcal{P}^r(\lambda) := \lambda^r + \alpha_1 \lambda^{r-1} + \cdots + \alpha_{r-1}\lambda + \alpha_r, \tag{4.4}$$

*are all negative.*

**Definition 4.2** (CLF)**.** *[82, Definition 1] For the control system* (4.1)*, a continuously differential function* $V(x) \in \mathbb{R}$ *is an exponentially stabilizing control Lyapunov function if there exists* $c_1$*,* $c_2$*,* $c_3 \in \mathbb{R}^+$ *such that the following equations hold*

$$c_1 \|x\|^2 \leq V(x) \leq c_2 \|x\|^2 \tag{4.5a}$$

$$\inf_{u \in \mathbb{R}^m} \left[L_f V(x) + L_g V(x) u + c_3 V(x)\right] \leq 0. \tag{4.5b}$$

## 4.2 IL-CBF Online Learning

This section elucidates the mechanism of learning IL-CBFs from sensory data. In particular, the detected local obstacle information is utilized to learn the local barrier functions to describe the partial obstacle boundaries; and the learned local barrier functions update along with continuously coming data to tackle the uncertain environment. Our developed IL-CBFs are employed to formulate the QP problem in Section 4.4 to conduct collision avoidance in the control level with prior-unforeseen obstacles.

As illustrated in Figure 4.2, the whole boundaries of the obstacles $\mathcal{O}_l$ in $\mathcal{E}$ could be descried by the barrier functions $h_l(p) \in \mathbb{R}$ using the complete knowledge of obstacles [2]. However, the obstacle information is unavailable in our investigated problem (4.2). Thus, the explicit forms of $h_l(p)$ that characterize the dangerous region $\bigcup_{l=1}^{L} \mathcal{O}_l$ are unavailable. We observe in Figure 4.2 that only partial obstacle boundaries of $\mathcal{O}_l$ pose threats to the mobile robot safety at certain period. This motivates us to utilize local sensory data to learn the local barrier functions, corresponding to the partial obstacle boundary within the mobile robot's sensor horizon, to address the collision avoidance problem.

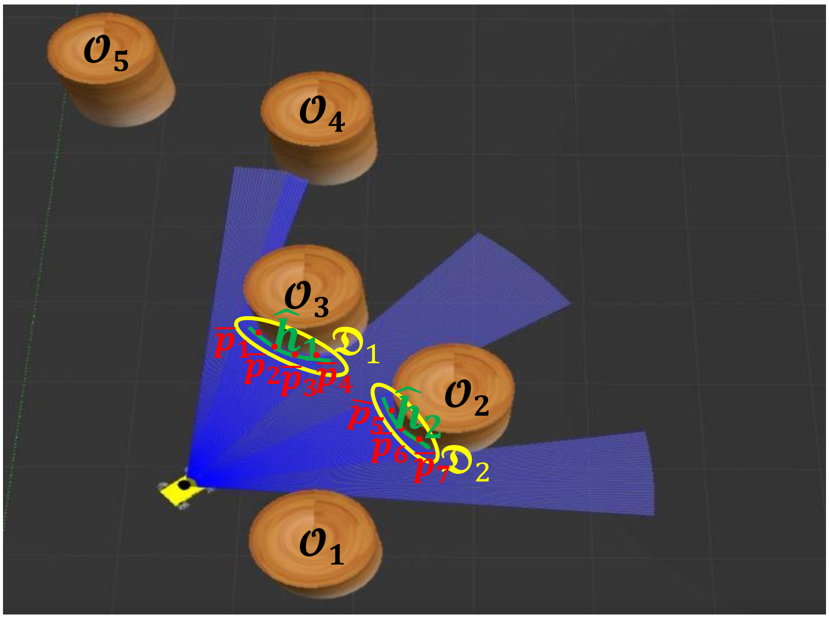


Figure 4.2: Graphical illustration of IL-CBFs and obstacles. The whole boundary of obstacles $\mathcal{O}_l$ are described by explicit CBFs $h_l = (x - x_{o_l})^2 + (y - y_{o_l})^2 - r_l^2$, $c_l = (x_{o_l}, y_{o_l})$, $l = 1, \cdots, 5$. The mobile robot observes $\mathfrak{D} = \{\bar{p}_1, \bar{p}_2, \cdots, \bar{p}_7\}$ and classifies $\mathfrak{D}$ into sub-groups $\mathfrak{D}_k$, $k = 1, 2$. Thus, $K = 2$, and $I_1 = 4$, $I_2 = 3$ here. The mobile robot learns $\hat{h}_k$ based on $\mathfrak{D}_k$.

Assume that the mobile robot is embedded with a sensor with a restricted angle $S_\theta$ and a limited horizon $S_r$. The value of $S_\theta$ is given, and the value of $S_r$ satisfies

$$S_r \geq D_{\text{brake}} := \|v_{\max}\|^2 / \|a_{\max}\|, \tag{4.6}$$

where $v_{\max}$, $a_{\max} \in \mathbb{R}^2$ are the maximum velocity and breaking acceleration of the mobile robot (4.1). $D_{\text{brake}}$ denotes the travelled distance when the mobile robot in the maximum velocity brakes using the maximum breaking acceleration.

**Remark 4.1.** *The setting of the sensor horizon* (4.6) *is beneficial to the emergence case where our developed safe feedback control strategy fails to guarantee safety. In this scenario, the mobile robot brakes to avoid collisions.*

The sensor provides a point cloud $\mathfrak{L}$. We term $\mathfrak{D} := \{\bar{p}_1, \bar{p}_2, \cdots\} \subset \mathfrak{L}$ as the data group of the sensed obstacle boundaries, wherein $\bar{p}_i := [\bar{x}_i, \bar{y}_i]^\top \in \mathbb{R}^2$ is the position of the $i$-th detected obstacle boundary point. In an environment $\mathcal{E}$ with densely populated obstacles, data points in $\mathfrak{D}$ might concern multiple isolated obstacles, as displayed in Figure 4.2. Therefore, we adopt the robust classifying algorithm–*density-based spatial clustering of applications with noise (DBSCAN)* [83]–to classify $\mathfrak{D}$ into multiple sub-groups $\mathfrak{D}_k := \{\bar{p}_{k_1}, \bar{p}_{k_2}, \cdots\}$, wherein $\bar{p}_{k_i} := [\bar{x}_{k_i}, \bar{y}_{k_i}]^\top \in \mathbb{R}^2$ denotes the $i$-th data point of the $k$-th data group, $i \in \mathcal{I} := \{i | i = 1, \cdots, I_k\}$ with $I_k \in \mathbb{N}^+$ being the volume of $\mathfrak{D}_k$, and $k \in \mathcal{K} := \{k | k = 1, \cdots, K\}$ with $K \in \mathbb{N}^+$ being the sum of the local obstacle boundary considered in current period.

**Remark 4.2.** *The* DBSCAN *algorithm is compatible with our IL-CBF learning process given that it could determine the number of to be learned IL-CBFs (i.e., the values of $K$) automatically without using prior knowledge of environments.*

**Algorithm 2** IL-CBF Online Learning Algorithm

```
Input: Point cloud $\mathfrak{D}$;
Output: $\hat{h}_k, k = 1, \cdots, K$;
K = DBSCAN ($\mathfrak{D}$)                                  ▷ Robust classifying
for $k = 1 : K$ do
    $\hat{\zeta}_k$ = M-estimate ($\mathfrak{D}_k$) (4.9)    ▷ Robust regression
    $\hat{h}_k = y - \mathcal{F}(x, \hat{\zeta}_k)$ (4.10)
end for
```

In the following, we clarify the mechanism of the IL-CBF learning focusing on the $k$-th data group $\mathfrak{D}_k$. Assume that $i$-th data pair $\bar{p}_{k_i}$ satisfies

$$\bar{y}_{k_i} = \mathcal{F}(\bar{x}_{k_i}, \zeta_k) + \varepsilon_k, \tag{4.7}$$

where $\mathcal{F}(\bar{x}_{k_i}, \zeta_k) \in \mathbb{R}$ is one $n$-th degree polynomial function with a parameter $\zeta_k \in \mathbb{R}^{n+1}$ to be learned; and $\varepsilon_k \sim N(0, \sigma^2)$ denotes an assumed Gaussian sensor noise with a zero mean and a constant variance $\sigma \in \mathbb{R}$.

**Remark 4.3.** *There exist multiple choices for $\mathcal{F}$, such as Gaussian models, linear fitting, and rational polynomials [84]. Considering the generality and simplicity issues, a polynomial model is chosen here.*

Based on (4.7) and the point cloud $\mathfrak{D}_k$ from the sensor, $\zeta_k$ is learned to minimize the approximation error:

$$\hat{\zeta}_k = \arg\min_{\zeta_k} \sum_{i=1}^{I_k} \left(\bar{y}_{k_i} - \mathcal{F}(\bar{x}_{k_i}, \zeta_k)\right)^2. \tag{4.8}$$

To address potential noises and outliers that exist in the measurement data, the robust regression technique–*M-estimate* [85]–is adopted here. By using the *M-estimate*, the learning of $\zeta_k$ in (4.8) is rewritten as

$$\hat{\zeta}_k = \arg\min_{\zeta_k} \sum_{i=1}^{I_k} \rho\left(\frac{\bar{y}_{k_i} - \mathcal{F}(\bar{x}_{k_i}, \zeta_k)}{\gamma}\right), \tag{4.9}$$

where $\rho(r) = c^2/(1 - (1 - (r/c)^2)^3)$ is a robust loss function with $c = 1.345$; $\gamma$ is a scale parameter estimated as $\gamma = 1.48\,[\mathrm{med}_i\,|(\bar{y}_{k_i} - \mathcal{F}(\bar{x}_{k_i}, \zeta_{k0})) - \mathrm{med}_i\,(\bar{y}_{k_i} - \mathcal{F}(\bar{x}_{k_i}, \zeta_{k0}))|]$, $\zeta_{k0}$ is the initial value of $\zeta_k$. Details about the *M-estimate* approach are referred to [85].

Using the learned $\hat{\zeta}_k$ (4.9), we construct the IL-CBF $\hat{h}_k$ as

$$\hat{h}_k = y - \mathcal{F}(x, \hat{\zeta}_k). \tag{4.10}$$

The IL-CBF learning process mentioned above is summarized in Algorithm 2. The mobile robot uses Algorithm 2 to update the learned IL-CBFs continuously based on the newly observed sensory data during the operation process. The IL-CBF learning is favored with computation simplicity. Thus, it is practical to update the learned IL-CBFs each step. This is favourable for the mobile robot to adapt to diverse environments.

**Algorithm 3** GD-CLF Online Learning Algorithm

```
Input: Point cloud $\mathfrak{A} := \{\tilde{p}_1, \tilde{p}_2, \cdots\}$; Robot position $p$.
Output: $\tilde{p}_{d_j}$, and $V_j$, $j = 1, \cdots, J$;
1: $\tilde{p}_{d_1} = \arg\min_{\tilde{p}_i \in \mathfrak{A}} \|\tilde{p}_i - p_d\|$ and get $V_1$ (4.11)
2: if $\left\|p - \tilde{p}_{d_j}\right\| \leq \delta$ then
3:     $\tilde{p}_{d_j} = \arg\min_{\tilde{p}_i \in \mathfrak{A}} \|\tilde{p}_i - p_d\|$
4:     $j = j + 1$ and update $V_j$ (4.11)
5: end if
```

**Remark 4.4.** *Alternatively, we are able to achieve the CBF learning in an incremental way along with a steady stream of data, i.e., attempting to gradually learn one global barrier function that describes the whole obstacle boundary. However, authors found in practice that this increment learning approach shows no obvious advantage in terms of collision avoidance but introduces additional computational loads. Thus,we forgo using all detected data to gradually build a perfect map, rather only using instantaneous local sensory information.*

**Remark 4.5.** *The clarified IL-CBF learning in this section is especially compatible with low-end sensors that only provide low-dimensional data. These limited data, however, is not enough to build a global map or describe the whole obstacle boundary.*

## 4.3 GD-CLF Automatic Construction

The data group $\mathfrak{D}$ concerning the detected obstacle boundaries is utilized in Section 4.2 to facilitate the collision avoidance in uncertain environments. This section exploits the remaining local collision-free sensory data group $\mathfrak{A} := \mathfrak{L} \ominus \mathfrak{D}$ to complete the long-horizon task. Specifically, we first utilize the data group $\mathfrak{A}$ to discover subgoals using a Euclidean distance metric. Then, we construct the associated GD-CLF for each subtask (subgoal). The automatically constructed GD-CLFs serve as constraints of the QP optimization in Section 4.4, whose solution ensures that the mobile robot travels toward the discovered subgoals incrementally and reach the destination finally.

Normally, the common CLF in Definition 4.2 is inefficient to account for a long-horizon goal. Thus, through a divide-and-conquer perspective, we use sensory data $\mathfrak{A}$ to discover the subgoals $\tilde{p}_{d_j} := [x_{d_j}, y_{d_j}]^\top \in \mathbb{R}^2$, $j \in \mathcal{J} := \{j | j = 1, \cdots, J\}$ with $J \in \mathbb{N}^+$, based on a Euclidean distance metric (line 3 of Algorithm 3). In particular, we choose the nearest collision-free waypoint toward the goal position $p_d$ as the next subgoal . These automatically determined intermediate waypoints (such as $\tilde{p}_{d_1}$, $\tilde{p}_{d_2}$, and $\tilde{p}_{d_3}$ in Figure 4.3) forwardly progress toward the final desired position $p_d$ (same with $\tilde{p}_{d_4}$).

The automatically determined subgoals $\tilde{p}_{d_j}$ from Algorithm 3 divide the long-horizon task into $J$ short-horizon subtasks. For each subtask, we construct the GD-CLF :

$$V_j = (p - \tilde{p}_{d_j})^\top P (p - \tilde{p}_{d_j}) + (v - v_{dj})^\top Q (v - v_{dj}), j \in \mathcal{J} \tag{4.11}$$

where $P$, $Q \in \mathbb{R}^{2\times 2}$ are predetermined positive definite matrices; and $v_{d_j} \in \mathbb{R}^2$ could be a zero or a prior-given constant velocity vector. The constructed GD-CLF $V_j$ (4.11) updates as the subgoal $\tilde{p}_{d_j}$ refreshes using Algorithm 3.

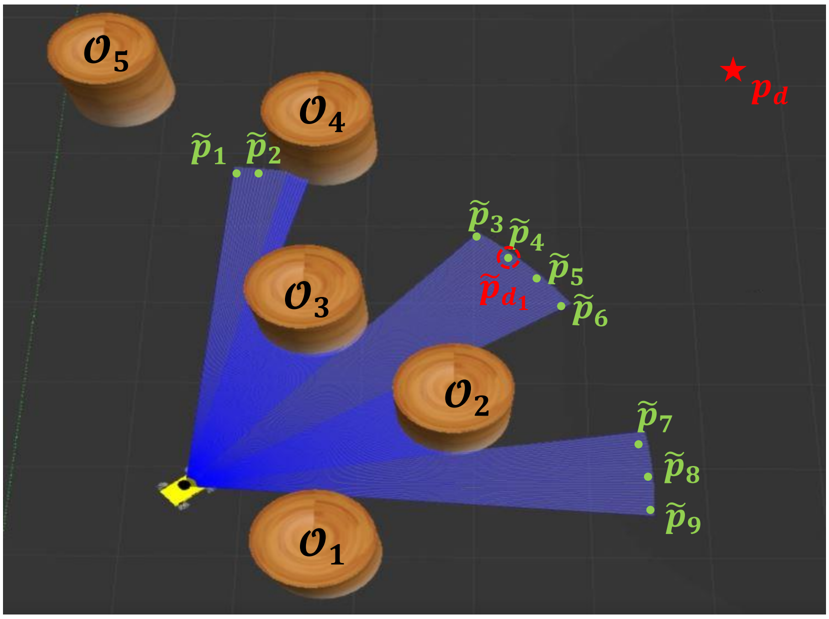


Figure 4.3: Graphical illustration of GD-CLFs and subgoals. The mobile robot uses the collision-free data group $\mathfrak{A} = \{\tilde{p}_1, \tilde{p}_2, \cdots, \tilde{p}_9\}$ and Algorithm 3 to determine the position $\tilde{p}_4 \in \mathfrak{A}$ as its first subgoal $\tilde{p}_{d_1}$. Then, the constructed GD-CLF $V_1$ guides the robot toward $\tilde{p}_{d_1}$. The robot would determine its $j+1$-th subgoal when it arrives at $\delta$-neighboured (descried as a green dotted circle with radius $\delta$) around the $j$-th subgoal .

**Remark 4.6.** *Note that we construct the IL-CBFs* (4.10) *in Section 4.2 and the GD-CLFs* (4.11) *in Section 4.3 assuming that $\mathcal{U} = \mathbb{R}^2$ for convenience, i.e., the influence of input saturation is ignored temporally. This problem is later tackled in Section 4.5 by explicitly analysing the potential conflicts between IL-CBFs, GD-CLFs, and input constraints.*

## 4.4 Safe Feedback Control Strategy

This section incorporates the learned IL-CBFs (4.10) and the constructed GD-CLFs (4.11) in a QP optimization to generate the safe feedback control strategy that drives the mobile robot to safely reach the target position incrementally.

By dividing the period $[t_0, t_f]$ into multiple intervals $[t_0 + mT, t_0 + (m+1)T]$ [86], where $m \in \mathbb{N}^+$, and $T \in \mathbb{R}^+$ is the sampling time, we reformulate the original safe operation problem (4.2) into a sequence of QPs at each interval:

$$\min_{u,\nu} \; u(t)^\top u(t) + \bar{c}_1 \nu^2(t) \tag{4.12a}$$

$$\text{s.t. } (4.1),\; (4.2\text{b}),\; (4.2\text{c})$$

$$\ddot{\hat{h}}_k + \alpha_{k_1}\dot{\hat{h}}_k + \alpha_{k_2}\hat{h}_k \geq 0,\;\; k \in \mathcal{K} \tag{4.12b}$$

$$\dot{V}_j + \bar{c}_2 V_j \leq \nu,\;\; j \in \mathcal{J}, \tag{4.12c}$$

where $\nu(t) \in \mathbb{R}$ is a relaxation variable to relax the GD-CLF constraint to improve the QP feasibility [87]; $\alpha_{k_1}$, $\alpha_{k_2}$, $\bar{c}_1$, $\bar{c}_2$ $\in \mathbb{R}$ are parameters to be determined. The reformulated

QP problem (4.12) unifies safety requirement (4.2c), (4.12b), task requirements (4.12c), and optimization over control efforts (4.12a) to generate a multi-objective feedback controller that drives the mobile robot to progressively reach subgoals while avoiding obstacles. Note that our developed safe feedback control strategy from (4.12) only requires the information of the mobile robot position $p$ and the target position $p_d$ to solve the safe operation problem (4.2) in uncertain environments.

## 4.5 Optimized Admissible Control Space

The potential conflicts between the constraints (4.2c), (4.12b), and (4.12c) might result in the infeasibility problem of the QP (4.12) formulated in Section 4.4. This section formulates an optimization over the ACS of the IL-CBF associated constraint (4.12b) to improve the QP feasibility.

For analytical convenience, we denote the ACSs for constraints (4.12b) and (4.12c) as $\mathcal{A}_1 := \left\{u \in \mathbb{R}^2 | \ddot{\hat{h}}_k + \alpha_{k_1}\dot{\hat{h}}_k + \alpha_{k_2}\hat{h}_k \geq 0, k \in \mathcal{K}\right\}$, and $\mathcal{A}_2 := \left\{u \in \mathbb{R}^2 | \dot{V}_j + c_2 V_j \leq \nu\right\}$, respectively. Thereby, the shared control space concerning constraints (4.2c), (4.12b), and (4.12c) would be $\mathcal{S} = \mathcal{A}_1 \cap \mathcal{A}_2 \cap \mathcal{U}$. It is desirable that $\mathcal{S} \neq \emptyset$ always holds, i.e., the feasibility of the QP problem is always guaranteed. This is a nontrivial problem; especially multiple constraints are considered. Improving the possibility of satisfying $\mathcal{S} \neq \emptyset$ is equivalent to enlarge the volume of $\mathcal{S}$. Given that the relationship between sets $\mathcal{A}_1$ and $\mathcal{A}_2$ is hard to be described and the volume of $\mathcal{U}$ is predetermined, we could transform the enlargement of the volume of $\mathcal{S}$ into the enlargement of the volumes of ACSs $\mathcal{A}_1$ and $\mathcal{A}_2$ independently. A relaxation variable $\nu$ has been used in (4.12c) to enlarge the volume of $\mathcal{A}_2$. In the following, we attempt to enlarge the volume of the ACS $\mathcal{A}_1$ to improve the feasibility of the QP problem (4.12). In particular, we firstly seek for a criterion for the volume of the ACS $\mathcal{A}_1$ in Section 4.5.1 by investigating the relationship between sets $\mathcal{A}_1$ and $\mathcal{U}$. Then, a linear programming (LP) optimization problem is formulated in Section 4.5.2 to optimize the volume criterion found above to enlarge the volume of the ACS $\mathcal{A}_1$.

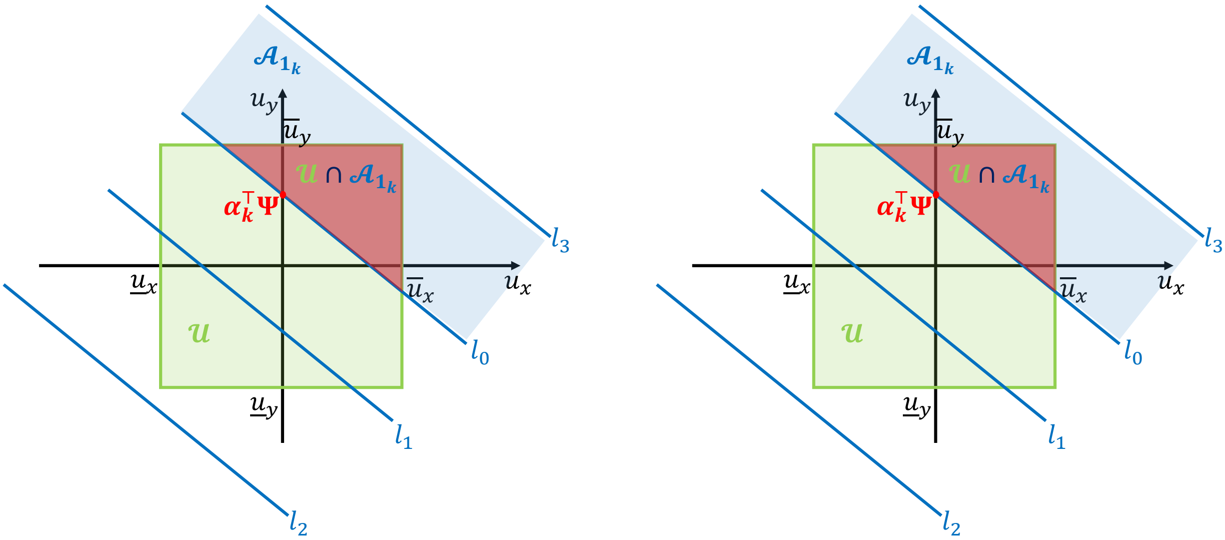
### 4.5.1 Criterion of ACS

The enlargement of $\mathcal{A}_1$ is equivalent to enlarge each IL-CBF $\hat{h}_k$ associated ACS that is denoted as $\mathcal{A}_{1_k} := \left\{u \in \mathbb{R}^2 | \ddot{\hat{h}}_k + \alpha_{k_1}\dot{\hat{h}}_k + \alpha_{k_2}\hat{h}_k \geq 0\right\}$, $k \in \mathcal{K}$. The explicit form of the learned $k$-th IL-CBF follows $\hat{h}_k = y - \hat{\zeta}_k^\top \Phi$, where $\Phi = [1, x, x^2, \cdots, x^n]$. We substitute the explicit $\hat{h}_k$ into (4.12b) and rewrite the inequality as

$$A u_x + u_y + a_k^\top \Psi > 0, \tag{4.13}$$

where $A = \hat{\zeta}_k^\top \frac{\partial \Phi}{\partial x} \in \mathbb{R}$, $\alpha_k = [\alpha_{k_1}, \alpha_{k_2}]^\top \in \mathbb{R}^2$, $\Psi = \left[\hat{\zeta}_k^\top \frac{\partial \Phi}{\partial x} v_x - v_y, \hat{\zeta}_k^\top \frac{\partial^2 \Phi}{\partial x^2} v_x^2 + \hat{\zeta}_k^\top \Psi - y\right]^\top \in \mathbb{R}^2$.

Based on the reformulated (4.13), the geometric interpretations of the ACS $\mathcal{A}_{1_k}$ as well as the limited control input set $\mathcal{U}$ are depicted in Figure 4.4. We found that a smaller value of $a_k^\top \Psi$ implies a larger area of the ACS $\mathcal{A}_{1_k}$. Thus, it is reasonable to choose the value of $a_k^\top \Psi$ as a metric to quantify the volume of the ACS $\mathcal{A}_{1_k}$, which is optimized in the subsequent subsection.



(a) $A > 0$ case.

(b) $A < 0$ case

Figure 4.4: The geometric interpretation of the sets $\mathcal{A}_{1_k}$ and $\mathcal{U}$. Here $l_k = Au_x + u_y + a_k^\top \Psi = 0$. The comparison of the volume of $\mathcal{A}_{1_k}$ follows $\mathcal{A}_{1_k}^{l_2} > \mathcal{A}_{1_k}^{l_1} > \mathcal{A}_{1_k}^{l_0} > \mathcal{A}_{1_k}^{l_3}$ for two cases. For the $l_3$ case, $\mathcal{A}_{1_k} \cap \mathcal{U} = \emptyset$, i.e., there is no feasible control input to ensure safety based on the current chosen IL-CBF.

### 4.5.2 Optimization of ACS

This subsection clarifies the optimization over the metric $a_k^\top \Psi$, which is formulated as a LP:

$$\min_{\alpha_k} \quad \alpha_k^\top \Psi \tag{4.14a}$$

$$\text{s.t. } 0 < \alpha_{k_1}, \alpha_{k_2} < \overline{\alpha}_k \tag{4.14b}$$

$$a_{k1}^2 - 4\alpha_{k_2} \geq 0 \tag{4.14c}$$

where $\overline{\alpha}_k \in \mathbb{R}^+$ is the predetermined bound for the optimization variable. The formulated LP (4.14) is solved by the off-the-self *fmincon* solver. The core idea of the above LP is to select suitable values of $\alpha_{k_1}$ and $\alpha_{k_2}$ to minimize $\alpha_k^\top \Psi$ while respecting constraints (4.14b) and (4.14c). A decreased $\alpha_k^\top \Psi$ leads to a enlarged $\mathcal{A}_{1_k}$. Thereby, the QP feasibility is improved.

**Remark 4.7.** *The constraints* (4.14b) *and* (4.14c) *are the simplification of the following three constraints: (1)* $a_{k1}^2 - 4\alpha_{k_2} \geq 0$*; (2)* $\frac{-\alpha_{k_1} + \sqrt{a_{k1}^2 - 4\alpha_{k_2}}}{2} < 0$*; (3)* $\frac{-\alpha_{k_1} - \sqrt{a_{k1}^2 - 4\alpha_{k_2}}}{2} < 0$*. These three constraints ensure that the roots of* (4.12b)*'s related polynomials are all negative. These constraints ensure that the optimized parameter* $\alpha_k^*$ *leads to valid HO-CBFs in Definition 4.1.*

## 4.6 Numerical Simulation

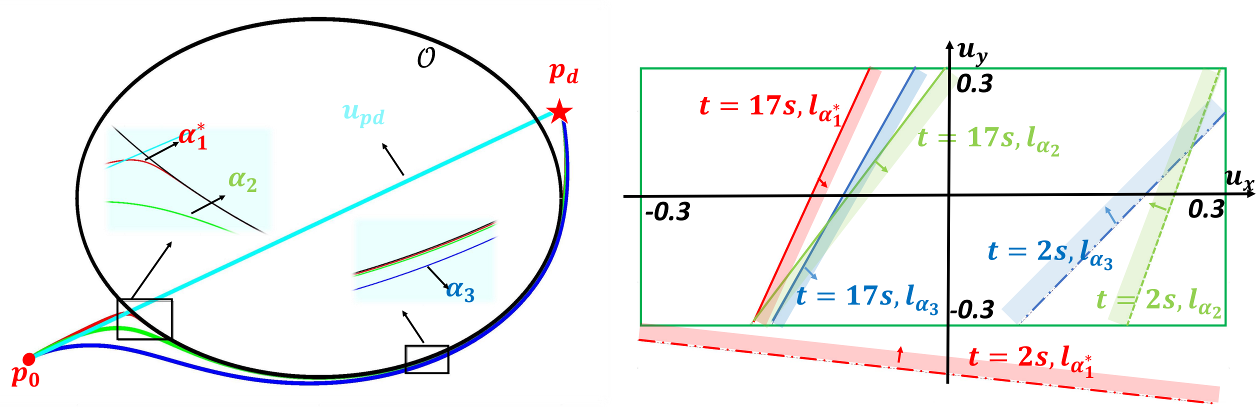
This section conducts numerical simulations to validate the efficiency of our proposed safe feedback control strategy (4.12). In particular, Section 4.6.1 focuses on a benchmark [86] to validate the effectiveness of the LP optimization (4.14). The resulting enlarged ACS leads to a better control performance. Then, we validate the efficiency of our integrated approach under two representative environments: an obstacle-filled outdoor scenario in Section 4.6.2, and a maze indoor scenario in Section 4.6.3. The mobile robot safely operate in the unforeseen outdoor or maze indoor environment and complete the given long-horizon reach task using the safe feedback control strategy, generated by solving the QP (4.12) within consideration of our developed IL-CBF (4.10) and GD-CLF (4.11). During the whole operation process, the QP feasibility is preserved via the LP optimization (4.14).

### 4.6.1 Validation of Optimized ACS

This subsection validates the effectiveness of our developed optimized ACS strategy (4.14) clarified in Section 4.5.2 based on a benchmark reach-avoid task [86]. A mobile robot modelled as (4.1) is desired to move from an initial position $p_0$ to a desired position $p_d$ while avoiding one circle obstacle $\mathcal{O}$ (centered at $c = (1, 1)$ and with radius $r = 1$). The detailed simulation settings are referred to Table 4.1. Note that to avoid IL-CBFs and GD-CLFs' influence on the QP feasibility, this subsection uses a prior-known CBF to achieve collision avoidance, and a well-tuned Proportional–Derivative (PD) controller (assumed with desired performance) to accomplish the reach task. We formulate the following QP (4.15) to solve

Table 4.1: The parameter settings of the reach-avoid task.

| **Initial values** | $p_0 = [-0.2, 0.1]^\top$, $v_0 = [0, 0]^\top$, $T = 10$ Hz |
|---|---|
| **Target values** | $p_d = [2, 1.5]^\top$, $v_d = [0, 0]^\top$ |
| **CBF** | $h = (x-1)^2 + (y-1)^2 - 1$ |
| **PD controller** | $u_{pd} = -0.2(p - p_d) - 0.9(v - v_d)$ |
| **QP and LP** | $\overline{u}_x$, $\overline{u}_y = 0.3$, $\alpha_1(t_0) = [5, 6]^\top$, $\overline{\alpha}_1 = 7$. |



(a) The comparison regarding $p(t)$.

(b) The comparison regarding ACS.

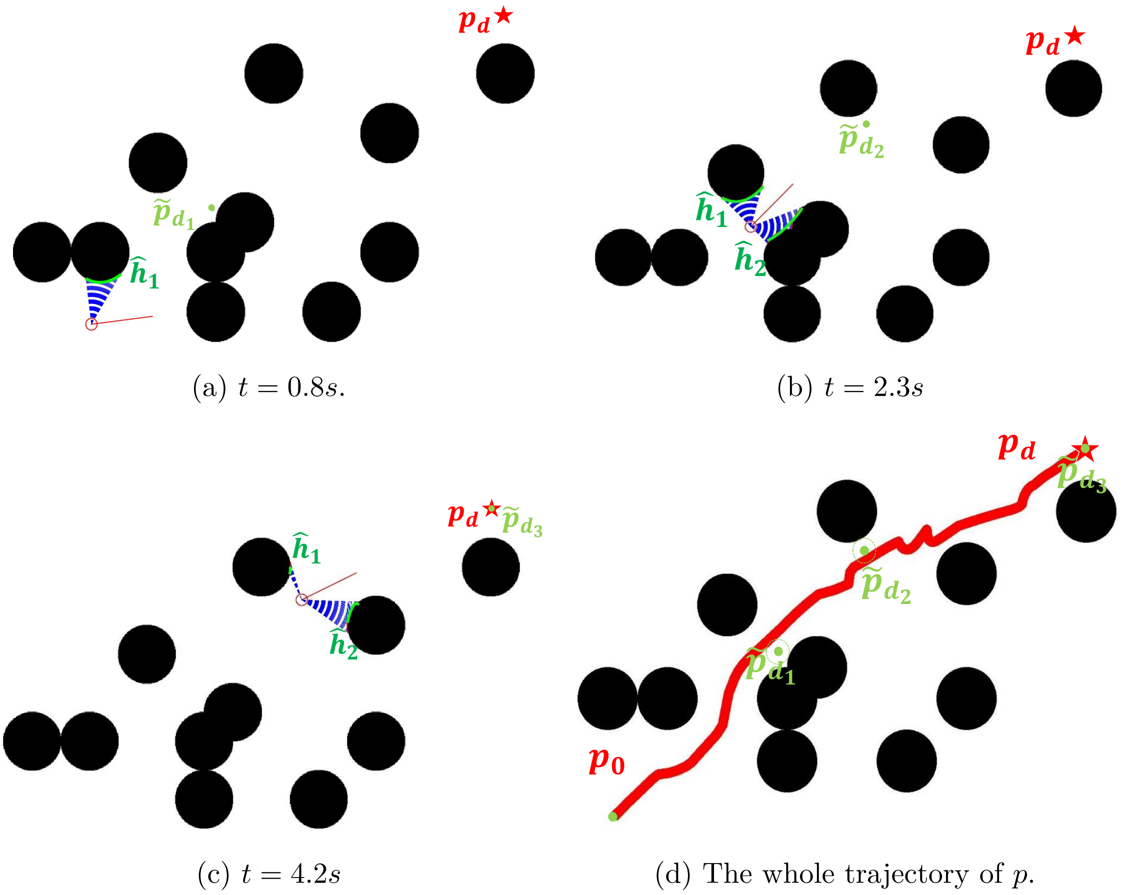
Figure 4.5: The performance comparison between the optimized $\alpha_1^*$ and the predetermined $\alpha_2$, $\alpha_3$ associated QP solutions. The green rectangle represents the input constraint set. The arrows point toward the ACS.

the reach-avoid task mentioned above.

$$\min_u \;\|u - u_{pd}\|^2 \tag{4.15a}$$
$$\text{s. t.} \; -0.3 < u_x, u_y < 0.3 \tag{4.15b}$$
$$\ddot{h} + \alpha^*_{1_1}\dot{h} + \alpha^*_{1_2}h \geq 0, \tag{4.15c}$$

where $\alpha^*_{1_1}$ and $\alpha^*_{1_2}$ are the optimized variables after solving the LP (4.14) based on the known CBF $h$. For comparison, prior-chosen constant vectors $\alpha_2 = [4, 1]^\top$, $\alpha_3 = [4, 2]^\top$ are picked to construct the constraint (4.15c). Note that the feasibility of the QP (4.15) is easily lost without choosing suitable values of $\alpha$ required for the HO-CBF (4.3) in Definition 4.1. Here $\alpha_2$ and $\alpha_3$ are well debugged parameters to ensure the QP feasibility.

As displayed in Figure 4.5a, the nominal $u_{pd}$ is an unsafe control input given that the mobile robot driven by the $u_{pd}$ crosses the obstacle $\mathcal{O}$. The minimally corrected $u_{pd}$ by solving the QP (4.15) drives the mobile robot to safely reach the destination. Furthermore, as shown in Figure 4.5a, the trajectory of the optimized $\alpha^*_1$ case is tighter around the obstacle as a consequence of enlarged ACS, i.e., closer to the desired trajectory (the cyan line) associated with $u_{pd}$. The ACSs of the constraint (4.15c) at $t = 2s$ and $t = 17s$ are displayed in Figure 4.5b. It is shown that the $\alpha^*_1$'s associated ACS is larger than the related ones of $\alpha_2$ and $\alpha_3$. This validate the effectiveness of the LP optimization (4.14).



(a) $t = 0.8s$.

(b) $t = 2.3s$

(c) $t = 4.2s$

(d) The whole trajectory of $p$.

Figure 4.6: The illustration of the outdoor scenario.

### 4.6.2 Validation in An Outdoor Scenario

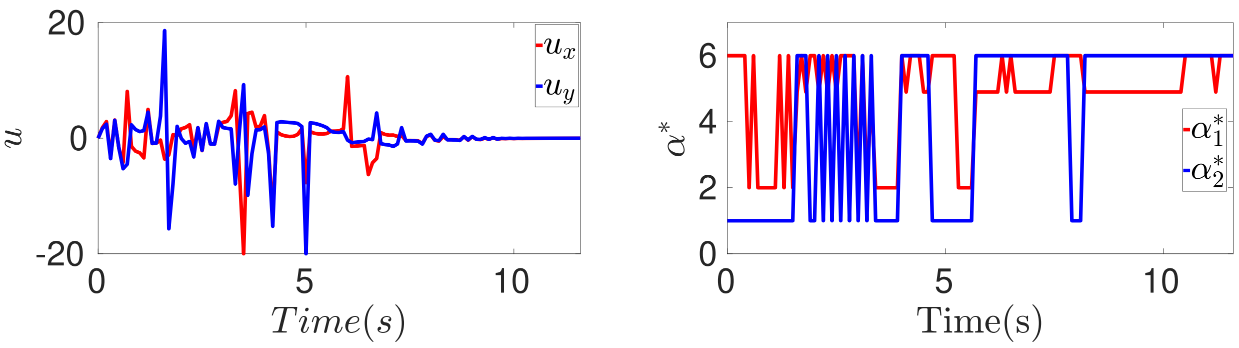
This subsection validates the efficiency of our proposed safe feedback control strategy (4.12) in an obstacle densely cluttered environment (see Figure 4.6). The numerical simulation is conducted on the basis of the Mobile Robotics Simulation Toolbox [88] and the quadprog solver of the Optimization Toolbox [89]. The detailed parameter settings to solve the formulated QP (4.12) and LP (4.14) are presented in Table 4.2.

Table 4.2: The parameter settings of the outdoor scenario.

| | |
|---|---|
| **Initial values** | $p_0 = [2, 4]^\top$, $v_0 = [1, 1]^\top$, $T = 10$ Hz |
| **Target values** | $p_d = [10, 10]^\top$, $v_d = [0, 0]^\top$ |
| **IL-CBF** | $\Phi = [1, x, x^2]$, $S_\theta = [-\pi/2, \pi/2]$, $S_r = 0.5\ m$ |
| **GD-CLF** | $P = \begin{bmatrix} 25 & 12.5 \\ 12.5 & 25 \end{bmatrix}$, $Q = \begin{bmatrix} 50 & 25 \\ 25 & 50 \end{bmatrix}$,<br>$S_\theta = [-\pi, \pi]$, $S_r = 4\ m$, $\bar{c}_2 = 1.5$ |
| **QP and LP** | $\overline{u}_x$, $\overline{u}_y = 20$, $\bar{c}_1 = 1$, $\alpha(t_0) = [5, 6]^\top$, $\overline{\alpha} = 6$ |

It is shown in Figure 4.6a-Figure 4.6c that the mobile robot exploits the sensed obstacle boundary data to learn the IL-CBFs $\hat{h}_1$, $\hat{h}_2$ based on Algorithm 2, and uses collision-free data to discover the subgoals $\tilde{p}_{d_1}$, $\tilde{p}_{d_2}$ via Algorithm 3. As displayed in Figure 4.6d, the mobile robot safely reaches the subgoals $\tilde{p}_{d_1}$, $\tilde{p}_{d_2}$ sequentially and finally reach the destination $p_d$ (same with $\tilde{p}_{d_3}$). Thus, it is concluded that the learned IL-CBFs (4.10) ensure collision avoidance with unforeseen obstacles, and the constructed GD-CLFs (4.11) based on the discovered subgoals guarantee the task fulfillment.

The evolution trajectories of the control inputs, and the optimized parameter $\alpha^*$ are displayed in Figure 4.7a and Figure 4.7b, respectively. The input saturation is satisfied, and the LP (4.14) outputs the optimized $\alpha^*$ to ensure the feasibility of the QP (4.12) during the whole operation process. A supplemental video for the outdoor scenario is referred to `https://youtu.be/FZsNcOUzEVs`.



(a) The trajectory of input $u$.

(b) The trajectory of optimized $\alpha^*$.

Figure 4.7: The trajectories of $u$, and $\alpha^*$ for the outdoor scenario.

### 4.6.3 Validation in An Indoor Scenario

This subsection further validates the effectiveness of our designed safe feedback control strategy in a maze environment (see Figure 4.8). It is worth mentioning that the application of common CBFs in a maze environment is seldom found in existing works. This is because multiple typical CBFs are required to achieve collision avoidance in such a maze environment, and certain CBFs would unavoidably treat collision-free spaces as unsafe regions. In this case, the mobile robot behaves conservatively and the QP might lost its feasibility. In particular, it is nontrivial to design barrier functions to separate safe and unsafe regions even though we have the full knowledge of the maze environment displayed in Figure 4.8. However, our developed IL-CBFs could efficiently deal with this maze environment. The detailed parameters to accomplish the safe operation in the maze environment is displayed in Table 4.3. The accompanying simulation videos are available at `https://youtu.be/FZsNcOUzEVs`.

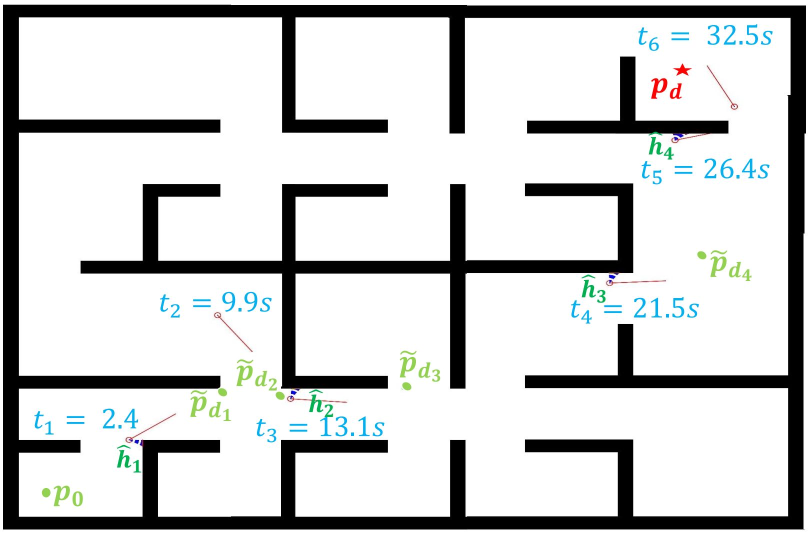
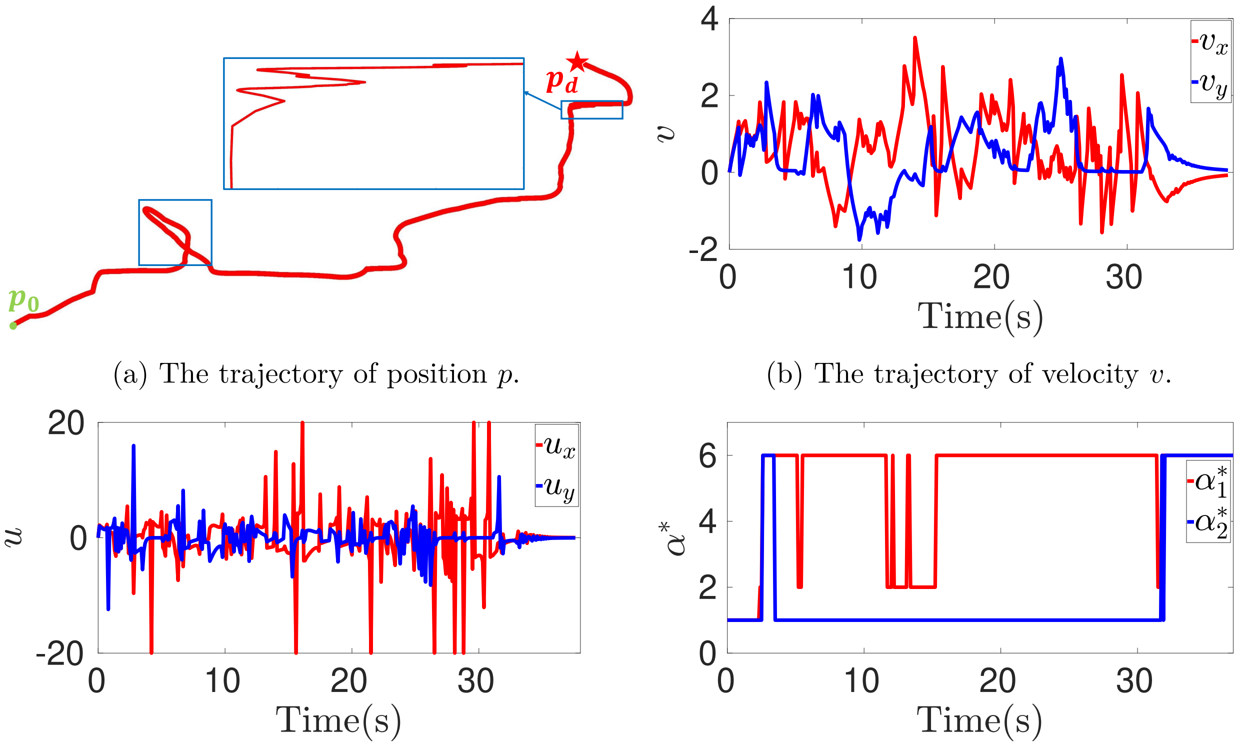


Figure 4.8: The illustration of the indoor scenario.

Table 4.3: The parameter settings of the indoor scenario.

| **Initial values** | $p_0 = [2,2]^\top$, $v_0 = [0,0]^\top$, $T = 10$ Hz |
|---|---|
| **Target values** | $p_d = [22,18]^\top$, $v_d = [0,0]^\top$ |
| **IL-CBF** | $\Phi = [1, x, x^2]$, $S_\theta = [-\pi/2, \pi/2]$, $S_r = 0.5\ m$ |
| **GD-CLF** | $P = \begin{bmatrix} 25 & 12.5 \\ 12.5 & 25 \end{bmatrix}$, $Q = \begin{bmatrix} 50 & 25 \\ 25 & 50 \end{bmatrix}$,<br>$S_\theta = [-\pi, \pi]$, $S_r = 4\ m$, $\bar{c}_2 = 1.5$ |
| **QP and LP** | $\overline{u}_x$, $\overline{u}_y = 20$, $\bar{c}_1 = 1$, $\alpha(t_0) = [5,6]^\top$, $\overline{\alpha} = 6$ |

As displayed in Figure 4.8 and Figure 4.9a, the mobile robot operates safely in the maze environment and finally reach the goal position $p_d$. However, we observe inefficient operation (shown in the blue rectangle of Figure 4.9a) of the mobile robot in this unforeseen maze environment. This is due to the simple heuristic (i.e., shortest distance rule) used in



(a) The trajectory of position $p$.

(b) The trajectory of velocity $v$.

(c) The trajectory of input $u$.

(d) The trajectory of optimized $\alpha^*$.

Figure 4.9: The trajectories of position $p$, velocity $v$, control input $u$, and optimized $\alpha^*$ for the indoor scenario.

Algorithm 3. This problem could be avoided by changing the sensor range in an adaptive way. We deliberately present this incomplete case to show the potential drawback of our method. The trajectories of the mobile robot's velocity, control input and optimized $\alpha^*$ are displayed in Figure 4.9b, Figure 4.9c, and Figure 4.9d, respectively. The input saturation is always satisfied and $\alpha^*$ updates to ensure the QP feasibility.

## 4.7 Summary

This work presents a safe feedback control policy that couples sensor with control to fulfill safe operation in uncertain environments. Our developed IL-CBFs are united with GD-CLFs in a QP optimization framework to generate the safe feedback control strategies. The formulated LP optimization improves the QP feasibility by enlarging the ACSs of IL-CBFs. Multiple comparative numerical simulations are conducted to validate the effectiveness of the proposed method.

The proposed integrated perception and control approach provides the limited-performance mobile robot with a low-cost solution (regarding hardware requirements and computation loads) to the nontrivial safe operation problem in uncertain environments. Our method enjoys the theoretical guarantees (safety and optimality regarding control efforts) favoured in the traditional control field, but also achieve the same promising performance as end-to-end learning methods. However, our method is in essence a local and reactive method. Thus, comparing to the global method, the solution might trap in local minimum. The current approach are developed under perfect measurement data and system dynamics. The future work aims to systematically analyze the influence of model uncertainties and environmental disturbances to the safe feedback control strategy. Besides, the developed IL-CBF will be extended to a high-dimensional system in a dynamic uncertain environment, and also the agent-to-agent collision avoidance in a multi-agent system.

# Part II

# Reinforcement Learning Approaches

# Off-Policy Risk-Sensitive Reinforcement Learning Based Constrained Optimal Control 5

The set-theoretic methods in Part I mainly investigate the guaranteed performance control; however, optimality is not considered. After that, RL based approaches are used in Part II to solve optimal control problems within consideration of robustness and safety.

This chapter proposes an off-policy risk-sensitive RL based control framework to jointly optimize the task performance and the constraint satisfaction in a disturbed environment. The provable robust safety guarantee is provided employing nominal model knowledge and an assumed known disturbance bound. The organization of this chapter is as follows. The risk-aware value function, constructed using the pseudo control and the risk-sensitive input and state penalty terms, is introduced in Section 5.1 to convert the original constrained robust stabilization problem into an equivalent unconstrained optimal control problem. The optimal control solutions of continuous time nonlinear systems are extremely difficult to determine, if not impossible. Therefore, practitioners turn to suboptimal schemes and approximate solutions. Section 5.2 elucidates the approximate solution to the value function of the HJB equation, which results in the approximate optimal control policy that satisfies both input and state constraints under disturbances. Besides, the critic NN weight convergence is guaranteed by replaying experience data to the weight update law. Moreover, online and offline algorithms are developed to serve as principled ways to record informative experience data, which contributes to provide the sufficient excitation required for the weight convergence. Simulation results shown in Section 5.3 illustrate the effectiveness of our proposed control framework. Finally, Section 5.4 summarizes this chapter.

## 5.1 Problem Formulation

### 5.1.1 Formulation of Constrained Robust Stabilization Problem

Consider the continuous time nonlinear dynamical system:

$$\dot{x} = f(x) + g(x)u(x) + k(x)d(x), \tag{5.1}$$

where $x \in \mathbb{R}^n$ and $u(x) \in \mathbb{R}^m$ are system states and inputs. $f(x) : \mathbb{R}^n \to \mathbb{R}^n$, $g(x) : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ are the known drift and input dynamics, respectively. $k(x) : \mathbb{R}^n \to \mathbb{R}^{n\times r}$ represents the known differential system function. $d(x) : \mathbb{R}^n \to \mathbb{R}^r$ denotes the unknown additive disturbance. The general case that the additive disturbance is unmatched (i.e., $k(x) \neq g(x)$) is considered here. Assuming that $f(0) = 0$ and $d(0) = 0$, which means that the equilibrium point is $x = 0$.

Before proceeding, the following assumptions are provided. These assumptions are common in ADP related works and facilitate the theoretical analysis.

**Assumption 5.1.** *[90] $f(x)+g(x)u$ is Lipschitz continuous on a set $\Omega \subseteq \mathbb{R}^n$ that contains the origin, and the system is stabilizable on $\Omega$. There exists $g_M \in \mathbb{R}^+$ such that the input dynamics is bounded by $\|g(x)\| \leq g_M$.*

**Assumption 5.2.** *[91] The unknown additive disturbance $d(x)$ is bounded by a known nonnegative function $d_M(x)$: $\|d(x)\| \leq d_M(x)$, and $d_M(0)=0$.*

Based on the aforementioned settings, we formulate the constrained robust stabilization problem (CRSP) as follows.

**Problem 5.1** (CRSP)**.** *Given Assumptions 5.1-5.2, design a control strategy $u(x)$ to stabilize the closed-loop system* (5.1) *to the equilibrium point under additive disturbances $d(x)$, input saturation*

$$\mathbb{U}_j = \{u_j \in \mathbb{R} : |u_j| \leq \beta\}, j = 1, \cdots, m, \tag{5.2}$$

*where $\beta \in \mathbb{R}^+$ is a known saturation bound; and state constraints*

$$\mathbb{X}_i = \{x \in \mathbb{R}^n : h_i(x) < 0\}, i = 1, \cdots, n_c, \tag{5.3}$$

*where $\mathbb{X}_i$ is a closed and convex set that contains the origin in its interior; $h_i(x) : \mathbb{R}^n \to \mathbb{R}$ is a known continuous function that relates with the i-th state constraint; $n_c \in \mathbb{N}^+$ is the number of considered state constraints.*

## 5.1.2 Transformation to Optimal Control Problem

Problem 5.1 consists of three sub-problems: disturbance rejection, input saturation, and state constraint. It is nontrivial for ADP to directly deal with these sub-problems together [92]. Thus, in this section, with the pseudo control technique proposed in [91], [93], reformulated risk-sensitive input penalty terms based on [90], and our newly designed risk-sensitive state penalty terms, we first transform the CRSP clarified as Problem 5.1 into an equivalent optimal control problem. Then, we attempt to solve the sub-problems mentioned above simultaneously under an optimization framework.

### A. Pseudo Control and Auxiliary System

As illustrated in [91], for a system suffering a matched disturbance, its disturbance-rejection control strategy could be designed by solving its nominal system's optimal control problem, wherein a cost function including the square of the disturbance bound is considered. For the unmatched disturbance $k(x)d(x)$ considered in (5.1), however, the above robust control design strategy cannot be directly applied. Thus, to address the unmatched $k(x)d(x)$ under an optimization framework as well, it is firstly decomposed as [93]

$$k(x)d(x) = g(x)\bar{d}(x) + h(x)d(x), \tag{5.4}$$

where $\bar{d}(x) = g^{\dagger}(x)k(x)d(x) : \mathbb{R}^n \to \mathbb{R}^m$, and $h(x) = (I - g(x)g^{\dagger}(x))k(x) : \mathbb{R}^n \to \mathbb{R}^{n\times r}$. Here $\dagger$ denotes the Moore-Penrose inverse. Then, we introduce the following auxiliary system with a pseudo control $v(x) : \mathbb{R}^n \to \mathbb{R}^r$

$$\dot{x} = f(x) + g(x)u(x) + h(x)v(x), \tag{5.5}$$

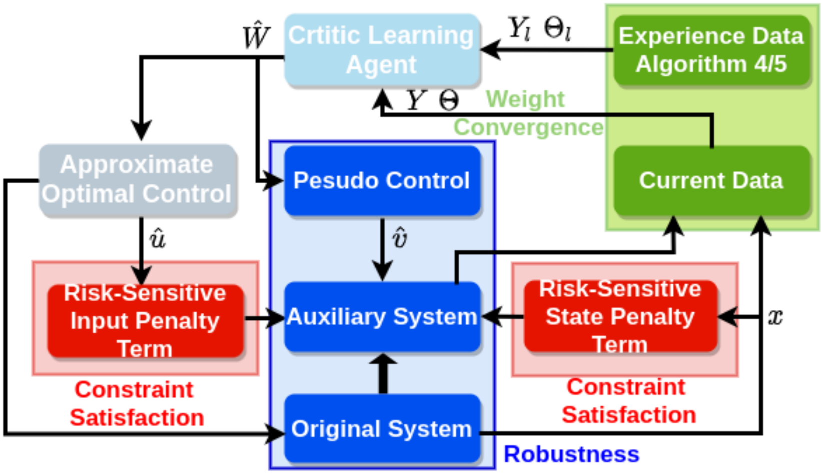


Figure 5.1: Schematic of the off-policy risk-sensitive RL based control framework containing three key components: 1) Robustness: pseudo control based auxiliary system enables us to address additive disturbances under an optimization framework in Section 5.1.2; 2) Constraint satisfaction: risk-sensitive input and state penalty terms are incorporated into the cost function to enforce input and state constraint satisfaction during the optimization process in Section 5.1.3; 3) Weight convergence: online or offline recorded informative experience data are replayed to support the online learning of the critic NN in Section 5.2.

to accomplish that both $g(x)\bar{d}(x)$ and $h(x)d(x)$ are matched disturbances with respect to the range of $g(x)$ and $h(x)$, respectively. Finally, similar to the robust control design strategy proposed in [91], by solving the optimal control problem of the auxiliary system (5.5) with a cost function including the square of the bounds of $\bar{d}(x)$ and $d(x)$, we could address the disturbance-rejection problem of the system (5.1) under an optimization framework. The corresponding rigorous proof is provided later in Theorem 5.1 and the following assumption is introduced for the later analysis.

**Assumption 5.3.** *[93] The continuous function $h(x)$ is bounded as $\|h(x)\| \leq h_M$; $\bar{d}(x)$ is bounded by a nonnegative function $l_M(x)$ : $\left\|\bar{d}(x)\right\| \leq l_M(x)$, and $l_M(0) = 0$.*

### B. Risk-Sensitive Input and State Penalty Terms

To tackle input and state constraints under an optimization framework, here we follow the idea of risk-sensitive RL where multiple risk measures, e.g., high moment or conditional value at risk, are used to deal with constraints of Markov decision processes [94]. However, the available risk measures in the risk-sensitive RL field cannot guarantee strict constraint satisfaction and/or not efficient (even inappropriate) to address constraints of continuous time nonlinear systems. Thus, we propose risk-sensitive input penalty term (RS-IP) in Definition 5.1 and risk-sensitive state penalty term (RS-SP) in Definition 5.2 as new risk measures during the learning process to enforce strict satisfaction of input and state constraints of continuous time nonlinear systems.

**Definition 5.1** (RS-IP)**.** *A continuous and differential function $\phi(u)$ is a risk-sensitive input penalty term if it has the following properties:*

*(1) A bounded monotonic odd function with $\phi(0) = 0$;*
*(2) The first-order partial derivatives of $\phi(u)$ is bounded.*

Here the RS-IP term is a reformulation of the nonquadratic functional used in [90], [95] to confront input constraints.

**Definition 5.2** (RS-SP). *Given the closed region $\mathbb{X}_i$, $i = 1, \cdots, n_c$, defined as* (5.3)*, a continuous scalar function $S_i(x) : \mathbb{X}_i \to \mathbb{R}$, $i = 1, \cdots, n_c$, is a risk-sensitive state penalty term if the following proprieties hold:*
*(1) $S_i(0) = 0$, and $S_i(x) > 0, \forall x \neq 0$;*
*(2) $S_i(x) \to \infty$ if $x$ approaches $\partial \mathbb{X}_i$;*
*(3) For initial value $x(0) \in \mathrm{Int}(\mathbb{X}_i)$, there exists $s \in \mathbb{R}^+$ such that $S_i(x(t)) \leq s, \forall t \geq 0$ along solutions of the dynamics.*

Comparing with similar works [41], [42] that use state penalty functions to tackle state constraints but without strict constraint satisfaction proofs, our proposed RS-SP term enables us to provide the strict constraint satisfaction proofs in Theorem 5.1. Here the novel RS-SP term is inspired by the so-called barrier Lyapunov function [4] that we utilized in Chapter 2 and Chapter 3. The first point of Definition 5.2 denotes that $S_i(x)$ is an effective Lyapunov function candidate, which enables $S_i(x)$ to serve as part of Lyapunov function for the system stability proof. The last two points imply that $\inf_{x \to \partial \mathbb{X}_i} S_i(x) = \infty$ and $\inf_{x \in \mathrm{Int}(\mathbb{X}_i)} S_i(x) \geq 0$, which means that $S_i(x)$ serves as a barrier certificate for an allowable operating region $\mathbb{X}_i$.

### C. Optimal Control Problem

Based on the auxiliary system (5.5) and Definitions 5.1-5.2, an equivalent optimal control problem (OCP) of the CRSP in Problem 5.1 is clarified as Problem 5.2. Comparing with traditional ADP that accomplishes partial objectives of performance, robustness, and input/state constraint satisfaction [41], [90], [92], [96], the applied problem transformation here enables us to consider such multiple objectives together.

**Problem 5.2** (OCP). *Given Assumptions 5.1-5.3, consider the auxiliary system* (5.5)*, find $u(x)$ and $v(x)$ to minimize the cost function*

$$V(x(t)) = \int_t^\infty r(x(\tau), u(x(\tau)), v(x(\tau)))\, d\tau, \tag{5.6}$$

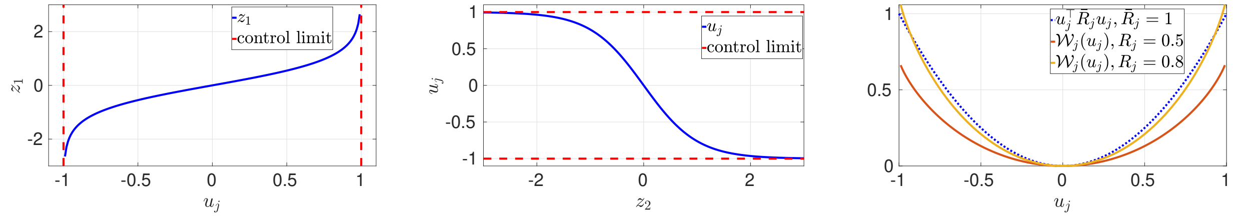
*where the utility function follows $r(x, u(x), v(x)) = r_d(x) + \rho v^\top(x)v(x) + r_c(x, u(x))$ with $\rho \in \mathbb{R}^+$, $r_d(x) = l_M^2(x) + \rho d_M^2(x)$, and $r_c(x, u(x)) = \mathcal{W}(u(x)) + \mathcal{L}(x)$. The input penalty function $\mathcal{W}(u(x))$ follows*

$$\mathcal{W}(u(x)) = \sum_{j=1}^{m} 2 \int_0^{u_j} \beta R_j \phi^{-1}(\vartheta_j/\beta)\, d\vartheta_j, \tag{5.7}$$

*where $\phi(\cdot)$ is the RS-IP term in Definition 5.1; $R_j$ is the $j$-th diagonal element of a positive definite diagonal matrix $R \in \mathbb{R}^{m \times m}$. The state penalty function $\mathcal{L}(x)$ is defined as*

$$\mathcal{L}(x) = x^\top Q x + \sum_{i=1}^{n_c} k_i S_i(x), \tag{5.8}$$

*where $Q \in \mathbb{R}^{n \times n}$ is a positive definite matrix; $k_i$ is the risk sensitivity parameter that follows $k_i = 1/(1 + d_i^2)$, where $d_i$ is the distance from the state $x$ to the boundary of $h_i(x)$; $S_i(\cdot)$ is the RS-SP term in Definition 5.2 for the $i$-th state constraint.*



(a) Plot of $z_1$ with $R_j$ =1 (b) Plot of $u_j = -\beta \tanh(z_2)$ . (c) Plot of $u_j^\top \bar{R}_j u_j$ and $\mathcal{W}_j(u_j)$ .

Figure 5.2: Graphical illustration of the working scheme of the input penalty function $\mathcal{W}(u(x))$ with $\beta = 1$, $z_1 = \beta R_j \tanh^{-1}(u_j/\beta)$, $z_2 \in \mathbb{R}$ and $u_j \in (-1, 1)$.

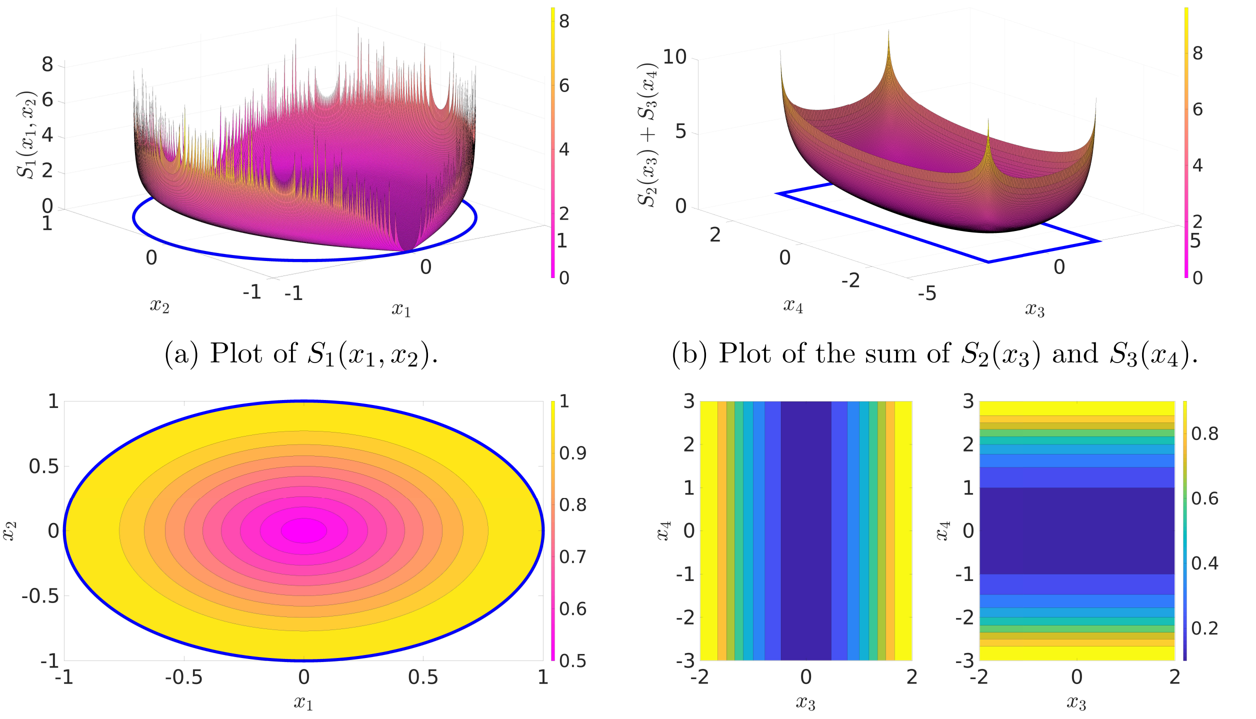
Unlike ADP related works [41], [90] that incorporate nonquadratic functionals to tackle input saturation but without considering control effort related performance, $\mathcal{W}(u(x))$ in (5.7) could take into consideration of requirements for both control limits and control energy expenditures by choosing a suitable matrix $R$. More details are introduced in Section 5.1.3. The common used risk-neutral quadratic function $x^\top Qx$ [92], [96] (capturing the desired state performance) is augmented with the newly designed weighted RS-SP term $\sum_{i=1}^{n_c} k_i S_i(x)$ (addressing multiple state constraints) to construct $\mathcal{L}(x)$ in (5.8), which enables us to consider the state-related performance and constraints together. The incorporation of $S_i(x)$ into $\mathcal{L}(x)$ deteriorates the desired performance represented by $x^\top Qx$. Therefore, we propose the risk sensitivity parameter $k_i$, which relates with the distance from the constraint boundary, to specify the inevitable trade-off between the state-related performance and constraint satisfaction during the learning process. Note that this kind of trade-off is ignored in existing related works [41], [42]. The detailed mechanism of $\mathcal{L}(x)$ is illustrated in Section 5.1.3.

## 5.1.3 Mechanism of Input and State Penalty Functions

The mechanism of $\mathcal{W}(u(x))$ and $\mathcal{L}(x)$ to enable the learning process to preserve performance without violating strict input/state constraint satisfaction is detailly clarified here.

### A. Mechanism of Input Penalty Function $\mathcal{W}(u(x))$

By Definition 5.1, the explicit form of the RS-IP term is chosen as $\phi(\cdot) = \tanh(\cdot)$ [90], [97]. Given the inevitable trade-off between input-related performance and constraint satisfaction, $\mathcal{W}(u(x))$ is designed to address the input constraints (5.2) and approximate $u^\top \bar{R}u$ (a common desired performance criterion for control efforts) simultaneously, where $\bar{R} \in \mathbb{R}^{m\times m}$ is a prior-chosen positive definite matrix reflecting designers' preferences. The mechanism of $\mathcal{W}(u(x))$ to tackle input constraints could be clarified from two perspectives, see Figure 5.2a and Figure 5.2b, respectively. In the first perspective, input constraints are considered in a long time-horizon. $\mathcal{W}(u(x))$ in (5.7) is an integration of $\beta R_j \tanh^{-1}(u_j/\beta)$ that is denoted as $z_1$ in Figure 5.2a. When any $u_j$, $j = 1, \cdots, m$, approaches to the input constraint boundaries $\pm\beta$, it follows that the value of $\mathcal{W}(u(x))$ will be infinity. Since the optimization process aims to minimize the cost function, the resulting optimal control strategy will be away from $\pm\beta$; Otherwise, a high value of the cost function occurs. From the other perspective, according to the later result in (5.12), the resulting optimal control strategy based on $\mathcal{W}(u(x))$ is in a form of $\tanh(\cdot)$ whose boundness enforces strict satisfaction of input constraints, as shown



(a) Plot of $S_1(x_1, x_2)$.

(b) Plot of the sum of $S_2(x_3)$ and $S_3(x_4)$.

(c) Plot of $k_1$ for $S_1(x_1, x_2)$.

(d) Plot of $k_2$ for $S_2(x_3)$ (left) and $k_3$ for $S_3(x_4)$ (right).

Figure 5.3: Graphical illustration of the working scheme of the RS-SP terms $S_1(x_1, x_2)$, $S_2(x_3)$, $S_3(x_4)$ and their corresponding risk sensivity parameters $k_1$, $k_2$ and $k_3$.

in Figure 5.2b. The construction of $\mathcal{W}(u(x))$ to reflect the desired performance for control energy is shown in Figure 5.2c. Consider $\mathcal{W}_j(u_j)$, the $j$-th summand of $\mathcal{W}(u(x))$. It follows

$$\mathcal{W}_j(u_j) = 2\beta R_j u_j \tanh^{-1}(u_j/\beta) + \beta^2 R_j \log\Big(1 - u_j^2/\beta^2\Big). \tag{5.9}$$

As displayed in Figure 5.2c, $\mathcal{W}_j(u_j)$ approximates the desired control energy criterion $u_j^\top \bar{R} u_j$ well via adjusting the value of $R_j$. Based on the above discussion, we know that $\mathcal{W}(u(x))$ in (5.7) tackles input constraints while preserving performance concerning control energy expenditures.

### B. Mechanism of State Penalty Function $\mathcal{L}(x)$

According to Definition 5.2, when a potential state constraint violation happens, the corresponding RS-SP term will approach to infinity. Since the optimal control strategy aims to minimize the total cost, states will be pushed away from the direction where a high value of the RS-SP based $\mathcal{L}(x)$ occurs. Thus, the state constraint violation is avoided. To satisfy Definition 5.2, we choose $S_i(x) = \log\left(h_i(x)\right)$ here. Note that the explicit form of $\log(h_i(x))$ is adjusted based on given state constraints, which is exemplified later. For a better explanation of the mechanism of the RS-SP term $S_i(x)$ and the corresponding risk sensitivity parameter $k_i$, we present a four-dimensional system example with the safe regions defined as $\mathbb{X}_1 = \{x_1, x_2 \in \mathbb{R} : h_1(x_1, x_2) = x_1^2 + x_2^2 - 1 < 0\}$ [98], $\mathbb{X}_2 = \{x_3 \in \mathbb{R} : h_2(x_3) = |x_3| - 2 < 0\}$, and $\mathbb{X}_3 = \{x_4 \in \mathbb{R} : h_3(x_4) = |x_4| - 3 < 0\}$ [39]. The corresponding RS-SP terms are designed as $S_1(x_1, x_2) = \log(\alpha(x_2)/(\alpha(x_2) - x_1^2))$ with $\alpha(x_2) = 1 - x_2^2$, $S_2(x_3) = \log(4/(4 - x_3^2))$, and $S_3(x_4) = \log(9/(9 - x_4^2))$, respectively. As displayed in Figure 5.3a-5.3b, these RS-SP terms act as barriers at constraint boundaries and confine the states remain in the safe regions. This inherent risk-sensitive property enables us to tackle state constraints under an optimization framework. As long as initial states lie in the safe regions and the cost function is always bounded as time evolves, the subsequent state evolution will be restricted to the safe regions. From Figure 5.3c-5.3d, we know that the role of $S_i(x)$ will be discouraged by $k_i$ when states are far away from the boundary of $h_i(x)$. Therefore, state-related performance is maintained when no state constraint violation occurs.

### 5.1.4 HJB Equation of OCP

Aiming at the transformed OCP in Problem 5.2, for any admissible control policies $u, v \in \Psi(\Omega)$, where $\Psi(\Omega)$ is the admissible control set [90, Definition 1], the associated optimal cost function follows

$$V^*(x(t)) = \min_{u,v\in\Psi(\Omega)} \int_t^\infty r(x(\tau), u(x(\tau)), v(x(\tau)))\, d\tau, \tag{5.10}$$

and the HJB equation satisfies

$$0 = \min_{u,v\in\Psi(\Omega)} [\nabla V^{*^T} (f(x) + g(x)u(x) + h(x)v(x)) + r(x, u(x), v(x))]. \tag{5.11}$$

Assuming that the minimum on the right side of (5.11) exits and is unique [19]. Then, the closed forms of optimal control policies $u^*(x)$ and $v^*(x)$ are obtained as [90]

$$u^*(x) = -\beta \tanh\left(\frac{1}{2\beta} R^{-1} g^\top(x)\nabla V^*\right), \tag{5.12}$$

$$v^*(x) = -\frac{1}{2\rho} h^\top(x)\nabla V^*. \tag{5.13}$$

### 5.1.5 Problem Equivalence

Here we defer a detailed explanation of the method to get the optimal control policies (5.12) and (5.13) in Section 5.2, and focus now on the proof of equivalence between Problem 5.1 and Problem 5.2. Comparing with the result provided in [91] that merely considers additive disturbances, as shown in Theorem 5.1, the additional consideration of input and state constraints further complicates the theoretical analysis.

**Theorem 5.1.** *Consider the system described by* (5.1) *and controlled by the optimal control policy* (5.12)*. Suppose Assumptions 5.1-5.3 hold and the initial states and control inputs lie in the predefined constraint satisfying sets* (5.2) *and* (5.3)*. The optimal control policy* (5.12) *guarantees robust stabilization of the system* (5.1) *without violating the input constraint* (5.2) *and state constraint* (5.3)*, if there exists a scalar $\epsilon_s \in \mathbb{R}^+$ such that the following inequality is satisfied*

$$\mathcal{L}(x) > 2\rho v^{*\top}(x)v^*(x) + \epsilon_s. \tag{5.14}$$

*Proof. (i) Proof of stability.* As for $V^*(x)$ defined as (5.10), we know that when $x = 0$, $V^*(x) = 0$, and $V^*(x) > 0$ for $\forall x \neq 0$. Thus, it can serve as a Lyapunov function candidate for stability proofs. Taking time derivative of $V^*(x)$ along the system (5.1) yields

$$\begin{aligned}\dot{V}^* &= \nabla {V^*}^\top (f(x) + g(x)u^*(x) + k(x)d(x)) \\ &= \nabla {V^*}^\top (f(x) + g(x)u^*(x) + h(x)v^*(x)) \\ &\quad + \nabla {V^*}^\top g(x)g^\dagger(x)k(x)d(x) + \nabla {V^*}^\top h(x)(d(x) - v^*(x)).\end{aligned} \tag{5.15}$$

In light of (5.11), we get

$$\nabla {V^*}^\top (f(x) + g(x)u^*(x) + h(x)v^*(x)) = -\mathcal{W}(u^*(x)) - \mathcal{L}(x) - \rho {v^*}^\top(x)v^*(x) - l_M^2(x) - \rho d_M^2(x). \tag{5.16}$$

From (5.12), we get

$$\nabla {V^*}^\top g(x) = -2\beta R \tanh^{-1}(u^*(x)/\beta). \tag{5.17}$$

Based on (5.13), the following equation establishes

$$\nabla {V^*}^\top h(x) = -2\rho v^*(x). \tag{5.18}$$

Substituting (5.16), (5.17) and (5.18) into (5.15) yields

$$\begin{aligned}\dot{V}^* = &-\mathcal{W}(u^*(x)) - \mathcal{L}(x) - \rho {v^*}^\top(x)v^*(x) - l_M^2(x) - \rho d_M^2(x) \\ &- 2\beta R \tanh^{-1}(u^*(x)/\beta)g^\dagger(x)k(x)d(x) - 2\rho {v^*}^\top(x)d(x) + 2\rho {v^*}^\top(x)v^*(x).\end{aligned} \tag{5.19}$$

By setting $\varsigma_j = \tanh^{-1}(\tau_j/\beta)$, we get

$$\begin{aligned}\mathcal{W}(u^*(x)) &= 2\beta \sum_{j=1}^{m} \int_0^{u_j^*} R_j \tanh^{-1}(\tau_j/\beta)\, d\tau_j \\ &= 2\beta^2 \sum_{j=1}^{m} \int_0^{\tanh^{-1}(u_j^*/\beta)} R_j \varsigma_j (1 - \tanh^2(\varsigma_j))\, d\varsigma_j \\ &= \beta^2 \sum_{j=1}^{m} R_j (\tanh^{-1}(u_j^*(x)/\beta))^2 - \epsilon_t,\end{aligned} \tag{5.20}$$

where $\epsilon_t = 2\beta^2 \sum_{j=1}^{m} \int_0^{\tanh^{-1}(u_j^*(x)/\beta)} R_j \varsigma_j \tanh^2(\varsigma_j)\, d\varsigma_j$. Based on the integral mean-value theorem, there exist a series of $\theta_j \in [0, \tanh^{-1}(\mu_j^*(x)/\beta], j = 1, \cdots, m$, such that

$$\epsilon_t = 2\beta^2 \sum_{j=1}^{m} R_j \tanh^{-1}(\mu_j^*(x)/\beta)\theta_j \tanh^2(\theta_j). \tag{5.21}$$

Bearing in mind the relation (5.17) and the fact $0 < \tanh^2(\theta_j) \leq 1$, it follows that

$$\begin{aligned}\epsilon_t &\leq 2\beta^2 \sum_{j=1}^{m} R_j \tanh^{-1}(\mu_j^*(x)/\beta)\theta_j \\ &\leq 2\beta^2 \sum_{j=1}^{m} R_j (\tanh^{-1}(\mu_j^*(x)/\beta))^2 \\ &= \frac{1}{2} \nabla {V^*}^\top g(x) R^{-1} g^\top(x) \nabla V^*.\end{aligned} \tag{5.22}$$

According to the definition of the admissible policy [90], $V^*$ is finite. Moreover, there exists $w_M > 0$ such that $\|\nabla V^*\| \leq \omega_M$. Based on Assumption 5.1, we could rewrite (5.22) as

$$\epsilon_t \leq b_{\epsilon_t}. \tag{5.23}$$

where $b_{\epsilon_t} = \frac{1}{2}\left\|R^{-1}\right\| g_M^2 \omega_M^2$. Based on Assumption 5.2, the following equations establish:

$$\begin{aligned} -2\beta R \tanh^{-1}(u^*(x)/\beta) g^\dagger(x) k(x) d(x) &\leq \left\|\beta R \tanh^{-1}(u^*(x)/\beta)\right\|^2 + \left\|g^\dagger(x) k(x) d(x)\right\|^2 \\ &\leq \beta^2 \sum_{j=1}^m R_j^2 (\tanh^{-1}(u^*(x)/\beta))^2 + l_M^2(x), \end{aligned} \tag{5.24}$$

$$-2\rho v^{*\top}(x) d(x) \leq \rho \left\|v^*(x)\right\|^2 + \rho \left\|d(x)\right\|^2 \leq \rho \left\|v^*(x)\right\|^2 + \rho d_M^2(x). \tag{5.25}$$

Substituting (5.20), (5.23), (5.24) and (5.25) into (5.19), we have

$$\begin{aligned} \dot{V}^* &\leq -\mathcal{L}(x) + 2\rho v^{*\top}(x) v^*(x) + b_{\epsilon_t} + \beta^2 \sum_{j=1}^m (R_j^2 - R_j)(\tanh^{-1}(u^*(x)/\beta))^2 \\ &= -\mathcal{L}(x) + 2\rho v^{*\top}(x) v^*(x) + \epsilon_s. \end{aligned} \tag{5.26}$$

where $\epsilon_s = b_{\epsilon_t} + \beta^2 \sum_{j=1}^m (R_j^2 - R_j)(\tanh^{-1}(u^*(x)/\beta))^2$. Thus, $\dot{V}^* < 0$ establishes, if the condition (5.14) holds. It yields that the optimal control policy $u^*(x)$ robustly stabilizes the system (5.1).

*(ii) Proof of input and state constraint satisfaction.* Denote $V^*(0)$ as the value of the Lyapunov function candidate $V^*$ at $t = 0$. According to the definition of admissible control policies, $V^*(0)$ is a bounded function. If $\mathcal{L}(x) > 2\rho v^{*\top}(x) v^*(x) + \epsilon_s$, $\dot{V}^* < 0$ establishes, which means that $V^*(t) < V^*(0)$, $\forall t$. The boundness of $V^*(t)$ implies that state constraints will not be violated; Otherwise, $V^*(t) \to \infty$ if any state constraint violations happens according to Definition 5.2. Since the hyperbolic tangent function satisfies $-1 \leq \tanh(\cdot) \leq 1$, the optimal control policy in (5.12) follows $-\beta \leq u^*(x) \leq \beta$, i.e., inputs are confined into the safety set (5.2). The proof provided here means that the optimal control policy $u^*(x)$ for the system (5.1) guarantees satisfaction of both constraints in terms of the system states and control inputs. □

It is proven in Theorem 5.1 that the CRSP (Problem 5.1) is equivalent to the OCP (Problem 5.2) under the inequality (5.14). Thus, in order to solve the original CRSP, the current task is to obtain the optimal control law (5.12) focusing on the transformed OCP, which is detailly clarified in the next section.

## 5.2 Approximate Solution to OCP

To get the approximate solution to the OCP, instead of introducing a common actor-critic structure used in [19], [90], here we adopt a single critic structure which enjoys lower computation complexity [95]. Furthermore, departing from traditional methods that directly add additional noises into inputs to meet the PE condition required for the NN weight convergence [19], [24], here we reply experience data to the off-policy weight update law to achieve a sufficient excitation required for the critic NN weight convergence. Additionally, an online PER algorithm and an offline experience buffer construction algorithm are proposed as principled ways to provide the sufficient rich experience data.

### 5.2.1 Value Function Approximation

According to the Weierstrass high-order approximation theorem [99], there exists a weighting matrix $W^* \in \mathbb{R}^N$ such that the continuous value function is approximated as

$$V^*(x) = W^{*\top}\Phi(x) + \epsilon(x), \tag{5.27}$$

for $x \in \Omega$ with $\Omega$ being a compact set, where $\Phi(x) : \mathbb{R}^n \to \mathbb{R}^N$ is the NN activation function in a polynominal form, and $\epsilon(x) \in \mathbb{R}$ is the approximation error. Denote $\nabla\Phi \in \mathbb{R}^{N\times n}$ and $\nabla\epsilon(x) \in \mathbb{R}^n$ as the partial derivatives of $\Phi(x)$ and $\epsilon(x)$, respectively. As $N \to \infty$, both $\epsilon(x)$ and $\nabla\epsilon(x)$ converge to zero uniformly. Without loss of generality, the following assumption is given.

**Assumption 5.4.** *[19] There exist constants $b_\epsilon, b_{\epsilon x}, b_\Phi, b_{\Phi x} \in \mathbb{R}^+$ such that $\|\epsilon(x)\| \leq b_\epsilon$, $\|\nabla\epsilon(x)\| \leq b_{\epsilon x}$, $\|\Phi(x)\| \leq b_\Phi$, and $\|\nabla\Phi(x)\| \leq b_{\Phi x}$.*

For fixed admissible control policies $u(x)$ and $v(x)$, inserting (5.27) into (5.11) yields the Lyapunov equation (LE)

$$W^{*\top}\nabla\Phi(f(x) + g(x)u(x) + h(x)v(x)) + r(x, u(x), v(x)) = \epsilon_h, \tag{5.28}$$

where the residual error follows $\epsilon_h = -(\nabla\epsilon(x))^\top(f(x)+g(x)u(x)+h(x)v(x)) \in \mathbb{R}$. According to Assumption 5.1, the system dynamics is Lipschitz. This leads to the bounded residual error, i.e., there exists $b_{\epsilon_h} \in \mathbb{R}^+$ such that $\|\epsilon_h\| \leq b_{\epsilon_h}$.

Unlike the common analysis and derivation process in well-known ADP related works [92], [96], here we rewrite the NN parameterized LE (5.28) into a linear in parameter (LIP) form that reads

$$\Theta = -W^{*\top}Y + \epsilon_h, \tag{5.29}$$

where $\Theta = r(x, u(x), v(x)) \in \mathbb{R}$, and $Y = \nabla\Phi(f(x) + g(x)u(x) + h(x)v(x)) \in \mathbb{R}^N$. Note that both $\Theta$ and $Y$ could be obtained from real-time data.

Given the LIP form and the measurable $Y$, $\Theta$ in (5.29), from the perspective of adaptive control, we transform the critic NN weight $W^*$ learning into a parameter estimation problem of an LIP system, where $Y$ and $W^*$ are treated as the regressor matrix and the unknown parameter vector of a LIP system, respectively. This novel transformation enables us to design a simple weight update law with guaranteed weight convergence in Section 5.2.2.

### 5.2.2 Off-Policy Weight Update Law

The ideal critic NN weight $W^*$ in (5.29) is approximated by an estimated weight $\hat{W}$ which satisfies the following relation

$$\hat{\Theta} = -\hat{W}^\top Y, \tag{5.30}$$

where $\hat{\Theta} \in \mathbb{R}$ is the estimated utility function. Denoting the weight estimation error as $\tilde{W} = \hat{W} - W^* \in \mathbb{R}^N$. Then, we get

$$\tilde{\Theta} = \Theta - \hat{\Theta} = \tilde{W}^\top Y + \epsilon_h. \tag{5.31}$$

To achieve $\hat{W} \to W^*$ and $\tilde{\Theta} \to \epsilon_h$, $\hat{W}$ should be updated to minimize $E = \frac{1}{2}\tilde{\Theta}^\top\tilde{\Theta}$. Furthermore, in order to guarantee the weight convergence while minimizing $E$, here we exploit experience data to support the online learning process. The utilized experience data could achieve the sufficient excitation required for the weight convergence. This departs from related works [19], [24] that incorporate external noises to satisfy the PE condition. Finally, we design a simple yet efficient off-policy weight update law of the critic NN that follows

$$\dot{\hat{W}} = -\Gamma k_c Y \tilde{\Theta} - \sum_{l=1}^{P} \Gamma k_e Y_l \tilde{\Theta}_l, \tag{5.32}$$

where $\tilde{\Theta} = \Theta + \hat{W}^\top Y$ according to (5.30) and (5.31). $\Gamma \in \mathbb{R}^{N \times N}$ is a constant positive definite gain matrix. $k_c, k_e \in \mathbb{R}^+$ are constant gains to balance the relative importance between current and experience data to the online learning process. $P \in \mathbb{N}^+$ is the volume of the experience buffers $\mathfrak{B}$ and $\mathfrak{E}$, i.e., the maximum number of recorded data points. $Y_l \in \mathbb{R}^N$ and $\Theta_l \in \mathbb{R}$ denote the $l$-th collected data of the corresponding experience buffers $\mathfrak{B}$ and $\mathfrak{E}$, respectively. Our developed critic NN weight update law (5.32) is in a different form comparing with the counterpart in well-known ADP related works (see [19], [92], [96] and the references therein). Our proposed weight update law (5.32) is easily implemented and enjoys guaranteed weight convergence without causing undesirable oscillations and additional control energy expenditures.

To analyse the weight convergence of the critic NN, a rank condition about the experience buffer $\mathfrak{B}$, which serves as a richness criterion of the recorded experience data, is firstly clarified in Assumption 5.5.

**Assumption 5.5.** *Given $\mathfrak{B} = [Y_1, ..., Y_P] \in \mathbb{R}^{N \times P}$, there holds rank$(\mathfrak{B}) = N$.*

Comparing with the traditional PE condition given in [64], the rank condition regarding $\mathfrak{B}$ in Assumption 5.5 provides an index about the data richness that could be checked online, which is favourable to controller designers.

Based on the aforementioned settings, the NN weight convergence proof is shown as follows.

**Theorem 5.2.** *Given Assumption 5.5, the weight learning error $\tilde{W}$ converges to a small neighbourhood around zero.*

*Proof.* Consider the following candidate Lyapunov function

$$V_{er} = \frac{1}{2} \tilde{W}^\top \Gamma^{-1} \tilde{W}. \tag{5.33}$$

The time derivative of $V_{er}$ reads

$$\begin{aligned} \dot{V}_{er} &= \tilde{W}^\top \Gamma^{-1} (-\Gamma k_c Y \tilde{\Theta} - \Gamma \sum_{l=1}^{P} k_e Y_l \tilde{\Theta}_l) \\ &= -k_c \tilde{W}^\top Y \tilde{\Theta} - \tilde{W}^\top \sum_{l=1}^{P} k_e Y_l \tilde{\Theta}_l \\ &\leq -\tilde{W}^\top B \tilde{W} + \tilde{W}^\top \epsilon_{er}, \end{aligned} \tag{5.34}$$

where $B = \sum_{l=1}^{P} k_e Y_l Y_l^\top$, and $\epsilon_{er} = -k_c Y \epsilon_h - \sum_{l=1}^{P} k_e Y_l \epsilon_{h_l}$. The boundness of $Y$ and $\epsilon_h$ results in bounded $\epsilon_{er}$, i.e., there exists $b_{\epsilon_{er}} \in \mathbb{R}^+$ such that $\|\epsilon_{er}\| \leq b_{\epsilon_{er}}$. Since $B$ is positive definite according to Assumption 5.5, (5.34) could be written as

$$\dot{V}_{er} \leq -\left\|\tilde{W}\right\| \left(\lambda_{\min}(B) \left\|\tilde{W}\right\| - b_{\epsilon_{er}}\right). \tag{5.35}$$

Therefore, $\dot{V}_{er} < 0$ if $\left\|\tilde{W}\right\| > \frac{b_{\epsilon_{er}}}{\lambda_{\min}(B)}$. Finally, it is concluded that the weight estimation error of the critic NN will converge to the residual set

$$\Omega_{\tilde{W}} = \left\{\tilde{W} | \left\|\tilde{W}\right\| \leq \frac{b_{\epsilon_{er}}}{\lambda_{\min}(B)}\right\}. \tag{5.36}$$

□

By observing (5.36), the size of $\Omega_{\tilde{W}}$ relates with the bound of $\epsilon_{er}$. As $N \to \infty$, we know that $\epsilon_h \to 0$ results in $\epsilon_{er} \to 0$. Then, we get $\dot{V}_{er} \leq -\lambda_{\min}(B)\left\|\tilde{W}\right\|^2$, i.e., $\tilde{W} \to 0$ exponentially as $t \to \infty$. Equivalently, it is guaranteed that $\hat{W}$ converges to $W^*$. Finally, in conjugation with (5.12) and (5.13), the approximate optimal control policies are obtained as

$$\hat{u}(x) = -\beta \tanh\left(\frac{1}{2\beta}R^{-1}g^\top(x)\nabla\Phi^\top(x)\hat{W}\right), \tag{5.37}$$

$$\hat{v}(x) = -\frac{1}{2\rho}h^\top(x)\nabla\Phi^\top(x)\hat{W}. \tag{5.38}$$

In the following part, the main conclusions are provided based on the off-policy weight update law (5.32) and the approximate optimal control policies (5.37), (5.38).

**Theorem 5.3.** *Consider the dynamics* (5.5)*, the off-policy weight update law of the critic NN in* (5.32)*, and the control policies* (5.37) *and* (5.38)*. Given Assumptions 5.1-5.5, for sufficiently large $N$, the approximate control policies* (5.37) *and* (5.38) *stabilize the system* (5.5)*. Moreover, the critic NN weight learning error $\tilde{W}$ is uniformly ultimately bounded.*

*Proof.* Consider the following candidate Lyapunov function

$$J = V^*(x) + \frac{1}{2}\tilde{W}^\top\Gamma^{-1}\tilde{W}. \tag{5.39}$$

Taking time derivative of (5.39) along the system (5.5) yields

$$\dot{J} = \dot{L}_V + \dot{L}_W. \tag{5.40}$$

where $\dot{L}_V = \dot{V}^*(x)$ and $\dot{L}_W = \tilde{W}^\top\Gamma^{-1}\dot{\tilde{W}}$.

The first term $\dot{L}_V$ follows

$$\begin{aligned}\dot{L}_V &= \nabla V^{*\top}(f(x) + g(x)\hat{u}(x) + h(x)\hat{v}(x)) \\ &= \nabla V^{*\top}(f(x) + g(x)u^*(x) + h(x)v^*(x)) \\ &\quad + \nabla V^{*\top}g(x)(\hat{u}(x) - u^*(x)) + \nabla V^{*\top}h(x)(\hat{v}(x) - v^*(x)).\end{aligned} \tag{5.41}$$

According to (5.16), (5.17) and (5.18), (5.41) is rewritten as

$$\begin{aligned}\dot{L}_V = &-\mathcal{L}(x) - \mathcal{W}(u^*(x)) - \rho v^*(x)^\top v^*(x) - l_M^2(x) - \rho d_M^2(x) \\ &- 2\beta R\tanh^{-1}(u^*(x)/\beta)(\hat{u}(x) - u^*(x)) - 2\rho v^{*\top}(x)(\hat{v}(x) - v^*(x)).\end{aligned} \tag{5.42}$$

Besides, we get

$$
\begin{aligned}
-2\beta R\tanh^{-1}(u^*(x)/\beta)(\hat{u}(x)-u^*(x)) &\leq \beta^2\left\|R\tanh^{-1}(u^*(x)/\beta)\right\|^2+\|\hat{u}(x)-u^*(x)\|^2\\
&\leq \beta^2\sum_{j=1}^{m}R_j^2(\tanh^{-1}(u_j^*(x)/\beta))^2+\|\hat{u}(x)-u^*(x)\|^2.
\end{aligned}
\tag{5.43}
$$

Based on (5.20)-(5.26), the following equation also establishes

$$
\begin{aligned}
&-\mathcal{W}(u^*(x))-2\beta\tanh^{-1}(u^*(x)/\beta)(\hat{u}(x)-u^*(x))\\
&\leq \beta^2\sum_{j=1}^{m}(R_j^2-R_j)(\tanh^{-1}(u_j^*(x)/\beta))^2+b_{\epsilon_t}+\|\hat{u}(x)-u^*(x)\|^2\\
&\leq \epsilon_s+\|\hat{u}(x)-u^*(x)\|^2.
\end{aligned}
\tag{5.44}
$$

Substituting (5.44) into (5.42) yields

$$
\begin{aligned}
\dot{L}_V &\leq -\mathcal{L}(x)-\rho v^*(x)^\top v^*(x)-l_M^2(x)-\rho d_M^2(x)+\epsilon_s+\|\hat{u}(x)-u^*(x)\|^2-2\rho v^{*\top}(x)(\hat{v}(x)-v^*(x))\\
&=-\mathcal{L}(x)-l_M^2(x)-\rho d_M^2(x)+\epsilon_s-\rho\hat{v}^\top(x)\hat{v}(x)+\|\hat{u}(x)-u^*(x)\|^2+\rho\|\hat{v}(x)-v^*(x)\|^2.
\end{aligned}
\tag{5.45}
$$

As for $\rho\hat{v}^\top(x)\hat{v}(x)$ in (5.45), according to (5.38), we get

$$
\begin{aligned}
\rho\hat{v}^\top(x)\hat{v}(x) &= \frac{1}{4\rho}\hat{W}^\top\nabla\Phi(x)h(x)h^\top(x)\nabla\Phi^\top(x)\hat{W}\\
&=\frac{1}{4\rho}(W^*+\tilde{W})^\top\nabla\Phi(x)h(x)h^\top(x)\nabla\Phi^\top(x)(W^*+\tilde{W})\\
&=\frac{1}{4\rho}W^{*\top}\mathscr{H}W^*+\frac{1}{4\rho}\tilde{W}^\top\mathscr{H}\tilde{W}+\frac{1}{2\rho}W^{*\top}\mathscr{H}\tilde{W}.
\end{aligned}
\tag{5.46}
$$

where $\mathscr{H}=\nabla\Phi(x)h(x)h^\top(x)\nabla\Phi^\top(x)$.

As for $\rho\|\hat{v}(x)-v^*(x)\|^2$in (5.45), according to (5.38), we get

$$
\rho\|\hat{v}(x)-v^*(x)\|^2=\rho\left\|\frac{1}{2\rho}h^\top(x)\nabla\Phi(x)\tilde{W}\right\|^2=\frac{1}{4\rho}\tilde{W}^\top\mathscr{H}\tilde{W}. \tag{5.47}
$$

For simplicity, denote $\mathscr{G}^*=\frac{1}{2\beta}R^{-1}g^\top(x)\nabla\Phi^\top(x)W^*$ and $\hat{\mathscr{G}}=\frac{1}{2\beta}R^{-1}g^\top(x)\nabla\Phi^\top(x)\hat{W}$, $\hat{\mathscr{G}}=[\hat{\mathscr{G}}_1,\cdots,\hat{\mathscr{G}}_m]\in\mathbb{R}^m$ with $\hat{\mathscr{G}}_j\in\mathbb{R}, j=1,\cdots,m$. Based on (5.12) and (5.37), the Taylor series of $\tanh(\mathscr{G}^*)$ follows

$$
\begin{aligned}
\tanh(\mathscr{G}^*) &= \tanh\Big(\hat{\mathscr{G}}\Big)+\frac{\partial\tanh\Big(\hat{\mathscr{G}}\Big)}{\partial\hat{\mathscr{G}}}(\mathscr{G}^*-\hat{\mathscr{G}})+O((\mathscr{G}^*-\hat{\mathscr{G}})^2)\\
&=\tanh\Big(\hat{\mathscr{G}}\Big)-\frac{1}{2\beta}(I_{m\times m}-\mathscr{D}(\hat{\mathscr{G}}))R^{-1}g^\top(x)\nabla\Phi^\top(x)\tilde{W}+O((\mathscr{G}^*-\hat{\mathscr{G}})^2),
\end{aligned}
\tag{5.48}
$$

where $\mathscr{D}(\hat{\mathscr{G}})=\operatorname{diag}(\tanh^2(\hat{\mathscr{G}}_1),\cdots,\tanh^2(\hat{\mathscr{G}}_m))$, $O((\mathscr{G}^*-\hat{\mathscr{G}})^2)$ is a higher order term of the Taylor series. By following [100, Lemma 1], the higher order term is bounded as

$$
\left\|O((\mathscr{G}^*-\hat{\mathscr{G}})^2)\right\|\leq 2\sqrt{m}+\frac{1}{\beta}\left\|R^{-1}\right\|g_M b_{\Phi x}\left\|\tilde{W}\right\|. \tag{5.49}
$$

Using (5.12), (5.37) and (5.48), we get

$$\begin{aligned}\hat{u}(x) - u^*(x) &= \beta(\tanh(\mathscr{G}^*) - \tanh\Big(\hat{\mathscr{G}}\Big)) + \epsilon_u^* \\ &= -\frac{1}{2}(I_{m\times m} - \mathscr{D}(\hat{\mathscr{G}}))R^{-1}g^\top(x)\nabla\Phi^\top(x)\tilde{W} + \beta O((\mathscr{G}^* - \hat{\mathscr{G}})^2) + \epsilon_u^*.\end{aligned} \tag{5.50}$$

where $\epsilon_u^* = \beta\tanh\Big(\frac{1}{2\beta}R^{-1}g^\top(x)(\nabla\Phi^\top(x)W^* + \nabla\epsilon)\Big) - \beta\tanh\Big(\frac{1}{2\beta}R^{-1}g^\top(x)\nabla\Phi^\top(x)W^*\Big)$, and assuming that it is bounded by $\|\epsilon_u^*\| \leq b_{\epsilon_u^*}$.

As for $\|\hat{u}(x) - u^*(x)\|^2$ in (5.45), since $\Big\|I_{m\times m} - \mathscr{D}(\hat{\mathscr{G}})\Big\| \leq 2$ [100], combining (5.49) with (5.50), we get

$$\begin{aligned}\|\hat{u}(x) - u^*(x)\|^2 &\leq 3\beta^2\left\|O((\mathscr{G}^* - \hat{\mathscr{G}})^2)\right\|^2 + 3\|\epsilon_u^*\|^2 + 3\left\|-\frac{1}{2}(I_{m\times m} - \mathscr{D}(\hat{\mathscr{G}}))R^{-1}g^\top(x)\nabla\Phi^\top(x)\tilde{W}\right\|^2 \\ &\leq 6\left\|R^{-1}\right\|^2 g_M^2 b_{\Phi x}^2\left\|\tilde{W}\right\|^2 + 12m\beta^2 + 3b_{\epsilon_u^*}^2 + 12\beta\sqrt{m}\left\|R^{-1}\right\| g_M b_{\Phi x}\left\|\tilde{W}\right\|.\end{aligned} \tag{5.51}$$

Substituting (5.46), (5.47), (5.51) into (5.45) yields

$$\begin{aligned}\dot{L}_V \leq &-\frac{1}{2\rho}W^{*\top}\mathscr{H}\tilde{W} - \mathcal{L}(x) - l_M^2(x) - \rho d_M^2(x) - \frac{1}{4\rho}W^{*\top}\mathscr{H}W^* + \epsilon_s \\ &+ 6\left\|R^{-1}\right\|^2 g_M^2 b_{\Phi x}^2\left\|\tilde{W}\right\|^2 + 12m\beta^2 + 3b_{\epsilon_u^*}^2 + 12\beta\sqrt{m}\left\|R^{-1}\right\| g_M b_{\Phi x}\left\|\tilde{W}\right\|.\end{aligned} \tag{5.52}$$

As for the second term $\dot{L}_W$, based on (5.32) and (5.34),

$$\dot{L}_W \leq -\tilde{W}^\top X\tilde{W} + \tilde{W}^\top\epsilon_{er}. \tag{5.53}$$

Finally, as for $\dot{J}$, substituting (5.52) and (5.53) into (5.40), and based on the fact that $\|W^*\| \leq b_{W^*}$, $\|\nabla\Phi(x)\| \leq b_{\Phi_x}$, $\|h(x)\| \leq h_M$, we get

$$\begin{aligned}\dot{J} \leq &-\mathcal{L}(x) - l_M^2(x) - \rho d_M^2(x) - \frac{1}{4\rho}W^{*\top}\mathscr{H}W^* - \tilde{W}^\top X\tilde{W} + M\tilde{W} \\ &+ 6\left\|R^{-1}\right\|^2 g_M^2 b_{\Phi x}^2\left\|\tilde{W}\right\|^2 + 12\beta\sqrt{m}\left\|R^{-1}\right\| g_M^2 b_{\Phi x}^2\left\|\tilde{W}\right\| + 12m\beta^2 + 3b_{\epsilon_u^*}^2 + \epsilon_s \\ \leq &-\mathcal{L}(x) - l_M^2(x) - \rho d_M^2(x) - \frac{1}{4\rho}W^{*\top}\mathscr{H}W^* - (\lambda_{\min}(B) - 6\left\|R^{-1}\right\|^2 g_M^2 b_{\Phi_x}^2)\left\|\tilde{W}\right\|^2 \\ &+ 12m\beta^2 + 3b_{\epsilon_u^*}^2 + (12\beta\sqrt{m}\left\|R^{-1}\right\| g_M^2 b_{\Phi x}^2 + b_M)\left\|\tilde{W}\right\| + \epsilon_s \\ = &-\mathcal{A} - \mathcal{B}\left\|\tilde{W}\right\|^2 + \mathcal{C}\left\|\tilde{W}\right\| + \mathcal{D},\end{aligned} \tag{5.54}$$

where $M = \epsilon_{er} - \frac{1}{2\rho}W^{*\top}\mathscr{H}$, and there exists $b_M = b_{\epsilon_{er}} + \frac{1}{2\rho}b_{\Phi x}^2 h_M^2 b_{W^*} \in \mathbb{R}^+$ such that $\|M\| \leq b_M$; $\mathcal{A} = \mathcal{L}(x) + l_M^2(x) + \rho d_M^2(x) + \frac{1}{4\rho}W^{*\top}\mathscr{H}W^*$ is positive definite; $\mathcal{B} = \lambda_{\min}(B) - 6\|R^{-1}\|^2 g_M^2 b_{\Phi_x}^2$, $\mathcal{C} = 12\beta\sqrt{m}\|R^{-1}\| g_M^2 b_{\Phi x}^2 + b_M$ and $\mathcal{D} = 12m\beta^2 + 3b_{\epsilon_u^*}^2 + \epsilon_s$.

Let the parameters be chosen such that $\mathcal{B} > 0$. Since $\mathcal{A}$ is positive definite, the above Lyapunov derivative is negative if

$$\left\|\tilde{W}\right\| > \frac{\mathcal{C}}{2\mathcal{B}} + \sqrt{\frac{\mathcal{C}^2}{4\mathcal{B}^2} + \frac{\mathcal{D}}{\mathcal{B}}}. \tag{5.55}$$

**Algorithm 4** Online Prioritized Experience Replay Algorithm

```
Input: Iteration index: $n_r$; Buffer size: $P$; Threshold: $\xi$.
Output: Experience buffers: $\mathfrak{B}$, $\mathfrak{C}$.
1: if $n_r \leq P$ then
2:     Record current $Y$, $\Theta$ into $\mathfrak{B}$, $\mathfrak{C}$ respectively.
3: else
4:     if $\|W_{n_r} - W_{n_r-1}\| > \xi$ then
5:         Record prioritized $Y$, $\Theta$ leading to high $\lambda_{\min}(\mathfrak{B})$.
6:     else
7:         Record current $Y$, $\Theta$ sequentially to update $\mathfrak{B}$,$\mathfrak{C}$.
8:     end if
9: end if
```

Thus, the critic weight learning error converges to the residual set defined as

$$\tilde{\Omega}_{\tilde{W}} = \left\{ \tilde{W} | \left\| \tilde{W} \right\| \leq \frac{\mathcal{C}}{2\mathcal{B}} + \sqrt{\frac{\mathcal{C}^2}{4\mathcal{B}^2} + \frac{\mathcal{D}}{\mathcal{B}}} \right\}. \tag{5.56}$$

□

Assumption 5.5 used in Theorem 5.3 is the prerequisite to ensure that $\hat{W}$ converges to $W^*$. The guaranteed weight convergence enables us to directly apply $\hat{W}$ in (5.32) to construct the approximate optimal control policies (5.37), (5.38). Assumption 5.5 is not restrictive and could be satisfied by the algorithms proposed in the next subsection.

### 5.2.3 Online and Offline Experience Buffer Construction

To get rich enough experience data to satisfy Assumption 5.5, given the sampling deficiency problem of the sequent way of data usage in existing ADP related works [39], [101], [102] and inspired by the concurrent learning technique developed for system identification [65], here we design both online and offline principled methods to provide the sufficient rich experience data. These recorded informative experience data are then relayed to the weight update law to achieve the required excitation for the guaranteed critic NN weight convergence.

#### A. Online PER Algorithm

Before the estimated weight converges (line 4-5), Algorithm 4 chooses the minimum eigenvalue (i.e., $\lambda_{\min}(\mathfrak{B})$) as the priority scheme to filter experience data $Y$ and $\Theta$ recorded into the experience buffers $\mathfrak{B}$ and $\mathfrak{C}$, respectively. Here the prioritized criterion is different from ones used in existing PER algorithms [103]. We prefer experience data accompanied with a larger $\lambda_{\min}(\mathfrak{B})$ given the facts that: a) a nonzero $\lambda_{\min}(\mathfrak{B})$ ensures that $rank(\mathfrak{B}) = N$ in Assumption 5.5 holds [65], [104], i.e., the convergence of $\hat{W}$ to $W^*$ is guaranteed; b) according to (5.35) and (5.36), a larger $\lambda_{\min}(\mathfrak{B})$ leads to a faster weight convergence rate and a smaller residual set. Although efficient, the priority scheme $\lambda_{\min}(\mathfrak{B})$ accompanies with additional computation loads. Thus, once the convergence is achieved (line 6-7), i.e., we have obtained sufficient excitation, we alternate to a low-cost mode where recent data are sequentially recorded. This kind of cyclic replacement way of data usage enjoys robustness

**Algorithm 5** Offline Experience Buffer Construction Algorithm

```
Input: Mesh size : $\delta \in \mathbb{R}^n$, or data point number: $c \in \mathbb{R}^n$;
    $A = [\underline{A}, \overline{A}]$ with $\underline{A}, \overline{A} \in \mathbb{R}^n$; Empty sets: $\mathcal{X} \in \mathbb{R}^{n \times d}$.
Output: $\mathcal{F}$; $\mathcal{G}$; $\mathcal{H}$; $\mathcal{K}$; $\mathcal{R}$; $P$.
Sampling: $\mathcal{X}$; $P = \prod_{i=1}^{n} (\overline{A}_i - \underline{A}_i)/\delta_i$, or $\prod_{i=1}^{n} c_i$.
Data collection: $\mathcal{F} \leftarrow \nabla\Phi^\top(\mathcal{X}) f(\mathcal{X})$; $\mathcal{R} \leftarrow r_d(\mathcal{X})$;
    $\mathcal{G} \leftarrow \nabla\Phi^\top(\mathcal{X}) g(\mathcal{X})$; $\mathcal{H} \leftarrow \nabla\Phi^\top(\mathcal{X}) h(\mathcal{X})$; $\mathcal{K} \leftarrow \mathcal{L}(\mathcal{X})$
```

to a dynamic environment since collected real-time data could reflect environmental changes in time. Unlike standard methods that first construct a huge experience buffer and then sample partial data [105], to reduce computation loads and relieve hardware requirements, we directly build experience buffers with a limited buffer size $P$ here, and all of the recorded experience data are replayed to the critic NN for the online weight learning. The buffer size $P$ is a hyper-parameter that requires careful tuning. In order to satisfy Assumption 5.5, $P$ is selected such that $P \geq N$ holds.

### B. Offline Experience Buffer Construction Algorithm

Algorithm 5 aims to construct experience buffers $\mathcal{F}, \mathcal{G}, \mathcal{H}, \mathcal{K}, \mathcal{R} \in \mathbb{R}^{N \times P}$ full with offline recorded experience data, which are then used to support the online weight learning. For simplicity, here the offline experience data are generated from pre-simulation within the given operation region $A$. Specifically, for $i$-th dimension of an allowable operation region $A_i \in \mathbb{R}$, we sample data isometrically with a defined mesh size $\delta_i \in \mathbb{R}^+$, or a prior given number $c_i \in \mathbb{N}^+$. The resulting sampling state space is denoted as $\mathcal{X}$. Note that ← in Algorithm 5 means that experience data are recorded into corresponding experience buffers. Rather than sampling partial data from the offline constructed experience buffers based on a uniform or a prioritized way [105], we replay all the offline recorded experience data in an average way for the online weight learning. Thereby, the off-policy weight update law (5.32) based on Algorithm 5 is redesigned as

$$\dot{\hat{W}} = -\Gamma k_c Y \tilde{\Theta} - \frac{1}{P} \sum_{l=1}^{P} \Gamma k_e Y_l \tilde{\Theta}_l. \tag{5.57}$$

The implementation of using the offline recorded experience data to support the online learning process enjoys two advantages: the rank condition in Assumption 5.5 is easily satisfied, and the possible influence of data noises is offset by averaging. It is worth mentioning that the mere exploitation of offline recorded data cannot tackle a dynamic environment well. Thus, during the online operation, the offline experience data recorded into experience buffers $\mathcal{F}, \mathcal{G}, \mathcal{H}, \mathcal{K}, \mathcal{R}$ will be sequentially replaced with online counterparts.

**Remark 5.1.** *The adopted simple three-layer NNs provides us with opportunities to revisit the ER technique and investigate principled ways to exploit experience data to support the online learning process. This is difficult in deep RL field because the complexity of deep NNs hinders researchers from understanding the mechanism of the ER technique [105].*

## 5.3 Numerical Simulation

This section numerically validates the effectiveness of the developed off-policy weight update laws (5.32), (5.57), the approximate optimal control policy (5.37), and Algorithms 4-5. Firstly, we consider an optimal regulation problem (ORP) of a nonlinear system [19] in Section 5.3.1. This ORP serves as a benchmark to prove that both Algorithm 4 based (5.32) and Algorithm 5 based (5.57) enable the estimated critic NN weight to converge to the actual value, which is marginally considered in existing single critic structure related works [95], [106]. Then, a reach task for a 2-DoF robot manipulator [48] is considered in Section 5.3.2 to validate the real-time performance of our proposed control framework. Finally, our developed RS-SP terms in Definition 5.2 are extended to achieve optimal tracking control with prescribed performance in Section 5.3.3.

### 5.3.1 ORP of Benchmark Nonlinear System

To validate that based on our proposed off-policy weight update laws (5.32), (5.57), and Algorithms 4-5, the estimated weight guarantees convergence to the actual value, a benchmark problem [19] is investigated here. Note that only an ORP without considering disturbances nor input/state constraints is investigated here. Otherwise, the actual value of the NN weight is unknown. The benchmark continuous time nonlinear system is given as

$$\dot{x} = f(x) + g(x)u, \quad x \in \mathbb{R}^2,$$

where $f(x) = \Big[-x_1 + x_2, -0.5x_1 - 0.5x_2(1 - (\cos 2x_1 + 2)^2)\Big]^\top$, $g(x) = \Big[0, \cos(2x_1) + 2\Big]^\top$. The standard quadratic cost function follows

$$V(x) = \int_0^\infty x^\top Q x + u^\top R u \, dt,$$

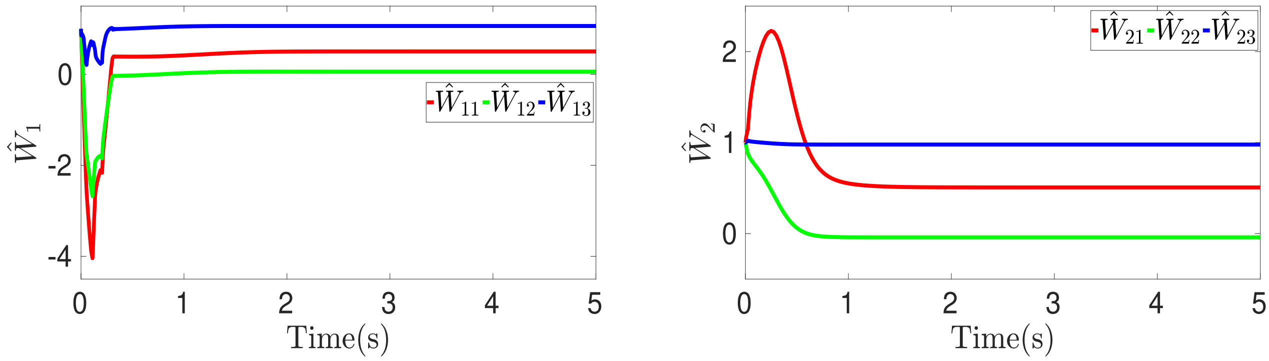
where $Q = I_{2\times 2}$ and $R = 1$. Following the method in [107], we obtain the optimal value function as $V^* = 0.5x_1^2 + x_2^2$. Thus, the optimal weight follows $W^* = [0.5, 0, 1]^\top$ by choosing the activation function as $\Phi(x) = [x_1^2, x_1x_2, x_2^2]^\top$. Initial values are set as $x(0) = [1, 1]^\top$, $\hat{u}(0) = 0$. For the off-policy weight update law (5.32) based on Algorithm 4, we choose $P = 5$, $k_c = 1$, $k_e = 1$, $\Gamma = \mathrm{diag}(2, 1.4, 1)$, and $\xi = 10^{-3}$. It is displayed in Figure 5.4a that after 1 s, the estimated weight converges to

$$\hat{W}_1 = [0.5040, 0.0592, 1.0625]^\top.$$

Regarding the weight update law (5.57) under Algorithm 5, we start with constructing offline experience buffers by sampling 10 data points separately for $x_1 \in A_1 = [-2, 2]$, $x_2 \in A_2 = [-4, 4]$. $P = 100$, $k_c = 1$, $k_e = 1$, and $\Gamma = \mathrm{diag}(5, 0.5, 0.01)$ are chosen during the online operation. As displayed in Figure 5.4b, the estimated weight converges to

$$\hat{W}_2 = [0.50721, -0.0417, 0.9783]^\top.$$

Thus, it is concluded that the weight update laws (5.32), (5.57) under Algorithms 4-5 ensure that $\hat{W}$ converges to $W^*$. Comparing with the results shown in [19], our proposed off-policy weight update laws enable $\hat{W}$ converge to $W^*$ with a fast speed without incorporating external noises to satisfy the PE condition.



(a) $\hat{W}_1$ under the online Algorithm 4.

(b) $\hat{W}_2$ under the offline Algorithm 5.

Figure 5.4: The trajectories of the estimated critic NN weight.

### 5.3.2 CRSP of Robot Manipulator

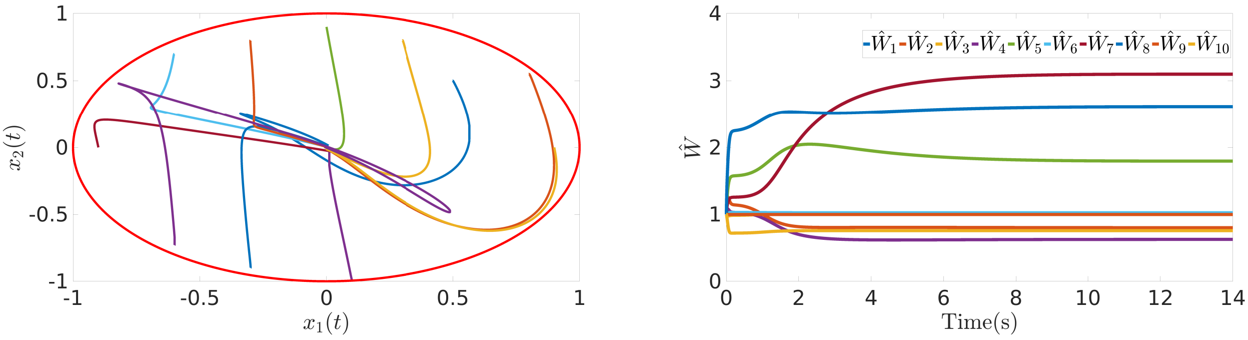
To further demonstrate the real-time performance of our proposed control framework, a reach task of a 2-DoF robot manipulator is considered here. In particular, a robot starting from multiple initial positions is driven to reach the desired point under disturbances, input and state constraints. The model of the robot manipulator follows [48]

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} = \tau,$$

where $q = [q_1, q_2]^\top \in \mathbb{R}^2$. The explicit model knowledge about $M(q)$, $C(q, \dot{q})$ is referred to Section 2.5.1 of Chapter 2.

Let $x = [x_1, x_2, x_3, x_4]^\top = [q_1, q_2, \dot{q}_1, \dot{q}_2]^\top \in \mathbb{R}^4$, the robot dynamics could be written in the state-space form as (5.1), where $f(x) = [x_3, x_4, (M^{-1}(-C)[x_3, x_4]^\top)^\top]^\top$, and $g(x) = [[0,0]^\top, [0,0]^\top, (M^{-1})^\top]^\top$. Besides, with $k(x) = [[1,0]^\top, [0,1]^\top, 0_{2\times 2}]^\top$, we assume that the robot suffers a disturbance $d(x) = [\delta_1 x_1 \sin(x_2), \delta_2 x_2 \cos(x_1)]^\top$ , where $\delta_1, \delta_2 \in [-1, 1]$. Given $\|d(x)\| \le \|x\|$, and $g^\dagger(x)k(x)d(x) = 0$, Assumptions 5.2-5.3 are satisfied by setting $d_M(x) = \|x\|$, and $\|l_M(x)\| = 0$. The input constraints are considered as $\mathbb{U}_j = \{u_j \in \mathbb{R} : |u_j| \le 3\}$, $j = 1, 2$. A circular state constraint $\mathbb{X} = \{x \in \mathbb{R}^2 : h(x_1, x_2) = x_1^2 + x_2^2 - 1 < 0\}$ is considered here. To approximate the value function well, we choose the activation function as $\Phi(x) = [x_1^2, x_2^2, x_3^2, x_4^2, x_1x_2, x_1x_3, x_1x_4, x_2x_3, x_2x_4, x_3x_4]^\top$, and the parameters for the weight update law (5.32) are set as $P = 15$, $k_c = 0.2$, $k_e = 0.01$, $\Gamma = I_{10\times 10}$, and $\xi = 10^{-3}$.

To fully demonstrate the effectiveness of our method to address state constraints even under input saturation and disturbances, the robot joint trajectories under multiple initial positions are shown in Figure 5.5a. We observe that the robot is driven to reach the desired point (i.e., zero point) while obeying the predefined state constraints. It is shown that when states approach to the constraint boundary, they will be driven back to safe states under our proposed method. As displayed in Figure 5.5b, the weight convergence result achieves at $t = 12$ s under our proposed weight update law (5.32). The aforementioned simulation results based on a 2-DoF robot manipulator validate that the off-policy weight update law (5.32) and the approximate optimal control policy (5.37) fulfill real-time requirements for practical applications.



(a) The trajectories of $x_1$ and $x_2$.

(b) The trajectories of $\hat{W}$.

Figure 5.5: The phase plot of states $x_1$ and $x_2$ under different initial values and the trajectory of the estimated weight $\hat{W}$ of the 2-DoF robot manipulator.

### 5.3.3 Optimal Tracking Control with Prescribed Performance

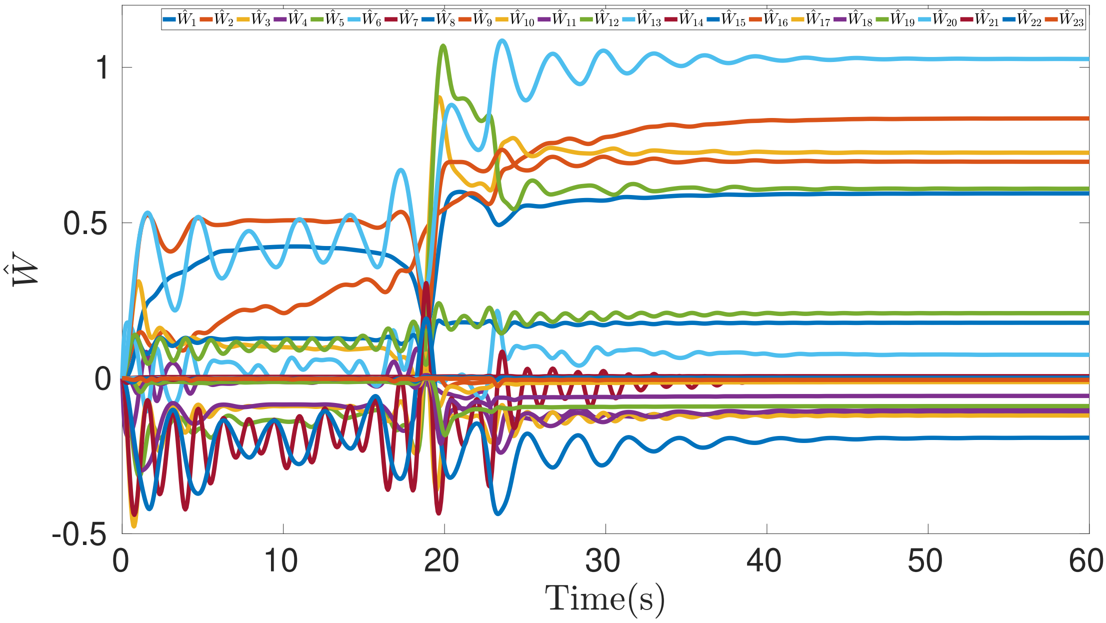


Figure 5.6: The weight convergence result of $\hat{W}$ for the PP-OTCP case.

This subsection extends our developed RS-SP terms to the optimal tracking control problem (OTCP) of the 2-DoF robot manipulator used in Section 5.3.2. The reference trajectory follows $x_r(t) = [0.5\cos(2t), \cos t, -\sin(2t), -\sin(t)]^\top$. The utilized RS-SP terms contribute to achieve prescribed performance of the tracking error $e = x - x_r \in \mathbb{R}^4$. The simulation is conducted based on the tracking control scheme illustrated in our work [51].

To encode the requirements of the tracking error, we choose the following prescribed performance function (PPF):

$$\rho_i(t) = (60\pi/180 - 3\pi/180)e^{-0.1t} + 3\pi/180, \quad i = 1, 2, 3, 4.$$

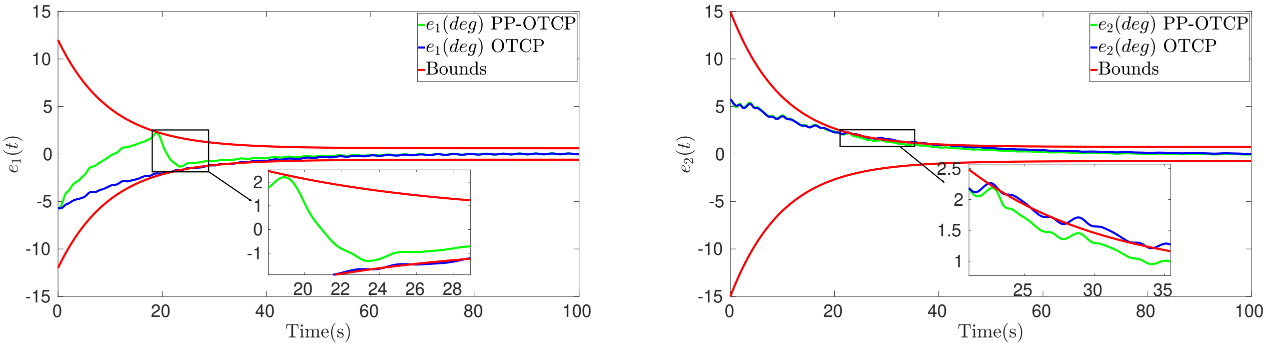
The PPF inspired RS-SP terms are utilized to construct the state penalty function

$$\mathcal{L} = \sum_{i=1}^{4} k_i \log \frac{\alpha_i^2}{\alpha_i^2 - \zeta_i^2} + h_i \log \frac{\beta_i^2}{\beta_i^2 - \delta_i^2},$$

where $\zeta_i = e_i/\rho_i$, $\delta_i = x_{r_i}/\rho_i$, and $k_i$, $h_i$ are risk awareness parameters to be designed. For simulation, we choose $k_1 = 1$, $\alpha_1 = 0.20$; $k_2 = 0.3$, $\alpha_2 = 0.25$; $k_3 = 1$, $\alpha_3 = 0.25$; $k_4 = 1$, $\alpha_4 = 0.25$; and $h_i = 0.01, \beta_i = 10, i = 1, 2, 3, 4$. Denoting $\eta = [e^\top, x_r^\top]^\top \in \mathbb{R}^8$, the chosen basis set $\Phi(\eta) \in \mathbb{R}^{23}$ to approximate the value function reads

$$\Phi(\eta) = \frac{1}{2}[\eta_1^2, \eta_2^2, 2\eta_1\eta_3, 2\eta_1\eta_4, 2\eta_2\eta_3, 2\eta_2\eta_4, \eta_1^2\eta_2^2, \eta_1^2\eta_5^2, \eta_1^2\eta_6^2, \eta_1^2\eta_7^2, \eta_1^2\eta_8^2, \eta_2^2\eta_5^2, \\ \eta_2^2\eta_6^2, \eta_2^2\eta_7^2, \eta_2^2\eta_8^2, \eta_3^2\eta_5^2, \eta_3^2\eta_6^2, \eta_3^2\eta_7^2, \eta_3^2\eta_8^2, \eta_4^2\eta_5^2, \eta_4^2\eta_6^2, \eta_4^2\eta_7^2, \eta_4^2\eta_8^2]^\top.$$

As shown in Figure 5.6, the weight convergence result achieves after 50 s. The tracking error comparison results displayed in Figure 5.7 validate the effectiveness of using $\mathcal{L}$ to achieve the tracking control with prescribed performance described by the PPF. In particular, the trajectories of $e_1$ and $e_2$ based on the common quadratic cost function [20] (termed as OTCP in Figure 5.7) violate the boundaries of the PPF. However, our proposed method (termed as PP-OTCP in Figure 5.7) efficiently drives the robot manipulator to track the reference trajectory $x_r$ precisely and satisfy the performance requirements described by the PPF.



(a) Trajectories of $e_1$.

(b) Trajectories of $e_2$.

Figure 5.7: The comparison results of the tracking errors.

## 5.4 Summary

An off-policy risk-sensitive RL-based control framework is proposed to stabilize a nonlinear system that subjects to additive disturbances, input and state constraints. The pseudo control and the resulting auxiliary system are firstly introduced to address additive disturbances under an optimization framework. Then, risk-sensitive input and state penalty terms, incorporated into the value function as optimization criteria, allow us to tackle both input and state constraints in a long time-horizon. This helps to avoid abrupt changes of control inputs that are unfavourable for the online learning process. The transformed OCP is approximately solved by a single critic learning structure with our developed off-policy weight update law. Multiple numerical comparison simulations validate the effectiveness of our developed control framework. One interesting point is that risk-sensitive state penalty terms allow us to realize the prescribed performances for full states of the optimal tracking control problem.

Our adopted single critic learning structure leads to computation simplicity and eliminates approximation errors caused by an actor NN. Furthermore, the exploitation of experience data to guarantee the weight convergence enables the proposed control strategy to be applicable to practical applications. However, the proposed control framework requires the knowledge of a nominal dynamics, which is not always available in practical applications. The future work aims to develop a low-cost model-free control strategy while preserving rigorous stability analysis.

# 6 Safe Approximate Optimal Control Through Reinforcement Learning and Safety Filter

The previous Chapter 5 requires a known disturbance bound to provide the robust safety guarantee. However, the utilized disturbance bound often results in controllers with conservative behaviors. Besides, Chapter 5 interprets safe regions as convex constraints, and construct associated RS-SP terms to encourage system states to remain at predetermined safe regions. However, computing these safe regions in advance is computationally expensive.

This chapter attempts to develop a novel RL augmented control approach, termed as incremental adaptive dynamic programming (IADP), to solve the problems mentioned above. In particular, measured input-state signals are used to facilitate model-free control. Thereby, the knowledge of dynamics and disturbances are avoided. The safety issue is investigated through a different perspective. We first formulate unsafe regions as convex constraints, then employ barrier functions to encode unsafe regions that should be stay away. The organization of this chapter is as follows. The problem formulation of the robust stabilization problem is first provided in Section 6.1. Then, measured input-state data is leveraged to get an equivalent incremental dynamics (no explicit model knowledge is used) to the original investigated system. Thereby, we sidestep the online identification process, as well as its accompanying computation complexity and parameter tuning efforts. Thereafter, the resulting incremental dynamics serves as a basis to allow the design of the model-free approximate optimal incremental control strategy in Section 6.2. Furthermore, we design a safety filter to ensure safety and achieve the trade-off between safety and performance under a satisfying framework in Section 6.3. Numerical simulation results shown in Section 6.4 demonstrate the effectiveness and the superiority of our developed approach. Finally, Section 6.5 summarizes this chapter.

## 6.1 Problem Formulation

Considering the following continuous time control-affine nonlinear system:

$$\dot{x} = f(x) + g(x)u(x) + d(t), \tag{6.1}$$

where $x \in \mathbb{R}^n$, $u(x) \in \mathbb{R}^m$ are system states and inputs, respectively. $f(x) : \mathbb{R}^n \to \mathbb{R}^n$, $g(x) : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ are continuous and locally Lipschitz drift and input dynamics. $d(t) \in \mathbb{R}^n$ represents a bounded time-varying external disturbance. Assume that no knowledge of dynamics (6.1) is available except for the dimensions of system states and inputs.

The main objective is to tackle the robust stabilization problem of the highly uncertain dynamics (6.1) operating in a disturbed environment, which is formulated as follows.

**Problem 6.1.** *Design a control strategy $u(x)$ such that the system* (6.1) *perturbed by a bounded disturbance $d(t)$ is stable under input saturation $\mathbb{U}_j = \{u_j \in \mathbb{R} : |u_j| \leq \beta\}, j = 1, \cdots, m$, where $\beta \in \mathbb{R}^+$ is a known saturation bound.*

**Remark 6.1.** *Although the explicit form of the controlled plant* (6.1) *is provided here, which is introduced for the analytical purpose and facilitates the controller design as well as the stability analysis in the following sections, our developed control approach relies on neither model parameters nor environmental information.*

### 6.1.1 Incremental Dynamics

The highly uncertain dynamics (6.1) cannot be directly used to design a controller to solve Problem 6.1. Therefore, based on measured input-state data, this section leverages the TDE technique [108], [109] to get an incremental dynamics that is an equivalent of (6.1). This formulated incremental dynamics reflects the system response of the controlled plant (6.1) without using explicit model parameters, or preceding identification procedures. Here, the attempt to relieve dependence on the accurate knowledge of dynamics departs from existing works where additional computation-intensive tools such as NNs [24]–[26], fuzzy models [110], GP [27], or observers [28] are required to address model uncertainties and/or environmental disturbances. The constructed incremental dynamics in this section serves as a basis for the development of the desired model-free control strategy and the rigorous closed-loop system stability analysis in the following sections.

Before proceeding, the following assumption is provided to facilitate the formulation of an incremental dynamics.

**Assumption 6.1.** *[20] The input dynamics $g = [g_1, g_2, \cdots, g_m]$ is bounded, and its columns $g_1, g_2, \cdots, g_m \in \mathbb{R}^n$ are linearly independent. The function $g^\dagger = (g^\top g)^{-1} g^\top : \mathbb{R}^n \to \mathbb{R}^{m\times n}$ is bounded and locally Lipschitz continuous.*

**Remark 6.2.** *Assumption 6.1 is common in ADP related works [20], [111]. Here, $g(x)$ is assumed to be full column rank such that its pseudo inverse $g^\dagger$ could be expressed as a simple algebraic formula (the inverse of $g^\top(x)g(x)$ exists). The introduced $g^\dagger$ is used to extend the TDE method usually applied to the E-L equation [108], [109] to the control-affine nonlinear system* (6.1)*. Note that it is a common assumption that the input dynamics $g$ is bounded. This property is widely observed in many physical systems, such as robot manipulator systems [48], vehicle dynamics [112], and aircraft models [113].*

To get the incremental dynamics, we start with introducing a constant matrix $\bar{g} \in \mathbb{R}^{n\times m}$ and multiply $\bar{g}^\dagger$ on the dynamics (6.1),

$$\bar{g}^\dagger \dot{x} = \bar{g}^\dagger f(x) + \bar{g}^\dagger g(x)u(x) + \bar{g}^\dagger d(t) = H(x, \dot{x}) + u(x), \tag{6.2}$$

where $H(x, \dot{x}) = (\bar{g}^\dagger - g^\dagger(x))\dot{x} + g^\dagger(x)f(x) + g^\dagger(x)d(t) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^m$. It is a lump term that embodies all the unknown model knowledge (i.e., $f(x)$, $g(x)$) as well as external disturbances (i.e., $d(t)$).

Then, with a sufficiently high sampling rate, based on the TDE technique [108], [109], the unknown $H(x, \dot{x})$ in (6.2) could be estimated by time-delayed signals as

$$\hat{H}(x, \dot{x}) = H(x_0, \dot{x}_0) = \bar{g}^\dagger \dot{x}_0 - u_0, \tag{6.3}$$

where $x_0 = x(t-L)$, $u_0 = u(x(t-L))$. $L \in \mathbb{R}^+$ is the delay time chosen as one or several sampling periods in practical digital implementations. Given that the smallest achievable $L$

in digital devices is the sampling period [114], thus we finally take the delay time $L$ to be the same as the sampling period to get an accurate estimation of $H(x, \dot{x})$ in (6.3). In other words, $x_0$, $u_0$ are the values of states and inputs at the previous sampling period.

Finally, substituting (6.3) into (6.2), we get the incremental dynamics as

$$\Delta\dot{x} = \bar{g}\Delta u + \bar{g}\xi, \tag{6.4}$$

where $\Delta\dot{x} = \dot{x} - \dot{x}_0 \in \mathbb{R}^n$, and $\Delta u = u(x) - u_0 \in \mathbb{R}^m$ are incremental states and control inputs, respectively. $\xi = H(x, \dot{x}) - \hat{H}(x, \dot{x}) \in \mathbb{R}^m$ denotes the so-called TDE error, which is proved to be bounded as given in Lemma 6.1. Here, with a predefined $\bar{g}$, the measured input-state data (i.e., $\dot{x}$, $\dot{x}_0$, $u$, and $u_0$) are adopted to reflect the system response in an incremental way without using model or environmental information.

**Remark 6.3.** *The so-called sufficiently high sampling rate, which is a prerequisite for estimating the unknown $H(x, \dot{x})$ by reusing past measured input-state data, can be chosen as the value that is larger than* 30 *times the system bandwidth [114], [115]. In this setting, a digital control system can be regarded as a continuous system so that $H(x, \dot{x})$ in* (6.2) *does not vary significantly during the sampling period. Thus, the TDE error $\xi$ in* (6.4) *is sufficiently small.*

**Remark 6.4.** *The TDE technique, which is usually used in the robotic field [108], [109], is extended to the continuous time control-affine nonlinear system* (6.1) *in this section. From a practical perspective, the applied TDE technique enables us to switch from the requirement of accurate mathematical models to sensor capabilities of providing accurate measurements of $\Delta\dot{x}$ (constructed from $\dot{x}$ and $\dot{x}_0$) and $\Delta u$ (constructed from $u(x)$ and $u_0$). Though the derived incremental dynamics* (6.4) *suffers a practical utility problem given that state derivatives, or even partial state variables are not directly measurable for certain cases, authors argue that state derivative estimation techniques [116], [117], numerical differential techniques [118], or state observer [119] could help. These potential solutions mentioned above deviate from the main objective of this chapter and thus remain as future works.*

However, although an equivalent of (6.1) is provided in (6.4) without using explicit knowledge of dynamics, the unknown TDE error $\xi$ hinders us to directly utilize (6.4) to design controllers. Therefore, a method will be developed to address the TDE error $\xi$ in the next subsection. Before proceeding, here we first provide the theoretical analysis about the boundness property of $\xi$, which facilitates the method to tackle the TDE error $\xi$ under an optimization framework in Section 6.1.2.

**Lemma 6.1.** *Given a sufficiently high sampling rate, $\exists\bar{\xi} \in \mathbb{R}^+$, there holds $\|\xi\| \leq \bar{\xi}$.*

*Proof.* Combining (6.2) with (6.3), the TDE error follows

$$\begin{aligned} \xi &= H(x, \dot{x}) - \hat{H}(x, \dot{x}) = H(x, \dot{x}) - H(x_0, \dot{x}_0) \\ &= (\bar{g}^\dagger - g^\dagger(x))(\dot{x} - \dot{x}_0) + (g_0^\dagger - g^\dagger(x))\dot{x}_0 + g^\dagger(x)f(x) - g_0^\dagger f_0 + g^\dagger(x)d(t) - g_0^\dagger d_0 \\ &= (\bar{g}^\dagger - g^\dagger(x))\Delta\dot{x} + (g_0^\dagger - g^\dagger(x))\dot{x}_0 + g^\dagger(x)(f(x) - f_0) + (g^\dagger(x) - g_0^\dagger)f_0 \\ &\quad + g^\dagger(x)(d(t) - d_0) + (g^\dagger(x) - g_0^\dagger)d_0. \end{aligned} \tag{6.5}$$

Besides, based on the system (6.1), we get

$$\begin{aligned} \Delta\dot{x} &= f(x) + g(x)u(x) + d(t) - f_0 - g_0u_0 - d_0 \\ &= g(x)\Delta u + (g(x) - g_0)u_0 + f(x) - f_0 + d(t) - d_0. \end{aligned} \tag{6.6}$$

Then, substituting (6.6) into (6.5) yields

$$\begin{aligned}\xi &= (\bar{g}^{\dagger} - g^{\dagger}(x))g(x)\Delta u + (\bar{g}^{\dagger} - g^{\dagger}(x))[(g(x) - g_0)u_0 + f(x) - f_0 + d(t) - d_0] + (g_0^{\dagger} - g^{\dagger}(x))\dot{x}_0 \\ &\quad + g^{\dagger}(x)(f(x) - f_0) + (g^{\dagger}(x) - g_0^{\dagger})f_0 + g^{\dagger}(x)(d(t) - d_0) + (g^{\dagger}(x) - g_0^{\dagger})d_0 \\ &= (\bar{g}^{\dagger}g(x) - I_{m\times m})\Delta u + \delta_1, \end{aligned} \tag{6.7}$$

where $\delta_1 = \bar{g}^{\dagger}(g(x) - g_0)u_0 + \bar{g}^{\dagger}(f(x) - f_0) + \bar{g}^{\dagger}(d(t) - d_0)$.

For a sufficiently high sampling rate, the gap between successive states is sufficiently small. Thus, it is reasonable to assume that there exists a positive constant $\bar{\delta}_1 \in \mathbb{R}^+$ such that $\|\delta_1\| \leq \bar{\delta}_1$. In addition, the bounded control input $u$ implies that $\|\Delta u\| \leq 2\beta$ holds. By choosing a suitable $\bar{g}$ such that $\left\|\bar{g}^{\dagger}g(x) - I_{m\times m}\right\| \leq c$ establishes, we could get

$$\|\xi\| \leq \left\|\bar{g}^{\dagger}g(x) - I_{m\times m}\right\| \|\Delta u\| + \|\delta_1\| \leq c\|\Delta u\| + \bar{\delta}_1 \leq 2\beta c + \bar{\delta}_1 = \bar{\xi}. \tag{6.8}$$

□

**Remark 6.5.** *By using the Taylor series expansion based incremental control technique, previous works [120]–[124] attempt to provide the incremental dynamics by offering the first-order approximation of $\dot{x}$ in the neighbourhood of $[x_0, u_0]$. It follows*

$$\begin{aligned}\dot{x} &= f(x) + g(x)u(x) \\ &= f_0 + g_0u_0 + \frac{\partial[f(x) + g(x)u(x)]}{\partial x}|_{x=x_0,u=u_0}(x - x_0) \\ &\quad + \frac{\partial[f(x) + g(x)u(x)]}{\partial u}|_{x=x_0,u=u_0}(u - u_0) + \mathcal{H.O.T}. \\ &\cong \dot{x}_0 + F[x_0, u_0]\Delta x + G[x_0, u_0]\Delta u,\end{aligned}$$

*where $F[x_0, u_0] = [\partial(f(x) + g(x)u(x))/\partial x]|_{x=x_0,u=u_0} \in \mathbb{R}^{n\times n}$ is the system matrix, and $G[x_0, u_0] = [\partial(f(x) + g(x)u(x))/\partial u]|_{x=x_0,u=u_0} \in \mathbb{R}^{n\times m}$ is the control effectiveness matrix. However, a recursive least square method is demanded to search for suitable gain matrices $F[x_0, u_0]$ and $G[x_0, u_0]$ to construct the incremental dynamics [120]–[122]. This required online identification of $F[x_0, u_0]$ and $G[x_0, u_0]$ introduces additional computational burden.*

### 6.1.2 Problem Transformation to Optimal Incremental Control

To address the unknown TDE error in the incremental dynamics (6.4), here we attempt to investigate the original robust stabilization problem shown as Problem 6.1 from an optimal control perspective, whereby the TDE error could be reflected in the performance index and further be attenuated during the optimization process. This departs from existing TDE related works [108], [109], [120]–[124] that directly ignore the influence of the TDE error on the controller performance. Moreover, the effort to solve Problem 6.1 under an optimization framework enables us to take the desired performance indexes regarding state deviations and control energy expenditures into consideration. These considered performance indexes endow the resulting TDE based model-free control strategy with guaranteed optimality.

The TDE error $\xi$ in (6.4) is unknown. Thus, the available incremental dynamics to design a controller to solve Problem 6.1 follows

$$\Delta\dot{x} = \bar{g}\Delta u. \tag{6.9}$$

To attenuate the TDE error $\xi$ that is overlooked in (6.9), as well as to optimize the performance of states and control inputs, we consider the cost function of (6.9) as

$$V(x(t)) = \int_t^\infty r(x(\tau), \Delta u(\tau))\, d\tau, \tag{6.10}$$

where $r(x, \Delta u) = x^\top Q x + \mathcal{W}(u_0 + \Delta u) + \bar{\xi}_o^2 : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^+$. The common quadratic positive definite term $x^\top Q x$ reflects users' preference for the controller performance concerning state deviations, where $Q \in \mathbb{R}^{n\times n}$ is a positive definite matrix. The nonquadratic positive definite control penalty function $\mathcal{W}(u_0 + \Delta u)$, which relates to the measured $u_0$ and to be designed $\Delta u$, is introduced to enforce the control limit on $u(x)$ based on the bounded tanh function. The explicit form of this part follows [90]

$$\mathcal{W}(u_0 + \Delta u) = 2\sum_{j=1}^{m} \int_0^{u_{0_j} + \Delta u_j} \beta \tanh^{-1}(\vartheta_j/\beta)\, d\vartheta_j. \tag{6.11}$$

where $\vartheta_j \in \mathbb{R}^m$. Originally, we could incorporate the quadratic TDE error bound $\bar{\xi}^2$ into $r(x, \Delta u)$ to attenuate the TDE error $\xi$ during the optimization process. However, according to (6.8) of Lemma 6.1, the explicit value of $\bar{\xi}$ is unknown. Thus, we seek for a bounded $\bar{\xi}_o^2$, where $\bar{\xi}_o = \bar{c}\,\|\Delta u\|$ and $\bar{c} \in \mathbb{R}^+$ is chosen as illustrated in Theorem 6.1, to replace $\bar{\xi}^2$ to accomplish the same goal. It is worth noting that the designed utility function $r(x, \Delta u)$ here enables us to perform the optimization of incremental control inputs.

**Remark 6.6.** *Note that there exist other options to address the TDE error $\xi$. For example, by treating the unknown TDE error $\xi$ in* (6.4) *as a kind of disturbance, we can introduce the widely used disturbance-observer based methods [125] or sliding mode control methods [126] to compensate the TDE error $\xi$. Comparing to these add-on methods, our strategy enjoys computational simplicity.*

The above settings allow us to formulate an optimal incremental control problem presented as Problem 6.2, whose equivalence to Problem 6.1 will be later proved in Theorem 6.1.

**Problem 6.2.** *Given Assumption 6.1 and Lemma 6.1, consider the incremental dynamics* (6.9)*, find an incremental control strategy $\Delta u$ to minimize the cost function defined as* (6.10)*.*

Before proceeding to formally solve Problem 6.2, by following [90, Definition 1] where admissible controls are defined based on (6.1), here we define the set of incremental control inputs that are considered admissible for Problem 6.2 dealt with in this section. The admissible incremental control given in Definition 6.1 facilitates the following derivation of the closed-form optimal incremental control strategy.

**Definition 6.1** (Admissible incremental control)**.** *An incremental control $\Delta\mu(x)$ is defined to be admissible with respect to* (6.10) *on $\Omega \subseteq \mathbb{R}^n$, denoted by $\Delta\mu(x) \in \Psi(\Omega)$, if $\Delta\mu(x)$ is continuous on $\Omega$, $\Delta\mu(0) = 0$, $\Delta u(x) = \Delta\mu(x)$ stabilizes* (6.9) *on $\Omega$, and $V(x)$ is finite $\forall x \in \Omega$.*

For any admissible incremental control policies $\Delta u \in \Psi(\Omega)$, using Leibniz's rule [127] to differentiate $V$ in (6.10) yields the following relation

$$0 = r(x, \Delta u) + \nabla V^T \dot{x} = r(x, \Delta u) + \nabla V^T (\Delta\dot{x} + \dot{x}_0) = r(x, \Delta u) + \nabla V^T (\bar{g}\Delta u + \dot{x}_0). \tag{6.12}$$

Define the Hamiltonian function as

$$H(x, \Delta u, \nabla V) = r(x, \Delta u) + \nabla V^T(\bar{g}\Delta u + \dot{x}_0). \tag{6.13}$$

Let $V^*(x)$ be the optimal cost function defined as

$$V^*(x) = \min_{\Delta u \in \Psi(\Omega)} \int_t^\infty r(x(\tau), \Delta u(\tau))\, d\tau. \tag{6.14}$$

Combining with (6.13), $V^*(x)$ satisfies the HJB equation

$$0 = \min_{\Delta u \in \Psi(\Omega)} [H(x, \Delta u, \nabla V^*)]. \tag{6.15}$$

Assume that the minimum on the right side of (6.15) exists and is unique [19]. By using the stationary optimality condition, i.e., $\partial H(x, \Delta u, \nabla V^*)/\partial \Delta u = 0$, we get the closed-form optimal incremental control strategy as

$$\Delta u^* = -\beta \tanh\left(\frac{1}{2\beta}\bar{g}^\top \nabla V^*\right) - u_0. \tag{6.16}$$

Then, we could construct the corresponding optimal control strategy as

$$u^* = u_0 + \Delta u^* = -\beta \tanh\left(\frac{1}{2\beta}\bar{g}^\top \nabla V^*\right). \tag{6.17}$$

Departing from traditional ADP related works [19], [20] where the total optimal control input $u^*$ is directly designed, here we first get the theoretically derived incremental optimal control strategy $\Delta u^*$ in (6.16), and then construct $u^*$ based on the measured $u_0$ and the designed $\Delta u^*$. This difference lies in that Problem 6.2 is formulated based on the incremental dynamics (6.9) that relates to incremental states and control inputs.

**Remark 6.7.** *Alternatively, we could replace the utilized $\mathcal{W}(u_0 + \Delta u)$ with $\mathcal{W}(\Delta u) = 2\sum_{j=1}^m \int_0^{\Delta u_j} \alpha \tanh^{-1}(\vartheta_j/\alpha)\, d\vartheta_j$. This enforces the constraint satisfaction of the incremental control inputs, which is denoted as $-\alpha \leq \Delta u_j \leq \alpha$, $\alpha \in \mathbb{R}^+$, $j = 1, \cdots, m$. By following the aforementioned derivation processes* (6.14)-(6.17)*, the corresponding optimal incremental control follows $\Delta u^* = -\alpha \tanh\left(\frac{1}{2\alpha}\bar{g}^\top \nabla V^*\right)$. Then, the resulting optimal control is $u^* = u_0 + \Delta u^*$. However, in this case, the control limit on $u(x)$ cannot be addressed. Given that input saturation is common in real life and violations of it might lead to serious consequences, we prefer to incorporate* (6.11) *into $r(x, \Delta u)$ to enforce the control limit on $u(x)$.*

To get $\Delta u^*$ (6.16) and $u^*$ (6.17), $\nabla V^*$ remains to be determined. We defer the explicit method to acquire $\nabla V^*$ in Section 6.2, and focus now on the equivalence proof to show that after solving Problem 6.2, the resulting $u^*$ (6.17) constructed from the designed $\Delta u^*$ (6.16) is the robust stabilization solution to Problem 6.1.

**Theorem 6.1.** *Given Assumption 6.1 and Lemma 6.1, consider the system described by* (6.1)*, if there exists a scalar $\bar{c} \in \mathbb{R}^+$ such that*

$$\bar{\xi} < \bar{c}\,\|\Delta u\|, \tag{6.18}$$

*the system* (6.1) *is robustly stabilized by the optimal control strategy* (6.17) *with the optimal incremental control strategy* (6.16)*.*

*Proof.* Given that $V^*(x=0)=0$, and $V^*>0$ for $\forall x\neq 0$, $V^*$ defined in (6.14) could serve as a Lyapunov function candidate for the stability proof. Taking time derivative of $V^*$ along the incremental dynamics (6.4), which is an equivalent of the original dynamics (6.1), we get

$$\dot{V}^* = \nabla V^{*\top}(\Delta\dot{x}+\dot{x}_0) = \nabla V^{*\top}(\bar{g}\Delta u^* + \bar{g}\xi + \dot{x}_0) = \nabla V^{*\top}(\bar{g}\Delta u^* + \dot{x}_0) + \nabla V^{*\top}\bar{g}\xi. \quad (6.19)$$

According to (6.15) and (6.16), the following equations hold:

$$\nabla V^{*\top}(\bar{g}\Delta u^* + \dot{x}_0) = -x^\top Qx - \mathcal{W}(u_0+\Delta u^*) - \bar{\xi}_o^2,\ \nabla V^{*\top}\bar{g} = -2\beta\tanh^{-1}\left(\frac{u_0+\Delta u^*}{\beta}\right). \quad (6.20)$$

Substituting (6.20) into (6.19) reads

$$\dot{V}^* = -x^\top Qx - \mathcal{W}(u_0+\Delta u^*) - \bar{\xi}_o^2 - 2\beta\tanh^{-1}\left(\frac{u_0+\Delta u^*}{\beta}\right)\xi. \quad (6.21)$$

As for $\mathcal{W}(u_0+\Delta u^*)$ in (6.21), based on the explicit form in (6.11) and by setting $\varsigma_j = \tanh^{-1}(\vartheta_j/\beta)$, it follows

$$\begin{aligned}\mathcal{W}(u_0+\Delta u^*) &= 2\beta\sum_{j=1}^m\int_0^{u_{0_j}+\Delta u_j^*}\tanh^{-1}(\vartheta_j/\beta)\,d\vartheta_j \\ &= 2\beta^2\sum_{j=1}^m\int_0^{\tanh^{-1}\left(\frac{u_{0_j}+\Delta u_j^*}{\beta}\right)}\varsigma_j(1-\tanh^2(\varsigma_j))\,d\varsigma_j \\ &= \beta^2\sum_{j=1}^m\left(\tanh^{-1}\left(\frac{u_{0_j}+\Delta u_j^*}{\beta}\right)\right)^2 - \epsilon_u, \end{aligned} \quad (6.22)$$

where $\epsilon_u = 2\beta^2\sum_{j=1}^m\int_0^{\tanh^{-1}\left(\frac{u_{0_j}+\Delta u_j^*}{\beta}\right)}\varsigma_j\tanh^2(\varsigma_j)\,d\varsigma_j$. Based on the integral mean-value theorem, there exists a series of $\theta_j\in[0,\tanh^{-1}\left(\frac{u_{0_j}+\Delta u_j^*}{\beta}\right)]$, $j=1,\cdots,m$, such that

$$\epsilon_u = 2\beta^2\sum_{j=1}^m\tanh^{-1}\left(\frac{u_{0_j}+\Delta u_j^*}{\beta}\right)\theta_j\tanh^2(\theta_j). \quad (6.23)$$

Based on (6.20) and the fact $0\le\tanh^2(\theta_j)\le 1$, it follows

$$\epsilon_u \le 2\beta^2\sum_{j=1}^m\left(\frac{u_{0_j}+\Delta u_j^*}{\beta}\right)\theta_j \le 2\beta^2\sum_{j=1}^m\left(\tanh^{-1}\left(\frac{u_{0_j}+\Delta u_j^*}{\beta}\right)\right)^2 = \frac{1}{2}\nabla V^{*\top}\bar{g}\bar{g}^\top\nabla V^*. \quad (6.24)$$

The definition of admissible incremental control in Definition 6.1 implies that $V^*$ is finite. Additionally, there exists $b_{\nabla V^*}\in\mathbb{R}^+$ such that $\|\nabla V^*\|\le b_{\nabla V^*}$. Thus, we could rewrite (6.24) as

$$\epsilon_u \le b_{\epsilon_u} = \frac{1}{2}\|\bar{g}\|^2 b_{\nabla V^*}^2. \quad (6.25)$$

Then, substituting (6.22), (6.25) into (6.21) yields

$$\dot{V}^* \le -x^\top Qx - (\bar{\xi}_o^2 - \|\xi\|^2) - [\beta\tanh^{-1}\left(\frac{u_0+\Delta u^*}{\beta}\right)+\xi]^2 + b_{\epsilon_u}. \quad (6.26)$$

By choosing $\bar{\xi}_o = \bar{c}\,\|\Delta u\|$, and $\bar{c}$ is chosen to satisfy $\bar{c}\,\|\Delta u\| > \bar{\xi}$, where $\bar{\xi}$ is defined in (6.8), the following inequality holds

$$\dot{V}^* \leq -x^\top Qx + b_{\epsilon_u}. \tag{6.27}$$

Thus, $\dot{V}^* < 0$ holds if $-\lambda_{\min}(Q)\,\|x\|^2 + b_{\epsilon_u} < 0$. Finally, it concludes that states converge to the residual set

$$\Omega_x = \{x|\,\|x\| \leq \sqrt{b_{\epsilon_u}/\lambda_{\min}(Q)}\}. \tag{6.28}$$

The aforementioned proof means that based on the optimal cost function (6.14), the derived optimal incremental control policy (6.16) of the system (6.9) robustly stabilizes the system (6.4). Given the equivalence between (6.1) and (6.4) clarified in Section 6.1.1, thus the optimal control input (6.17), which is constructed from the designed (6.16), robustly stabilizes the system (6.1). □

We have proved in Theorem 6.1 that the optimal incremental control problem clarified in Problem 6.2 is equivalent to the robust stabilization problem shown as Problem 6.1. Thus, to stabilize the highly uncertain dynamics (6.1) operating in a disturbed environment, the remaining part devotes to solving Problem 6.2.

## 6.2 Approximate Optimal Solution

To solve Problem 6.2, this section seeks for the approximate solution to the value function of the HJB equation (6.15) that is hard to solve directly. Departing from common ADP related works [19], [20] using an actor-critic learning structure, we use a single critic learning structure here, which decreases the computational burden and simplifies the theoretical analysis. The associated critic NN learning process adopts the off-policy weight update law developed in Chapter 5.

### 6.2.1 Approximated Value Function

Based on the Weierstrass high-order approximation theorem [46], for $x \in \Omega$ with $\Omega \subset \mathbb{R}^n$ being a compact set, the optimal value function is approximated as [19]

$$V^*(x) = W^{*\top}\Phi(x) + \epsilon(x), \tag{6.29}$$

where $W^* \in \mathbb{R}^N$ is a weighting matrix, $\Phi(x) : \mathbb{R}^n \to \mathbb{R}^N$ represents the activation function, and $\epsilon(x) \in \mathbb{R}$ denotes the approximation error. The partial derivative of $V^*(x)$ follows

$$\nabla V^*(x) = \nabla\Phi^\top(x)W^* + \nabla\epsilon(x), \tag{6.30}$$

where $\nabla\Phi(x) \in \mathbb{R}^{N\times n}$, $\nabla\epsilon(x) \in \mathbb{R}^n$. As $N \to \infty$, both $\epsilon(x)$ and $\nabla\epsilon(x)$ converge to zero uniformly. Without loss of generality, the following assumption is given, which is common in ADP related works.

**Assumption 6.2.** *[19] There exist constants $b_\epsilon, b_{\epsilon x}, b_\Phi, b_{\Phi x} \in \mathbb{R}^+$ such that $\|\epsilon(x)\| \leq b_\epsilon$, $\|\nabla\epsilon(x)\| \leq b_{\epsilon x}$, $\|\Phi(x)\| \leq b_\Phi$, and $\|\nabla\Phi(x)\| \leq b_{\Phi x}$.*

Considering a fixed incremental control input $\Delta u$, inserting (6.30) into (6.15) yields

$$W^{*\top}\nabla\Phi(\bar{g}\Delta u + \dot{x}_0) + r(x, \Delta u) = \epsilon_h, \tag{6.31}$$

where the residual error follows $\epsilon_h = -\nabla\epsilon^\top(\bar{g}\Delta u + \dot{x}_0) \in \mathbb{R}$. Assume that there exists $b_{\epsilon_h} \in \mathbb{R}^+$ such that $\|\epsilon_h\| \leq b_{\epsilon_h}$. Focusing on the NN parameterized (6.31), we rewrite it into the following LIP form

$$\Theta = -W^{*\top}Y + \epsilon_h, \tag{6.32}$$

where $\Theta = r(x, \Delta u) \in \mathbb{R}$, and $Y = \nabla\Phi(\bar{g}\Delta u + \dot{x}_0) \in \mathbb{R}^N$. Given that $\Theta$ and $Y$ could be obtained from real-time data, this formulated LIP form enables the learning of $W^*$ to be equivalent to a parameter identification problem of an LIP system from the perspective of adaptive control. The above applied transformation allows us to directly use our developed off-policy weight update law in Chapter 5 to solve Problem 6.2.

### 6.2.2 NN Weight Update Law

Following our previous results in Section 5.2.2, the critic NN weight updates as

$$\dot{\hat{W}} = -\Gamma k_c Y\tilde{\Theta} - \sum_{l=1}^{P} \Gamma k_e Y_l \tilde{\Theta}_l, \tag{6.33}$$

to get the approximate solution to the HJB function (6.15).

The guaranteed weight convergence of $\hat{W}$ to $W^*$ permits us to directly use the estimated critic NN weight $\hat{W}$ to construct the approximate optimal incremental control strategy. Therefore, based on the optimal incremental control strategy in (6.16), the approximate optimal incremental control strategy follows

$$\Delta\hat{u} = -\beta\tanh\left(\frac{1}{2\beta}\bar{g}^\top\nabla\Phi^\top\hat{W}\right) - u_0. \tag{6.34}$$

Accordingly, the approximate optimal control strategy applied at the plant (6.1) follows

$$\hat{u} = u_0 + \Delta\hat{u} = -\beta\tanh\left(\frac{1}{2\beta}\bar{g}^\top\nabla\Phi^\top\hat{W}\right). \tag{6.35}$$

**Remark 6.8.** *From a practical perspective, our designed model-free approximate optimal incremental control strategy* (6.34) *only requires one manually tuned constant matrix* $\bar{g}$. *This feature of IADP decreases the required parameter tuning efforts comparing to existing identification based methods to fulfill model-free control strategies [24]–[28], [110], where multiple hyperparameters or gains need to be tuned.*

Based on the off-policy weight update law (6.33), and the approximate optimal incremental control strategy (6.34) mentioned above, we provide the main conclusions as follows.

**Theorem 6.2.** *Consider the incremental dynamics* (6.9)*, the off-policy weight update law of the critic NN in* (6.33)*, and the approximate optimal incremental control policy* (6.34)*. Given Assumption 5.5 and Assumptions 6.1-6.2, for a sufficiently large* $N$*, the approximate optimal incremental control policy* (6.34) *stabilizes the incremental dynamics* (6.9)*, and the critic NN weight learning error* $\tilde{W}$ *is uniformly ultimately bounded.*

*Proof.* Consider the following candidate Lyapunov function

$$J = V^*(x) + \frac{1}{2}\tilde{W}^\top \Gamma^{-1} \tilde{W}. \tag{6.36}$$

By denoting $\dot{L}_V = \dot{V}^*(x)$ and $\dot{L}_W = \tilde{W}^\top \Gamma^{-1} \dot{\hat{W}}$, the time derivative of (6.36) reads

$$\dot{J} = \dot{L}_V + \dot{L}_W. \tag{6.37}$$

The first term $\dot{L}_V$ follows

$$\dot{L}_V = \nabla V^{*\top}(\bar{g}\Delta\hat{u} + \bar{g}\xi + \dot{x}_0) = \nabla V^{*\top}(\bar{g}\Delta u^* + \dot{x}_0) + \nabla V^{*\top}\bar{g}\xi + \nabla V^{*\top}\bar{g}(\Delta\hat{u} - \Delta u^*). \tag{6.38}$$

Then, substituting (6.20) into (6.38) gets

$$\dot{L}_V = -x^\top Qx - \mathcal{W}(u_0 + \Delta u^*) - \bar{\xi}_o^2 - 2\beta \tanh^{-1}\left(\frac{u_0 + \Delta u^*}{\beta}\right)\xi - 2\beta \tanh^{-1}\left(\frac{u_0 + \Delta u^*}{\beta}\right)(\Delta\hat{u} - \Delta u^*). \tag{6.39}$$

According to (6.22)-(6.24), (6.39) follows

$$\begin{aligned} \dot{L}_V \leq & - x^\top Qx - (\bar{\xi}_o^2 - \|\xi\|^2) - [\beta \tanh^{-1}\left(\frac{u_0 + \Delta u^*}{\beta}\right) + \xi]^2 + \frac{1}{2}\nabla V^{*\top}\bar{g}\bar{g}^\top \nabla V^* \\ & - 2\beta \tanh^{-1}\left(\frac{u_0 + \Delta u^*}{\beta}\right)(\Delta\hat{u} - \Delta u^*). \end{aligned} \tag{6.40}$$

The term $-2\beta \tanh^{-1}\left(\frac{u_0+\Delta u^*}{\beta}\right)(\Delta\hat{u} - \Delta u^*)$ in (6.40) follows

$$-2\beta \tanh^{-1}\left(\frac{u_0 + \Delta u^*}{\beta}\right)(\Delta\hat{u} - \Delta u^*) \leq \beta^2 \left\|\tanh^{-1}\left(\frac{u_0 + \Delta u^*}{\beta}\right)\right\|^2 + \|\Delta\hat{u} - \Delta u^*\|^2. \tag{6.41}$$

By using (6.16), (6.30), and the mean-value theorem, the optimal incremental control is rewritten as

$$\Delta u^* = -\beta \tanh\left(\frac{1}{2\beta}\bar{g}^\top \nabla\Phi^\top W^*\right) - \epsilon_{\Delta u^*} - u_0, \tag{6.42}$$

where $\epsilon_{\Delta u^*} = \frac{1}{2}(\underline{1} - \tanh^2(\eta))\bar{g}^\top \nabla\epsilon$, and $\eta \in \mathbb{R}^m$ is chosen between $\frac{1}{2\beta}\bar{g}^\top \nabla\Phi^\top W^*$ and $\frac{1}{2\beta}\bar{g}^\top \nabla V^*$, $\underline{1} = [1, \cdots, 1]^\top \in \mathbb{R}^m$. According to $\|\nabla\epsilon\| \leq b_{\epsilon x}$ in Assumption 6.2, $\|\epsilon_{\Delta u^*}\| \leq \frac{1}{2}\|\bar{g}\| b_{\epsilon x}$ holds. Then, combining (6.34) with (6.42), we get

$$\Delta\hat{u} - \Delta u^* = \beta(\tanh\left(\frac{1}{2\beta}\bar{g}^\top \nabla\Phi^\top W^*\right) - \tanh\left(\frac{1}{2\beta}\bar{g}^\top \nabla\Phi^\top \hat{W}\right) + \epsilon_{\Delta u^*}. \tag{6.43}$$

Denoting $\mathscr{G}^* = \frac{1}{2\beta}\bar{g}^\top \nabla\Phi^\top W^*$, and $\hat{\mathscr{G}} = \frac{1}{2\beta}\bar{g}^\top \nabla\Phi^\top \hat{W}$, where $\hat{\mathscr{G}} = [\hat{\mathscr{G}}_1, \cdots, \hat{\mathscr{G}}_m] \in \mathbb{R}^m$ with $\hat{\mathscr{G}}_j \in \mathbb{R}, j = 1, \cdots, m$. Based on (6.16) and (6.34), the Taylor series of $\tanh(\mathscr{G}^*)$ follows

$$\begin{aligned} \tanh(\mathscr{G}^*) &= \tanh\left(\hat{\mathscr{G}}\right) + \frac{\partial \tanh\left(\hat{\mathscr{G}}\right)}{\partial \hat{\mathscr{G}}}(\mathscr{G}^* - \hat{\mathscr{G}}) + O((\mathscr{G}^* - \hat{\mathscr{G}})^2) \\ &= \tanh\left(\hat{\mathscr{G}}\right) - \frac{1}{2\beta}(I_{m\times m} - \mathscr{D}(\hat{\mathscr{G}}))\bar{g}^\top \nabla\Phi^\top \tilde{W} + O((\mathscr{G}^* - \hat{\mathscr{G}})^2), \end{aligned} \tag{6.44}$$

where $\mathscr{D}(\hat{\mathscr{G}}) = \mathrm{diag}(\tanh^2(\hat{\mathscr{G}}_1), \cdots, \tanh^2(\hat{\mathscr{G}}_m))$, and $O((\mathscr{G}^* - \hat{\mathscr{G}})^2)$ is a higher order term of the Taylor series. By following [100, Lemma 1], this higher order term is bounded as

$$\left\| O((\mathscr{G}^* - \hat{\mathscr{G}})^2) \right\| \leq 2\sqrt{m} + \frac{1}{\beta} \|\bar{g}\| \, b_{\Phi x} \left\| \tilde{W} \right\|. \tag{6.45}$$

Based on (6.44), we rewrite (6.43) as

$$\Delta\hat{u} - \Delta u^* = \beta(\tanh(\mathscr{G}^*) - \tanh\left(\hat{\mathscr{G}}\right)) + \epsilon_{\Delta u^*} = -\frac{1}{2}(I_{m\times m} - \mathscr{D}(\hat{\mathscr{G}}))\bar{g}\nabla\Phi^\top \tilde{W} + \beta O((\mathscr{G}^* - \hat{\mathscr{G}})^2) + \epsilon_{\Delta u^*}. \tag{6.46}$$

According to [100], $\left\| I_{m\times m} - \mathscr{D}(\hat{\mathscr{G}}) \right\| \leq 2$ holds. Then, combining (6.45) with (6.46), $\|\Delta\hat{u} - \Delta u^*\|^2$ in (6.41) follows

$$\begin{aligned} \|\Delta\hat{u} - \Delta u^*\|^2 &\leq 3\beta^2 \left\| O((\mathscr{G}^* - \hat{\mathscr{G}})^2) \right\|^2 + 3 \|\epsilon_{\Delta u^*}\|^2 + 3 \left\| -\frac{1}{2}(I_{m\times m} - \mathscr{D}(\hat{\mathscr{G}}))\bar{g}^\top \nabla\Phi^\top \tilde{W} \right\|^2 \\ &\leq 6 \|\bar{g}\|^2 b_{\Phi x}^2 \left\| \tilde{W} \right\|^2 + 12m\beta^2 + \frac{3}{4} \|\bar{g}\|^2 b_{\epsilon x}^2 + 12\beta\sqrt{m} \|\bar{g}\| \, b_{\Phi x} \left\| \tilde{W} \right\|. \end{aligned} \tag{6.47}$$

Based on (6.20), (6.30), Assumption 6.2, and the fact that $\|W^*\| \leq b_{W^*}$, $\left\| \tanh^{-1}((u_0 + \Delta u^*)/\beta) \right\|^2$ in (6.41) follows

$$\begin{aligned} \left\| \tanh^{-1}\left(\frac{u_0 + \Delta u^*}{\beta}\right) \right\|^2 &= \left\| \frac{1}{4\beta^2} \nabla V^{*\top} \bar{g}\bar{g}^\top \nabla V^* \right\| \\ &\leq \frac{1}{4\beta^2} \|\bar{g}\|^2 b_{\Phi x}^2 b_{W^*}^2 + \frac{1}{4\beta^2} b_{\epsilon x}^2 \|\bar{g}\|^2 + \frac{1}{2\beta^2} \|\bar{g}\|^2 b_{\Phi x} b_{\epsilon x} b_{W^*}. \end{aligned} \tag{6.48}$$

Using (6.47) and (6.48), (6.41) reads

$$\begin{aligned} -2\beta \tanh^{-1}\left(\frac{u_0 + \Delta u^*}{\beta}\right)(\Delta\hat{u} - \Delta u^*) \leq & \frac{1}{4} \|\bar{g}\|^2 b_{\Phi x}^2 b_{W^*}^2 + \frac{1}{4} b_{\epsilon x}^2 \|\bar{g}\|^2 + \frac{1}{2} \|\bar{g}\|^2 b_{\Phi x} b_{\epsilon x} b_{W^*} + 12m\beta^2 \\ & + 6 \|\bar{g}\|^2 b_{\Phi x}^2 \left\| \tilde{W} \right\|^2 + \frac{3}{4} \|\bar{g}\|^2 b_{\epsilon x}^2 + 12\beta\sqrt{m} \|\bar{g}\| \, b_{\Phi x} \left\| \tilde{W} \right\|. \end{aligned} \tag{6.49}$$

Substituting (6.49) into (6.40), finally the first term $\dot{L}_V$ follows

$$\begin{aligned} \dot{L}_V \leq & - x^\top Q x - (\bar{\xi}_o^2 - \xi^\top \xi) - [\beta \tanh^{-1}\left(\frac{u_0 + \Delta u^*}{\beta}\right) + \xi]^2 + \frac{3}{4} \|\bar{g}\|^2 b_{\Phi x}^2 b_{W^*}^2 + \frac{3}{4} b_{\epsilon x}^2 \|\bar{g}\|^2 \\ & + \frac{3}{2} \|\bar{g}\|^2 b_{\Phi x} b_{\epsilon x} b_{W^*} + 6 \|\bar{g}\|^2 b_{\Phi x}^2 \left\| \tilde{W} \right\|^2 + 12m\beta^2 + \frac{3}{4} \|\bar{g}\|^2 b_{\epsilon x}^2 + 12\beta\sqrt{m} \|\bar{g}\| \, b_{\Phi x} \left\| \tilde{W} \right\|. \end{aligned} \tag{6.50}$$

As for the second term $\dot{L}_W$, based on (6.33) and the weight convergence proof already illustrated in Theorem 5.2 , it follows

$$\dot{L}_W \leq -\tilde{W}^\top B \tilde{W} + \tilde{W}^\top \epsilon_{\tilde{W}}. \tag{6.51}$$

Finally, as for $\dot{J}$, substituting (6.50) and (6.51) into (6.37), we get

$$\dot{J} \leq -\mathcal{A} - \mathcal{B} \left\| \tilde{W} \right\|^2 + \mathcal{C} \left\| \tilde{W} \right\| + \mathcal{D}, \tag{6.52}$$

where $\mathcal{A} = x^\top Q x + (\bar{\xi}_o^2 - \xi^\top \xi) + [\beta \tanh^{-1}\left(\frac{u_0 + \Delta u^*}{\beta}\right) + \xi]^2$, $\mathcal{B} = \lambda_{\min}(B) - 6 \left\|\bar{g}\right\|^2 b_{\Phi x}^2$, $\mathcal{C} = 12\beta\sqrt{m}\left\|\bar{g}\right\| b_{\Phi x} + \bar{\epsilon}_{\tilde{W}}$, and $\mathcal{D} = \frac{3}{4}\left\|\bar{g}\right\|^2 b_{\Phi x}^2 b_{W^*}^2 + \frac{3}{2} b_{\epsilon x}^2 \left\|\bar{g}\right\|^2 + \frac{3}{2}\left\|\bar{g}\right\|^2 b_{\Phi x} b_{\epsilon x} b_{W^*} + 12 m \beta^2$. Let the parameters be chosen such that $\mathcal{B} > 0$. Since $\mathcal{A}$ is positive definite, the above Lyapunov derivative (6.52) is negative if $\left\|\tilde{W}\right\| > \frac{\mathcal{C}}{2\mathcal{B}} + \sqrt{\frac{\mathcal{C}^2}{4\mathcal{B}^2} + \frac{\mathcal{D}}{\mathcal{B}}}$. Thus, the critic weight learning error converges to the residual set $\tilde{\Omega}_{\tilde{W}} = \left\{\tilde{W} | \left\|\tilde{W}\right\| \leq \frac{\mathcal{C}}{2\mathcal{B}} + \sqrt{\frac{\mathcal{C}^2}{4\mathcal{B}^2} + \frac{\mathcal{D}}{\mathcal{B}}}\right\}$. □

## 6.3 Safety Filter Implementation

Under a satisfying framework [128], this section introduces a safety filter to correct the learned approximate optimal control policy (6.35) via a minimally invasive way to ensure safe operation. The safety filter is implemented as a CBF based QP formulation:

$$\begin{aligned} u_s &= \arg\min_{u_s} \left\|u_s - \hat{u}\right\| \\ \text{s. t.} \quad & \ddot{h}_j + \alpha_{1_j}\dot{h}_j + \alpha_{2_j} h_j \geq 0, \ j = 1, 2, \cdots \end{aligned} \tag{6.53}$$

where $u_s \in \mathbb{R}^n$ is the corrected safe control input; $h_j$ is the $j$-th HO-CBF characterizing the $j$-th unsafe region, which is prior-given or learned via the method developed in Chapter 4; $\alpha_{1_j}$, $\alpha_{2_j} \in \mathbb{R}^+$ are chosen using the method developed in Section 4.5.2 to guarantee that the utilized $h_j$ is a valid HO-CBF. The barrier certified approximate optimal control policy from (6.53) are used for future data collection to support the value function learning process illustrated in Section 6.2.

The presented QP (6.53) implies the potential conflict between safety and performance. For practical applications, safety should be prioritized over performance. Therefore, the solution is to safely achieve a performance that is as close as possible to the desired performance. This is achieved by (6.53), wherein the relaxation of strict optimality allows us to introduce safety considerations into our method. This add-on safety filter method enjoys flexibility towards multiple tasks and environments, although the promising theoretical optimality guarantee is lost. This is satisfying for practical applications.

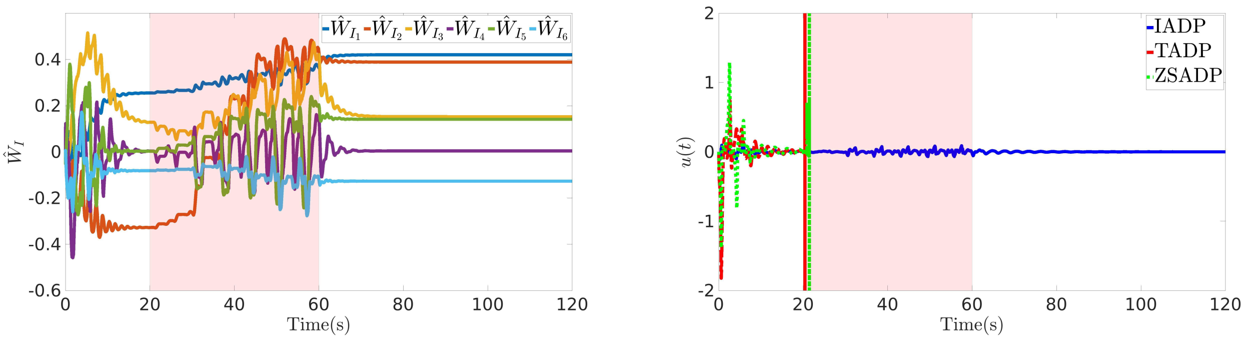
## 6.4 Numerical Simulation

This section conducts comparative numerical simulations to validate the effectiveness and superiority of our proposed IADP. Besides, the influence of different sampling rates on IADP's performance is investigated.

Here, we choose the widely investigated pendulum in ADP related works [129], [130] as a benchmark. The dynamics of the pendulum follows

$$\begin{cases} \frac{d\theta}{dt} = \vartheta + d \\ J\frac{d\vartheta}{dt} = u - Mgl\sin\theta - f_d\frac{d\theta}{dt}, \end{cases} \tag{6.54}$$

where $\theta, \vartheta \in \mathbb{R}$ denote the angle and the angular velocity of the pendulum, respectively. $M = 1/3$ kg and $l = 3/2$ m are the mass and length of the pendulum, respectively. Let $g = 9.8\ \mathrm{m/s^2}$ be the gravity, $J = 4/3Ml^2\ \mathrm{kg{\cdot}m^2}$ be the rotary inertia, and $f_d = 0.2$ be the frictional factor. Here $d$ represents an external disturbance.



(a) The trajectory of $\hat{W}_I$ of IADP.

(b) The trajectory of $\hat{u}$.

Figure 6.1: The estimated weight trajectory of IADP and the control trajectories of IADP, ZSADP, and TADP.

### 6.4.1 Validation in Complex Environment

To highlight the enhanced robustness of our proposed IADP over the zero-sum game based ADP (ZSADP) [131] and the transformed optimal control based ADP (TADP) [132], this subsection conducts numerical simulations under a complex simulation environment. The details are as follows: during the time from 20 $s$ to 60 $s$, the added non-vanishing disturbance $d(t)$ is a square wave with amplitude 0.5 and period 1$s$; the incorporated measurement noise is set as a white Gaussian noise with 10 dBW; Besides, the pendulum (6.54) is reset as

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \underbrace{\begin{bmatrix}-x_2\\ 4.9\sin x_1 - 0.2x_2\end{bmatrix}}_{f(x)} + \underbrace{\begin{bmatrix}0\\ -0.25\end{bmatrix}}_{g(x)} u + \underbrace{\begin{bmatrix}1\\ -0.2\end{bmatrix}}_{k(x)} d_1(x), \tag{6.55}$$

at $t = 20$ $s$ to model a significant physical change. The state-dependent disturbance is chosen as $d_1 = \omega_1\theta\sin(\omega_2\vartheta)$, where $\omega_1$ and $\omega_2$ are randomly generated within the scope $[-\sqrt{2}/2, \sqrt{2}/2]$ and $[-2, 2]$, respectively. The sampling rate is chosen as 1000 Hz. The detailed simulation settings for IADP, ZSADP, and TADP are as follows.

For IADP, we choose $\bar{g} = [0, 0.1]^\top$. Its cost function is considered as

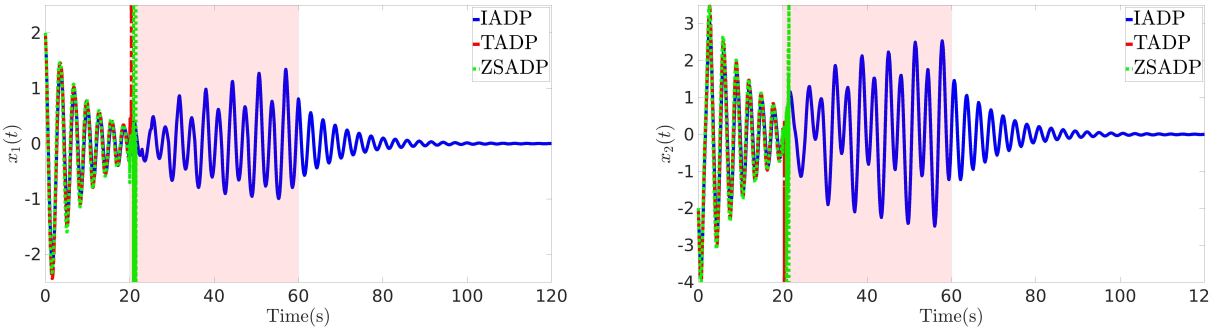
$$V_I = \int_t^\infty x^\top Qx + \mathcal{W}(u_0 + \Delta u) + \bar{\xi}_o^2 \, d\tau, \tag{6.56}$$

where $Q = I_{2\times 2}$, $\mathcal{W}(u_0+\Delta u) = 2\beta(u_0+\Delta u)\tanh^{-1}((u_0+\Delta u)/\beta)+\beta^2\log(1-(u_0+\Delta u)^2/\beta^2)$, and $\bar{\xi}_o = 2\|\Delta u\|$. The approximate optimal incremental control $\Delta\hat{u}$ and the approximate optimal control $\hat{u}$ follow (6.34) and (6.35), respectively. IADP requires neither explicit model nor environmental information except for a predefined constant matrix $\bar{g}$.

For ZSADP, following the method in [131], the cost function is chosen as

$$V_Z = \int_t^\infty x^\top Qx + \mathcal{W}(u_Z) - \gamma d_Z^\top d_Z \, d\tau, \tag{6.57}$$

where $\mathcal{W}(u_Z) = 2\beta u_Z\tanh^{-1}(u_Z/\beta) + \beta^2\log(1 - u_Z^2/\beta^2)$, $\gamma = 1$. For this case, the approximate optimal control policy follows $\hat{u}_Z = -\beta\tanh\!\left(\frac{1}{2\beta}g^\top\nabla\Phi^\top\hat{W}_Z\right)$, and the approximate worst-case disturbance policy is $\hat{d}_Z = \frac{1}{2\gamma^2}k^\top\nabla\Phi^\top\hat{W}_Z$. Here $\hat{u}_Z$ and $\hat{d}_Z$ depend on the concert $g(x)$ and $k(x)$ in (6.55), respectively.



(a) The trajectories of $x_1(t)$.

(b) The trajectories of $x_2(t)$.

Figure 6.2: The state trajectories of IADP, ZSADP, and TADP under a complex simulation environment.

For TADP, according to [132], the corresponding cost function follows

$$V_T = \int_t^\infty x^\top Q x + \mathcal{W}(u_T) + \rho v_T^\top v_T + l_M^2 + d_M^2 \, d\tau, \tag{6.58}$$

where $\rho = 0.1$. The approximate optimal control follows $\hat{u}_T = -\beta \tanh\Big(\frac{1}{2\beta} g^\top \nabla \Phi^\top \hat{W}_T\Big)$, and the approximate pseudo control follows $\hat{v}_T = -\frac{1}{2\rho} h^\top \nabla \Phi^\top \hat{W}_T$, where $h = (I_{2\times 2} - gg^\dagger)k$. For TADP, the explicit knowledge of $g(x)$ and $k(x)$ in (6.55) is required to construct $\hat{u}_T$ and $\hat{v}_T$.

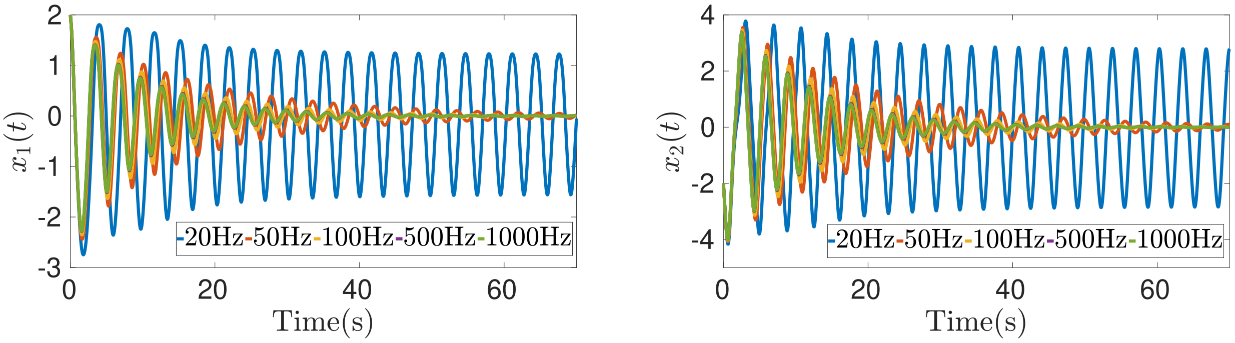
The aforementioned IADP, ZSADP, and TADP all adopt the single critic structure and our developed off-policy weight update law (6.33). To achieve a fair comparison, simulation parameters for three methods are set as same, which is detailly clarified as follows. To get the approximate solutions to the above value functions (6.56)-(6.58), $\Phi(x) = [x_1^2, x_1x_2, x_2^2, x_2^3, x_1x_2^2, x_1^2x_2]^\top$ is chosen. To guarantee the weight convergence, parameters are set as $P = 8$, $\Gamma = 10^{-3} I_{6\times 6}$, $k_c = 0.5$, and $k_e = 0.3$. The initial values are chosen as $x(0) = [2, -2]^\top$, $\hat{u}(0) = 0$, $\hat{d}_Z(0) = 0$ (for ZSADP), and $\hat{v}_T(0) = 0$ (for TADP).

The estimated weight trajectory of IADP shown in Figure 6.1a illustrates that our proposed off-policy weight update law (6.33) enables us to collect real-time data in time and finally achieve weight convergence even under multiple sources of uncertainties and disturbances,. The control trajectories shown in Figure 6.1b, and the state trajectories displayed in Figure 6.2 clarify the enhanced robustness of IADP. Specifically, IADP successfully stabilizes the pendulum under multiple sources of uncertainties and disturbances; however, the robustness of ZSADP and TADP are not enough to tackle such a complex environment. Thus, the control inputs and states of ZSADP and TADP diverge far away immediately when the simulation environment significantly changes at $t = 20$ $s$.

## 6.4.2 Validation of IADP under Different Sampling Rates

This subsection conducts multiple comparative numerical simulations to investigate the influence of different sampling rates on IADP's performance. Note that except for different values of the sampling rate, the conducted simulations in this subsection follow the same simulating environment and parameter settings as Section 6.4.1.

The evolution trajectories of the states $x_1$, $x_2$ under different sampling rates are displayed in Figure 6.3. It is shown that a higher sampling rate leads to better performance. Specif-



(a) The trajectories of $x_1(t)$.

(b) The trajectories of $x_2(t)$.

Figure 6.3: The state trajectories of IADP under different sampling rates.

ically, for the considered robust optimal regulation control task, the sampling frequency of 50 Hz is enough to achieve a satisfying performance. However, a system working under a higher sampling rate is more sensitive to measurement noises, requires faster converters and more storage, and consumes more computational resources. Thus, in practical applications, practitioners need to be aware of the trade-offs mentioned above and choose a suitable sampling rate accordingly.

## 6.5 Summary

This paper presents an efficient and low-cost model-free control strategy for robust optimal stabilization of continuous-time nonlinear control-affine systems. To reduce dependence on accurate mathematical models, the TDE technique permits us to obtain a measured input-state data based incremental dynamics, which is an equivalent representation of the original dynamics, without requiring explicit model knowledge or tedious identification procedures. Then, the HJB equation, which is constructed based on the incremental dynamics, is approximately solved through a single critic structure. The resulting approximate optimal incremental control strategy stabilizes the controlled plant incrementally. Besides, by transforming the critic NN weight learning as a parameter identification process and further using the collected experience data, we develop an efficient weight update law with guaranteed weight convergence. Multiple conducted numerical simulations have shown that IADP outperforms common ADP methods in terms of reduced control efforts and enhanced robustness.

The following properties of our proposed IADP are promising for practical applications: the simultaneous consideration of stability, optimality and robustness, the utilized simplified single critic structure, and the easily implemented off-policy weight update law. However, our proposed IADP builds on the assumption that the full internal states and their derivatives are available, which restricts IADP's generality and practicality. Thus, future works attempt to combine state observer and state derivative estimation techniques with IADP to address the scenario when internal states and their derivatives are not measurable. In addition, since the efficacy of IADP depends on accurate sensor measurements, we will investigate and address the influence of sensor biases or delays on our developed IADP.

# Time-Delayed Data Informed Reinforcement Learning for Optimal Tracking Control 7

This chapter develops a time-delayed data informed RL approach to enable autonomous systems precisely track planned safe reference trajectories in a disturbed environment. This chapter is organized as follows. Section 7.1 first presents the OTCP formulation. Then, the development of the incremental subsystems using the decoupled control and the time-delayed signals is clarified in Section 7.2. The utilized decoupled control technique endows our proposed tracking control scheme with scalability to systems in arbitrary dimensions. By reusing the time-delayed signals to estimate the unknown model knowledge, a laborious system identification process is avoided to achieve model-free tracking control. Thereafter, Section 7.3 presents our proposed tracking control scheme. Section 7.4 elucidates the approximate solution to the OTCP, wherein the intractable value function approximation problem of a high-dimensional system is conquered by solving multiple manageable low-dimensional subsystem value function approximation problems. Our developed tracking control scheme is experimentally and numerically validated in Section 7.5 and Section 7.6, respectively. Finally, the summary is drawn in Section 7.7.

## 7.1 Problem Formulation

This section assumes that the investigated plant (unknown dynamics) could be described by the E-L equation:

$$M(q)\ddot{q} + N(q, \dot{q}) + F(\dot{q}) = \tau, \tag{7.1}$$

where $M(q) : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ is the symmetric positive definite inertia matrix; $N(q, \dot{q}) = C(q, \dot{q})\dot{q} + G(q) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$, $C(q, \dot{q}) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{n\times n}$ is the matrix of centrifugal and Coriolis terms, $G(q) : \mathbb{R}^n \to \mathbb{R}^n$ represents the gravitational terms; $F(\dot{q}) : \mathbb{R}^n \to \mathbb{R}^n$ denotes the viscous friction; $q$, $\dot{q}$, $\ddot{q} \in \mathbb{R}^n$ are the vectors of angles, velocities, and accelerations, respectively; $\tau \in \mathbb{R}^n$ represents the input torque vector. Note that the mathematical model (7.1) is provided here for later theoretical analysis. The explicit value of $M(q)$, $C(q, \dot{q})$, $G(q)$, and $F(\dot{q})$ are unavailable to practitioners.

The objective is to design a model-free tracking control strategy $\tau$ to enable the plant (7.1) to track a bounded and smooth reference signal $x_d = [q_d^\top, \dot{q}_d^\top]^\top \in \mathbb{R}^{2n}$ while minimizing a predefined performance function. The considered high-dimensional and highly uncertain controlled plant (7.1) provides difficulty in solving the OTCP mentioned above.

## 7.2 Incremental Subsystem

This section benefits from the decoupled control technique and time-delayed signals [114], [133] to develop model-free incremental subsystems. The formulated incremental subsystems are equivalent to the dynamics (7.1), but no explicit model information is required. Specifically, the decoupled control technique is utilized to divide the high-dimensional system into multiple low-dimensional subsystems. Then, time-delayed data is used to estimate the unknown dynamics as well as the decoupled control related coupling terms. Here the constructed incremental subsystems serve as basis to design the model-free tracking control strategy in Section 7.3, and allow us to address the scalability problem of the value function approximation in Section 7.4.1.

The high-dimensional system (7.1) can be decoupled into multiple subsystems, wherein the $i$-th subsystem reads

$$M_{ii}\ddot{q}_i + \sum_{j=1,j\neq i}^{n} M_{ij}\ddot{q}_j + N_i + F_i = \tau_i, \quad i = 1, 2, \cdots, n. \tag{7.2}$$

Let $x_i = [x_{i_1}, x_{i_2}]^\top = [q_i, \dot{q}_i]^\top \in \mathbb{R}^2$, and $u_i = \tau_i \in \mathbb{R}$. We rewrite (7.2) as

$$\dot{x}_{i_1} = x_{i_2}, \tag{7.3a}$$

$$\dot{x}_{i_2} = f_i + g_i u_i, \tag{7.3b}$$

where $f_i = -(\sum_{j=1,j\neq i}^{n} M_{ij}\ddot{q}_j + N_i + F_i)/M_{ii} \in \mathbb{R}$, and $g_i = 1/M_{ii} \in \mathbb{R}$ are unknown. Clearly, $f_i$ and $g_i$ are upper bounded since $M(q)$, $N(q, \dot{q})$, and $F(\dot{q})$ in (7.1) are upper bounded [134]. Throughout this article, each subsystem is assumed to be controllable.

The unknown functions $f_i$ and $g_i$ hinder us to directly design tracking controllers based on the subsystem (7.3). Departing from common methods that identify the unknown $f_i$, $g_i$ explicitly through a tedious identification process [47], [135]–[139], we exploit time-delayed signals to estimate the unknown model knowledge. To achieve time delay estimation, we first introduce a predetermined constant $\bar{g}_i \in \mathbb{R}^+$ and multiply $\bar{g}_i^{-1}$ on (7.3b),

$$\bar{g}_i^{-1}\dot{x}_{i_2} = h_i + u_i, \tag{7.4}$$

where $h_i = (\bar{g}_i^{-1} - g_i^{-1})\dot{x}_{i_2} + g_i^{-1} f_i \in \mathbb{R}$ is a lumped term that embodies the unknown model knowledge $f_i$, $g_i$ of (7.3b).

Then, with a sufficiently high sampling rate (see Remark 6.3), by utilising time-delayed signals [52], [108], [109], the unknown $h_i$ in (7.4) could be estimated as

$$\hat{h}_i = h_{i,0} = \bar{g}_i^{-1}\dot{x}_{i_{2,0}} - u_{i,0}, \tag{7.5}$$

where $u_{i,0} = u_i(t - L)$, $\dot{x}_{i_{2,0}} = \dot{x}_{i_2}(t - L)$. We directly choose the delay time $L \in \mathbb{R}^+$ as the sampling period (the smallest achievable value of $L$ in practical implementations) to achieve an accurate estimation of $h_i$ [52].

Substituting (7.5) into (7.4), we get

$$\dot{x}_{i_2} = \dot{x}_{i_{2,0}} + \bar{g}_i(\Delta u_i + \xi_i), \tag{7.6}$$

where $\Delta u_i = u_i - u_{i,0} \in \mathbb{R}$ is the incremental control input; $\xi_i = h_i - \hat{h}_i \in \mathbb{R}$ denotes the so-called TDE error that is proved to be bounded in Lemma 7.1 of Section 7.3.

Combining (7.3) with (7.6), we finally obtain the $i$-th incremental subsystem dynamics

$$\dot{x}_{i_1} = x_{i_2}, \tag{7.7a}$$

$$\dot{x}_{i_2} = \dot{x}_{i_{2,0}} + \bar{g}_i(\Delta u_i + \xi_i), \tag{7.7b}$$

which is an equivalent of the original $i$-th subsystem (7.3) but without using explicit model information. The guideline to select the required suitable $\bar{g}_i$ to construct the $i$-th incremental subsystem (7.7) is provided in Remark 7.1. Here the time-delayed data ($\dot{x}_{i_{2,0}}$ and $u_{i,0}$ in particular) informs the value function learning process clarified in Section 7.4 about one model-free representation (7.7) of the original controlled plant (7.1). Thereby, we could achieve model-free control and also have a mathematical form of dynamics to conduct rigorous theoretical analysis using rich analysis tools from the control field.

**Remark 7.1.** *According to [134], it is reasonable to assume that $\underline{m}_i \leq M_{ii} \leq \overline{m}_i$, where $\underline{m}_i, \overline{m}_i \in \mathbb{R}^+$. According to* (7.3)*, $g_i = \frac{1}{M_{ii}}$. Thus, $\frac{1}{\overline{m}_i} \leq g_i \leq \frac{1}{\underline{m}_i}$ holds. To achieve $\left\| 1 - g_i(k)\bar{g}_i^{-1} \right\| < 1$ required in* (7.31)*, $\bar{g}_i > \frac{1}{2} g_i$ needs to be satisfied. Therefore, we could choose $\bar{g}_i > \frac{1}{2\underline{m}_i}$. The prior knowledge of $M_{ii}$ provides designers with hints to choose a suitable $\bar{g}_i$.*

This section has decoupled the original $n$-D (7.1) into $n$ equivalent 2-D incremental subsystems (7.7). Accordingly, we transform the OTCP of (7.1) into $n$ sub-OTCPs regarding (7.7). The following section will present our developed tracking control scheme by focusing on the sub-OTCP of (7.7).

**Remark 7.2.** *The decoupled control technique facilitates realtime control for a high-dimensional system by distributing the computation load into multiple processors. However, the utilized decoupled control technique presents a challenge of getting the value of the coupling terms, which is usually addressed by add-on tools such as (RBF) NNs [140], [141] that accompany with additional parameter tuning efforts and computational loads. Unlike these works, the time-delayed signals, which are initially used to achieve model-free control in a low-cost and easily implemented way (only a constant $\bar{g}_i$ to be debugged), enjoys an additional benefit that compensates the coupling terms in* (7.2)*.*

**Remark 7.3.** *The required state derivative information (i.e., $\dot{x}_{i_{2,0}}$) to construct the incremental subsystem dynamics* (7.7) *may not be directly measurable. In practice, the unmeasurable state derivative is usually obtained via numerical differentiation [118], [133]. Section 7.5 experimentally validates the effectiveness of the numerical differentiation technique. Alternatively, the state derivative could be estimated by the robust exact differentiator [117], or derivative estimator [116], [142], which is beyond the scope of this chapter.*

## 7.3 Tracking Control Scheme

This section details our proposed tracking control scheme, as displayed in Figure 7.1, to solve the sub-OTCP of (7.7). The incremental control input to be designed follows

$$\Delta u_i = \Delta u_{i_f} + \Delta u_{i_b}, \tag{7.8}$$

where the incremental dynamic inversion based $\Delta u_{i_f} \in \mathbb{R}$ serves to transform the time-varying sub-OTCPs into equivalent time-invariant sub-robust optimal regulation control problems (sub-RORCPs) in Section 7.3.1; and $\Delta u_{i_b} \in \mathbb{R}$ is the incremental control policy to optimally drive the tracking error to zero in Section 7.3.2. The detailed procedures to design $\Delta u_{i_f}$ and $\Delta u_{i_b}$ are detailly clarified in Section 7.3.1 and Section 7.3.2, respectively.

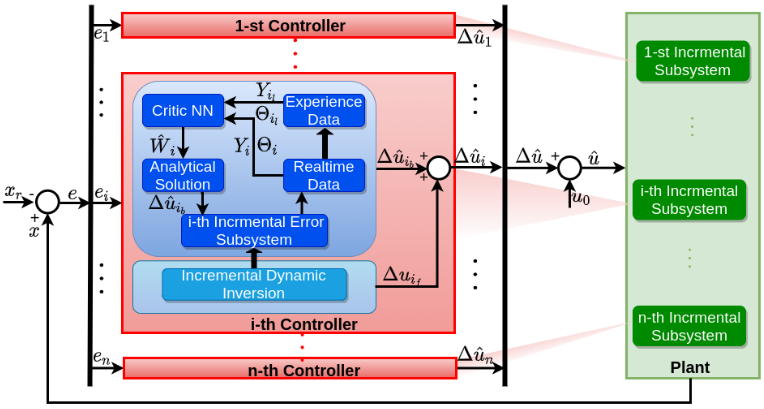


Figure 7.1: Schematic of the tracking control strategy. The original OTCP is first decoupled into the sub-OTCPs of the incremental subsystems, as illustrated in Section 7.2; Then, the sub-OTCPs are converted into the equivalent sub-RORCPs of the incremental error subsystems, as clarified in Section 7.3.1; Finally, the solutions to the transformed sub-RORCPs are learned via parallel training, as described in Section 7.3.2 and Section 7.4.

### 7.3.1 Generation of Incremental Error Subsystem

This subsection formulates the $i$-th incremental error subsystem via the properly chosen $\Delta u_{i_f}$. The formulated incremental error subsystem converts the sub-OTCP regarding (7.7) into its sub-RORCP, and facilitates the development of the optimal incremental control policy in Section 7.3.2. The detailed procedures to design $\Delta u_{i_f}$ and to generate the incremental error subsystem are as follows.

Let $e_i = [e_{i_1}, e_{i_2}]^\top \in \mathbb{R}^2$, where $e_{i_1} = x_{i_1} - q_{d_i} \in \mathbb{R}$ and $e_{i_2} = x_{i_2} - \dot{q}_{d_i} \in \mathbb{R}$. Combining with (7.7b) and (7.8) yields

$$\dot{e}_{i_2} = \dot{x}_{i_{2,0}} + \bar{g}_i(\Delta u_{i_f} + \Delta u_{i_b} + \xi_i) - \ddot{x}_{r_i}. \tag{7.9}$$

Designing the required $\Delta u_{i_f}$ in (7.9) as

$$\Delta u_{i_f} = \bar{g}_i^{-1}(\ddot{x}_{r_i} - \dot{x}_{i_{2,0}} - k_{i_1}e_{i_1} - k_{i_2}e_{i_2}), \tag{7.10}$$

and substituting (7.10) into (7.9), we get

$$\dot{e}_{i_2} = -k_{i_1}e_{i_1} - k_{i_2}e_{i_2} + \bar{g}_i\Delta u_{i_b} + \bar{g}_i\xi_i, \tag{7.11}$$

where $k_{i_1}$, $k_{i_2} \in \mathbb{R}^+$. Recall that $\dot{e}_{i_1} = e_{i_2}$. Then, combining with (7.11), we obtain the $i$-th incremental error subsystem

$$\dot{e}_i = A_i e_i + B_i \Delta u_{i_b} + B_i \xi_i, \tag{7.12}$$

where $A_i = \begin{bmatrix} 0 & 1 \\ -k_{i_1} & -k_{i_2} \end{bmatrix} \in \mathbb{R}^{2\times 2}$, and $B_i = \begin{bmatrix} 0 \\ \bar{g}_i \end{bmatrix} \in \mathbb{R}^2$. The sub-OTCP of (7.7) illustrated in Section 7.2 aims to drive the values of $e_i$ to zero in an optimal manner. This is equivalent to the sub-RORCP of the incremental error subsystem (7.12) given the unknown $\xi_i$. In other words, this subsection transforms the sub-OTCP of (7.7) into the sub-RORCP regarding (7.12) by designing $\Delta u_{i_f}$ in the form of (7.10).

**Remark 7.4.** *The developed* $\Delta u_{i_f}$ (7.10) *here acts as a supplementary control input to the* $\Delta u_{i_b}$ *designed in Section 7.3.2. In particular, the utilized* $\Delta u_{i_f}$ *generates an incremental error subsystem* (7.12)*. Then, we train* $\Delta u_{i_b}$ *in Section 7.3.2 based on the incremental error subsystem formulated in this subsection. This practice departs from most of existing ADP related works for the OTCP [20], [26], [143], wherein the tracking control strategies are trained on one specific reference trajectory dynamics. Thus, the flexibility of our developed tracking control scheme against varying desired trajectories is improved without directly using reference signals during the learning process.*

### 7.3.2 Optimal Incremental Control Policy

This subsection develops an optimal incremental control policy to solve the sub-RORCP of (7.12), i.e., robustly stabilizing the tracking error to zero in an optimal manner. Departing from common solutions to OTCPs [20], [26], [143], we additionally introduce a TDE error related term into the value function such that the influence of the TDE error on the controller performance is lessened under an optimization framework.

Given $\xi_i$ in (7.12) is unknown, thus the available incremental error subsystem for later analysis follows

$$\dot{e}_i = A_i e_i + B_i \Delta u_{i_b}. \tag{7.13}$$

To stabilize (7.13) in an optimal manner, the value function is considered as

$$V_i(t) = \int_t^{\infty} r_i(e_i(\nu), \Delta u_{i_b}(\nu))\, d\nu, \tag{7.14}$$

where $r_i(e_i, \Delta u_{i_b}) = e_i^\top Q_i e_i + W_i(\Delta u_{i_b}) + \bar{\xi}_{oi}^2$. The quadratic term $e_i^\top Q_i e_i$, where $Q_i \in \mathbb{R}^{2\times 2}$ is a positive definite matrix, is introduced to improve the tracking precision. The input penalty function $W_i(\Delta u_{i_b})$ follows

$$\mathcal{W}_i(\Delta u_{i_b}) = 2\int_0^{\Delta u_{i_b}} \beta \tanh^{-1}(\vartheta/\beta)\, d\vartheta, \tag{7.15}$$

which is utilized to punish and enforce the optimal incremental control input as $\|\Delta u_{i_b}\| \le \beta \in \mathbb{R}^+$. The limited $\Delta u_{i_b}$ is beneficial since a severe interruption might lead to an abrupt change of $\Delta u_{i_b}$, which might destabilize the learning process introduced in Section 7.4. The utilized TDE error related term $\bar{\xi}_{oi}^2$ in $r_i(e_i, \Delta u_{i_b})$ allows designers to attenuate the TDE error during the optimization process. The explicit form of $\bar{\xi}_{oi}$ follows $\bar{\xi}_{oi} = \bar{c}_i \|\Delta u_{i_b}\|$, where $\bar{c}_i \in \mathbb{R}^+$. The rationality of designing $\bar{\xi}_{oi}$ in the above form and the requirement for an appropriate $\bar{c}_i$ are provided in Theorem 7.1.

For $\Delta u_{i_b} \in \Psi$, where $\Psi$ is the set of admissible incremental control policies [52, Definition 1], the associated optimal value function follows

$$V_i^* = \min_{\Delta u_{i_b} \in \Psi} \int_t^{\infty} r_i(e_i(\nu), \Delta u_{i_b}(\nu))\, d\nu. \tag{7.16}$$

Define the Hamiltonian function as

$$H_i(e_i, \Delta u_{i_b}, \nabla V_i) = r(e_i, \Delta u_{i_b}) + \nabla V_i^T (A_i e_i + B_i \Delta u_{i_b}), \tag{7.17}$$

where $\nabla(\cdot) = \partial(\cdot)/\partial e_i$. Then, $V_i^*$ satisfies the HJB equation

$$0 = \min_{\Delta u_{i_b} \in \Psi} [H_i(e_i, \Delta u_{i_b}, \nabla V_i^*)]. \tag{7.18}$$

Assume that the minimum of (7.16) exists and is unique [19], [52]. By using the stationary optimality condition on the HJB equation (7.18), we gain an analytical-form optimal incremental control strategy as

$$\Delta u_{i_b}^* = -\beta \tanh\left(\frac{1}{2\beta} B_i^\top \nabla V_i^*\right). \tag{7.19}$$

To obtain $\Delta u_{i_b}^*$, we need to solve the HJB equation (7.18) to determine the value of $\nabla V_i^*$, which is detailly clarified in Section 7.4. In the following part of this subsection, based on the TDE error bound given in Lemma 7.1, we prove in Theorem 7.1 that the optimal incremental control policy $\Delta u_{i_b}^*$ (7.19) regarding (7.13) is the solution to the sub-RORCP of (7.12).

**Lemma 7.1.** *Given a sufficiently high sampling rate, $\exists \bar{\xi}_i \in \mathbb{R}^+$, there holds $\|\xi_i\| \leq \bar{\xi}_i$.*

*Proof.* combining (7.4) with (7.5), the TDE error for the $i$-th subsystem (7.7) follows

$$\begin{aligned} \xi_i &= h_i - \hat{h}_i = h_i - h_{i,0} \\ &= (\bar{g}_i^{-1} - g_i^{-1})\Delta \dot{x}_{i_2} + (g_{i,0}^{-1} - g_i^{-1})\dot{x}_{i_{2,0}} + g^{-1}(f_i - f_{i,0}) + (g_i^{-1} - g_{i,0}^{-1}) f_{i,0}, \end{aligned} \tag{7.20}$$

where $\Delta \dot{x}_{i_2} = \dot{x}_{i_2} - \dot{x}_{i_{2,0}}$. Combining with (7.3b) , (7.7b) and (7.8), $\Delta \dot{x}_{i_2}$ follows

$$\begin{aligned} \Delta \dot{x}_{i_2} &= f_i + g_i u_i - f_{i,0} - g_{i,0} u_{i,0} = g_i \Delta u_i + (g_i - g_{i,0}) u_{i,0} + f_i - f_{i,0} \\ &= g_i (\Delta u_{i_f} + \Delta u_{i_b}) + (g_i - g_{i,0}) u_{i,0} + f_i - f_{i,0}. \end{aligned} \tag{7.21}$$

Substituting (7.21) into (7.20), we get

$$\xi_i = (g_i \bar{g}_i^{-1} - 1)\Delta u_{i_f} + (g_i \bar{g}_i^{-1} - 1)\Delta u_{i_b} + \delta_{1i}, \tag{7.22}$$

where $\delta_{1i} = \bar{g}_i^{-1}(g_i - g_{i,0}) u_0 + \bar{g}_i^{-1}(f_i - f_{i,0})$.

For simplicity, we denote $\mu_i = \ddot{x}_{r_i} - k_{i_1} e_{i_1} - k_{i_2} e_{i_2} \in \mathbb{R}$. According to (7.5) and (7.10), $\Delta u_{i_f}$ in (7.22) follows

$$\begin{aligned} \Delta u_{i_f} &= \bar{g}_i^{-1}(\mu_i - \bar{g}_i h_{i,0} - \bar{g}_i u_{i,0}) \\ &= \bar{g}_i^{-1}\mu_i - (\bar{g}_i^{-1} - g_{i,0}^{-1})\dot{x}_{i_{2,0}} - g_{i,0}^{-1} f_{i,0} - u_{i,0} \\ &= \bar{g}_i^{-1}\mu_i - (\bar{g}_i^{-1} - g_{i,0}^{-1})(f_{i,0} + g_{i,0} u_{i,0}) - g_{i,0}^{-1} f_{i,0} - u_{i,0} \\ &= \bar{g}_i^{-1}\mu_i - \bar{g}_i^{-1}(f_{i,0} + g_{i,0} u_{i,0}) = \bar{g}_i^{-1}(\mu_i - \mu_{i,0}) - \bar{g}_i^{-1}(\dot{x}_{i_{2,0}} - \mu_{i,0}), \end{aligned} \tag{7.23}$$

where $\mu_{i,0} = \ddot{x}_{r_{i,0}} - k_{i_1} e_{i_{1,0}} - k_{i_2} e_{i_{2,0}}$. Besides, combining (7.7b) with (7.8), we get

$$\begin{aligned} \dot{x}_{i_2} &= \dot{x}_{i_{2,0}} + \bar{g}_i(\Delta u_{i_f} + \Delta u_{i_b}) + \bar{g}_i \xi_i \\ &= \dot{x}_{i_{2,0}} + \bar{g}_i \bar{g}_i^{-1}(\mu_i - \dot{x}_{i_{2,0}}) + \bar{g}_i \Delta u_{i_b} + \bar{g}_i \xi_i \\ &= \mu_i + \bar{g}_i \Delta u_{i_b} + \bar{g}_i \xi_i. \end{aligned} \tag{7.24}$$

Based on the result shown in (7.24), we get

$$\xi_i = \bar{g}_i^{-1}(\dot{x}_{i_2} - \mu_i - \bar{g}_i \Delta u_{i_b}). \tag{7.25}$$

Accordingly, the following equation establishes

$$\xi_{i,0} = \bar{g}_i^{-1}(\dot{x}_{i_{2,0}} - \mu_{i,0} - \bar{g}_i \Delta u_{i_{b,0}}). \tag{7.26}$$

Based on the result given in (7.26), (7.23) is rewritten as

$$\begin{aligned}\Delta u_{i_f} &= \bar{g}_i^{-1}(\mu_i - \mu_{i,0}) - \bar{g}_i^{-1}(\dot{x}_{i_{2,0}} - \mu_{i,0} - \bar{g}_i \Delta u_{i_{b,0}}) - \Delta u_{i_{b,0}} \\ &= \bar{g}_i^{-1}(\mu_i - \mu_{i,0}) - \xi_{i,0} - \Delta u_{i_{b,0}}.\end{aligned} \tag{7.27}$$

Substituting (7.27) into (7.22) yields

$$\xi_i =(1 - g_i\bar{g}_i^{-1})\xi_{i,0} + (1 - g_i\bar{g}_i^{-1})\bar{g}_i^{-1}(\mu_{i,0} - \mu_i) + (1 - g_i\bar{g}_i^{-1})(\Delta u_{i_{b,0}} - \Delta u_{i_b}) + \delta_{1i}. \tag{7.28}$$

In discrete-time domain, (7.28) could be represented as

$$\xi_i(k) =(1 - g_i(k)\bar{g}_i^{-1})\xi_i(k-1) + (1 - g_i(k)\bar{g}_i^{-1})\Delta\tilde{u}_{i_b} + \delta_{1i} + \delta_{2i}, \tag{7.29}$$

where $\Delta\tilde{u}_{i_b} = \Delta u_{i_b}(k-1) - \Delta u_{i_b}(k)$, $\delta_{2i} = (1 - g_i(k)\bar{g}_i^{-1})\bar{g}_i^{-1}(\mu_i(k-1) - \mu_i(k))$.

The constrained input $\|\Delta u_{i_b}(k)\| \leq \beta$ implies that the following equation holds

$$\|\Delta\tilde{u}_{i_b}\| \leq \|\Delta u_{i_b}(k-1)\| + \|\Delta u_{i_b}(k)\| \leq 2\beta. \tag{7.30}$$

We choose the value of $\bar{g}_i$ to meet $\left\|1 - g_i(k)\bar{g}_i^{-1}\right\| \leq \iota_i < 1$, where $\iota_i \in \mathbb{R}^+$. Under a sufficiently high sampling rate, it is reasonable to assume that there exists $\bar{\delta}_{1i}, \bar{\delta}_{2i} \in \mathbb{R}^+$ such that $\|\delta_{1i}\| \leq \bar{\delta}_{1i}$ , and $\|\delta_{2i}\| \leq \iota_i\bar{\delta}_{2i}$. Then, the following equations hold:

$$\begin{aligned}\|\xi_i(k)\| &\leq \iota_i \|\xi_i(k-1)\| + \iota_i \|\Delta\tilde{u}_{i_b}\| + \bar{\delta}_{1i} + \iota_i\bar{\delta}_{2i} \\ &\leq \iota_i^2 \|\xi_i(k-2)\| + (\iota_i^2 + \iota_i) \|\Delta\tilde{u}_{i_b}\| + (\iota_i + 1)(\bar{\delta}_{1i} + \iota_i\bar{\delta}_{2i}) \\ &\leq \cdots \\ &\leq \iota_i^k \|\xi_i(0)\| + \frac{\bar{\delta}_{1i} + \iota_i\bar{\delta}_{2i}}{1 - \iota_i} + \frac{\iota_i \|\Delta\tilde{u}_{i_b}\|}{1 - \iota_i} \\ &\leq \iota_i^k \|\xi_i(0)\| + \frac{\bar{\delta}_{1i} + \iota_i\bar{\delta}_{2i}}{1 - \iota_i} + \frac{2\iota_i\beta}{1 - \iota_i} = \bar{\xi}_i.\end{aligned} \tag{7.31}$$

As $k \to \infty$, $\bar{\xi}_i \to \frac{\bar{\delta}_{1i} + \iota_i\bar{\delta}_{2i}}{1-\iota_i} + \frac{2\iota_i\beta}{1-\iota_i}$. □

**Theorem 7.1.** *Consider the system* (7.12) *with a sufficiently high sampling rate, if there exists a scalar* $\bar{c}_i \in \mathbb{R}^+$ *such that the following inequality is satisfied*

$$\bar{\xi}_i < \bar{c}_i \|\Delta u_{i_b}\|, \tag{7.32}$$

*the optimal incremental control policy* (7.19) *regulates the tracking error to a small neighbourhood around zero while minimizing the value function* (7.14).

*Proof.* $V_i^*$ is a positive definite function, i.e., $V_i^*(e_i) \geq 0$ and iff $e_i = 0$, $V_i^*(e_i) = 0$. Thus, $V_i^*$ could serve as a candidate Lyapunov function. Taking time derivative of $V_i^*$ along the $i$-th incremental error subsystem (7.12) yields

$$\dot{V}_i^* = \nabla V_i^*(A_i e_i + B_i \Delta u_{i_b}^*) + \nabla V_i^* B_i \xi_i. \tag{7.33}$$

According to (7.17) and (7.18), the following equations establish

$$\begin{aligned}\nabla V_i^*(A_i e_i + B_i \Delta u_{i_b}^*) &= -e_i^\top Q_i e_i - W_i(\Delta u_{i_b}^*) - \bar{\xi}_{oi}^2 \\ \nabla V_i^* B_i &= -2\beta \tanh^{-1}(\Delta u_{i_b}^*/\beta).\end{aligned} \tag{7.34}$$

Substituting (7.34) into (7.33) yields

$$\dot{V}_i^* = -e_i^\top Q_i e_i - W_i(\Delta u_{i_b}^*) - \bar{\xi}_{oi}^2 - 2\beta \tanh^{-1}(\Delta u_{i_b}^*/\beta)\xi_i. \tag{7.35}$$

As for the $W_i(\Delta u_{i_b}^*)$ in (7.35), according to our previous result [52, Theorem 1], it follows that

$$W_i(\Delta u_{i_b}^*) = \beta^2 \sum_{j=1}^{m} \Big(\tanh^{-1}(\Delta u_{i_b}^*/\beta)\Big)^2 - \epsilon_{u_i}, \tag{7.36}$$

where $\epsilon_{u_i} \le \frac{1}{2}\bar{g}_i^2 \nabla V_i^{*\top} \nabla V_i^*$. Given that there exists $b_{\nabla V_i^*} \in \mathbb{R}^+$ such that $\|\nabla V_i^*\| \le b_{\nabla V_i^*}$. Thus, we could rewrite the bound of $\epsilon_{u_i}$ as $\epsilon_{u_i} \le b_{\epsilon_{ui}} \le \frac{1}{2}\bar{g}_i^2 b_{\nabla V_i^*}^2$.

Then, substituting (7.36) into (7.35), we get

$$\dot{V}_i^* = - e_i^\top Q_i e_i - [\beta \tanh^{-1}(\Delta u_{i_b}^*/\beta) + \xi_i]^2 - (\bar{\xi}_{oi}^2 - \xi_i^\top \xi_i) + b_{\epsilon_{ui}}. \tag{7.37}$$

We choose $\bar{\xi}_{oi} = \bar{c}_i \|\Delta u_{i_b}\|$, and $\bar{c}_i$ is picked to satisfy $\bar{c}_i \|\Delta u_{i_b}\| > \bar{\xi}_i$, where $\bar{\xi}_i$ is defined in (7.31). Then, the following equation holds

$$\dot{V}_i^* \le -e_i^\top Q_i e_i + b_{\epsilon_{ui}}. \tag{7.38}$$

Thus, if $-\lambda_{\min}(Q_i)\|e_i\|^2 + b_{\epsilon_{ui}} < 0$, $\dot{V}_i^* < 0$ holds. Here $\lambda_{\min}(\cdot)$ denotes the minimum eigenvalues of a symmetric real matrix. Finally, it concludes that states of the *i*-th incremental error subsystem (7.12) converges to the residual set

$$\Omega_{e_i} = \{e_i | \|e_i\| \le \sqrt{b_{\epsilon_{ui}}/\lambda_{\min}(Q_i)}\}. \tag{7.39}$$

□

Theorem 7.1 implies that the optimal incremental control policy $\Delta u_{i_b}^*$ (7.19) robustly stabilize (7.12). It has been clarified in Section 7.3.1 that the sub-RORCP of (7.12) equals to the sub-OTCP of (7.7) based on our designed $\Delta u_{i_f}$ (7.10). Thus, the designed $\Delta u_{i_b}^*$ and $\Delta u_{i_f}$ solve the sub-OTCP of (7.7) together.

## 7.4 Approximate Solutions

This section uses a parallel critic learning structure to seek for the approximate solutions to the value functions of the HJB equations (7.18) of $n$ incremental error subsystems (7.12). By reinvestigating the online NN weight learning process from a parameter identification perspective, we develop a simple yet efficient off-policy critic NN weight update law with guaranteed weight convergence by exploiting realtime and experience data together.

### 7.4.1 Value Function Approximation

For $e_i \in \Omega$, where $\Omega \subset \mathbb{R}^2$ is a compact set, the continuous optimal value function (7.16) is approximated by an critic agent as [19]

$$V_i^* = W_i^{*\top} \Phi_i(e_i) + \epsilon_i(e_i), \tag{7.40}$$

where $W_i^* \in \mathbb{R}^{N_i}$ is the critic NN weight, $\Phi_i(e_i) : \mathbb{R}^2 \to \mathbb{R}^{N_i}$ represents the activation function, and $\epsilon_i(e_i) \in \mathbb{R}$ denotes the approximation error.

**Remark 7.5.** *The utilized decoupled control technique in Section 7.2 solves the curse of complexity problem in* (7.40). *In particular, the constructed critic NN* (7.40) *relies on the error* $e_i \in \mathbb{R}^2$ *of the incremental error subsystem* (7.12). *The 2-D* $e_i$ *allows us to construct a low-dimensional* $\Phi_i(e_i)$ *(easy to choose) to approximate its associated* $V_i^*$ *regardless of the value of the system dimension* $n$. *For example, the 4-D activation functions* $\Phi_i(e_i)$ *in a fixed structure are chosen for subsystems of a 3-DoF robot manipulator in Section 7.5, and 6-DoF quadrotor in Section 7.6. Otherwise, for a global approximation, i.e.,* $V^* = W^{*\top}\Phi(e) + \epsilon(e)$ *with the tracking error* $e = x - x_d \in \mathbb{R}^{2n}$, *the dimension of* $\Phi(e)$ *increases exponentially as* $n$ *increases.*

To facilitate the later theoretical analysis, an assumption that is common in ADP related works is provided here.

**Assumption 7.1.** *[19] There exist constants* $b_{\epsilon_i}, b_{\epsilon_{ei}}, b_{\epsilon_{hi}}, b_{\Phi_i}, b_{\Phi_{ei}} \in \mathbb{R}^+$ *such that* $\|\epsilon_i(e_i)\| \leq b_{\epsilon_i}$, $\|\nabla\epsilon_i(e_i)\| \leq b_{\epsilon_{ei}}$, $\|\epsilon_{hi}\| \leq b_{\epsilon_{hi}}$, $\|\Phi_i(e_i)\| \leq b_{\Phi_i}$, *and* $\|\nabla\Phi_i(e_i)\| \leq b_{\Phi_{ei}}$.

Given a fixed incremental control input $\Delta u_{i_b}$, combining (7.18) with (7.40) yields

$$W_i^{*\top}\nabla\Phi_i(A_i e_i + B_i\Delta u_{i_b}) + r_i(e_i, \Delta u_{i_b}) = \epsilon_{h_i}, \tag{7.41}$$

where the residual error $\epsilon_{h_i} = -\nabla\epsilon_i^\top(A_i e_i + B_i\Delta u_{i_b}) \in \mathbb{R}$. The NN parameterized (7.41) is able to be written into a LIP form as

$$\Theta_i = -W_i^{*\top}Y_i + \epsilon_{h_i}, \tag{7.42}$$

where $\Theta_i = r_i(e_i, \Delta u_{i_b}) \in \mathbb{R}$, and $Y_i = \nabla\Phi_i(A_i e_i + B_i\Delta u_{i_b}) \in \mathbb{R}^{N_i}$. The values of $\Theta_i$ and $Y_i$ are both available to practitioners given the measurable $e_i$ and $\Delta u_{i_b}$. This formulated LIP form (7.42) enables the learning of $W_i^*$ to be equivalent to a parameter identification problem of a LIP system, which facilitates the development of an efficient weight update law in the subsequent subsection.

## 7.4.2 Critic NN Weight Update Law

An approximation of (7.42) follows

$$\hat{\Theta}_i = -\hat{W}_i^\top Y_i, \tag{7.43}$$

where $\hat{W}_i \in \mathbb{R}^{N_i}$, $\hat{\Theta}_i \in \mathbb{R}$ are estimates of $W_i^*$ and $\Theta_i$, respectively. To enable $\hat{W}_i$ converge to $W_i^*$, we design an off-policy critic NN weight update law for each subsystem as

$$\dot{\hat{W}}_i = -\Gamma_i k_{t_i} Y_i \tilde{\Theta}_i - \sum_{l=1}^{P_i} \Gamma_i k_{e_i} Y_{i_l} \tilde{\Theta}_{i_l}, \tag{7.44}$$

to update the critic NN weight $\hat{W}_i$ in a parallel way to minimize $E_i = \frac{1}{2}\tilde{\Theta}_i^\top\tilde{\Theta}_i$, where $\tilde{\Theta}_i = \Theta_i - \hat{\Theta}_i \in \mathbb{R}$. Here $\Gamma_i \in \mathbb{R}^{N_i \times N_i}$ is a constant positive definite gain matrix; $k_{t_i}, k_{e_i} \in \mathbb{R}^+$ are used to trade-off the contribution of realtime and experience data to the online NN weight learning process; $P_i \in \mathbb{R}^+$ is the number of the utilized recorded experience data.

To guarantee the weight convergence of (7.44), as proved in Theorem 7.2, the exploited experience data should be sufficient rich to satisfy the rank condition in Assumption 7.2. This assumption could be satisfied by sequentially reusing experience data in practice [52].

**Assumption 7.2.** *Given an experience buffer* $\mathfrak{B}_i = [Y_{i_1}, ..., Y_{i_{P_i}}] \in \mathbb{R}^{N_i \times P_i}$, *there holds* $rank(\mathfrak{B}_i) = N_i$.

**Theorem 7.2.** *Given Assumption 7.2, the NN weight learning error* $\tilde{W}_i$ *converges to a small neighbourhood around zero.*

*Proof.* The proof is similar to Theorem 5.2. Thus, it is omitted here for simplicity. □

The guaranteed weight convergence of $\hat{W}_i$ to $W_i^*$ in Theorem 7.2 permits us to use a computation-simple single critic NN learning structure for each subsystem, where the estimated critic NN weight $\hat{W}_i$ is directly used to construct the approximate optimal incremental control strategy:

$$\Delta\hat{u}_{i_b} = -\beta \tanh\left(\frac{1}{2\beta} B_i^\top \nabla\Phi_i^\top \hat{W}_i\right). \tag{7.45}$$

Finally, combining with (7.8), (7.10), and (7.45), we get the overall control input applied at the $i$-th subsystem (7.3)

$$\hat{u}_i = u_{i,0} + \Delta u_{i_f} + \Delta\hat{u}_{i_b}. \tag{7.46}$$

Based on the theoretical analysis mentioned above, we provide the main conclusions in the following theorem.

**Theorem 7.3.** *Given Assumptions 7.1–7.2, for a sufficiently large* $N_i$, *the off-policy critic NN weight update law* (7.44), *and the approximate optimal incremental control policy* (7.45) *guarantee the tracking error and the NN weight learning error uniformly ultimately bounded.*

*Proof.* Consider the candidate Lyapunov function for the $i$-th incremental error subsystem (7.12) as

$$L_i = V_i^* + \frac{1}{2}\tilde{W}_i^\top \Gamma_i^{-1} \tilde{W}_i. \tag{7.47}$$

By denoting $L_{i_1} = V_i^*$, its derivative follows

$$\begin{aligned} \dot{L}_{i_1} &= \nabla V_i^{*\top}(A_i e_i + B_i \Delta\hat{u}_{i_b} + B_i \xi_i) \\ &= \nabla V_i^{*\top}(A_i e_i + B_i \Delta u_{i_b}^*) + \nabla V_i^{*\top} B_i \xi_i + \nabla V_i^{*\top} B_i (\Delta\hat{u}_{i_b} - \Delta u_{i_b}^*). \end{aligned} \tag{7.48}$$

Substituting (7.34) into (7.48) reads

$$\begin{aligned} \dot{L}_{i_1} = &-e_i^\top Q_i e_i - \mathcal{W}(\Delta u_{i_b}^*) - \bar{\xi}_{oi}^2 - 2\beta \tanh^{-1}\left(\Delta u_{i_b}^*/\beta\right) \xi_i \\ &- 2\beta \tanh^{-1}\left(\Delta u_{i_b}^*/\beta\right) (\Delta\hat{u}_{i_b} - \Delta u_{i_b}^*). \end{aligned} \tag{7.49}$$

Combining with (7.36) and (7.37), (7.49) follows

$$\begin{aligned} \dot{L}_{i_1} \leq &- e_i^\top Q_i e_i - (\bar{\xi}_{oi}^2 - \|\xi_i\|^2) - \left[\beta \tanh^{-1}\left(\Delta u_{i_b}^*/\beta\right) + \xi_i\right]^2 \\ &+ \frac{1}{2} \nabla V_i^{*\top} B_i B_i^\top \nabla V_i^* - 2\beta \tanh^{-1}\left(\Delta u_{i_b}^*/\beta\right)(\Delta\hat{u}_{i_b} - \Delta u_{i_b}^*). \end{aligned} \tag{7.50}$$

The term $-2\beta \tanh^{-1}\left(\Delta u_{i_b}^*/\beta\right)(\Delta\hat{u}_{i_b} - \Delta u_{i_b}^*)$ in (7.50) follows

$$-2\beta \tanh^{-1}\left(\Delta u_{i_b}^*/\beta\right)(\Delta\hat{u}_{i_b} - \Delta u_{i_b}^*) \leq \beta^2 \left\|\tanh^{-1}\left(\Delta u_{i_b}^*/\beta\right)\right\|^2 \left\|\Delta\hat{u}_{i_b} - \Delta u_{i_b}^*\right\|^2. \tag{7.51}$$

According to (7.19) and (7.40), and the mean-value theorem, the optimal incremental control is rewritten as

$$\Delta u_{i_b}^* = -\beta \tanh\left(\frac{1}{2\beta} B_i^\top \nabla \Phi_i^\top W_i^*\right) - \epsilon_{\Delta u_i^*}, \tag{7.52}$$

where $\epsilon_{\Delta u_i^*} = \frac{1}{2}(1 - \tanh^2(\eta_i)) B_i^\top \nabla \epsilon_i$, and $\eta_i \in \mathbb{R}$ is chosen between $\frac{1}{2\beta} B_i^\top \nabla \Phi_i^\top W_i^*$ and $\frac{1}{2\beta} B_i^\top \nabla V_i^*$. According to $\|\nabla \epsilon_i\| \le b_{\epsilon_{ei}}$ in Assumption 7.1, $\left\|\epsilon_{\Delta u_i^*}\right\| \le \frac{1}{2}\|B_i\| b_{\epsilon_{ei}}$ holds. Then, by combining (7.45) with (7.52), we get

$$\Delta \hat{u}_{i_b} - \Delta u_{i_b}^* = \beta(\tanh(\mathscr{G}_i^*) - \tanh\left(\hat{\mathscr{G}}_i\right)) + \epsilon_{\Delta u_i^*}. \tag{7.53}$$

where $\mathscr{G}_i^* = \frac{1}{2\beta} B_i^\top \nabla \Phi_i^\top W_i^*$, and $\hat{\mathscr{G}}_i = \frac{1}{2\beta} B_i^\top \nabla \Phi_i^\top \hat{W}$. Based on (7.19) and (7.45), the Taylor series of $\tanh(\mathscr{G}_i^*)$ follows

$$\begin{aligned} \tanh(\mathscr{G}_i^*) &= \tanh\left(\hat{\mathscr{G}}_i\right) + \frac{\partial \tanh\left(\hat{\mathscr{G}}_i\right)}{\partial \hat{\mathscr{G}}_i}(\mathscr{G}_i^* - \hat{\mathscr{G}}_i) + \mathcal{O}((\mathscr{G}_i^* - \hat{\mathscr{G}}_i)^2) \\ &= \tanh\left(\hat{\mathscr{G}}_i\right) - \frac{1}{2\beta}(1 - \tanh^2(\hat{\mathscr{G}}_i)) B_i^\top \nabla \Phi_i^\top \tilde{W}_i + \mathcal{O}((\mathscr{G}_i^* - \hat{\mathscr{G}}_i)^2), \end{aligned} \tag{7.54}$$

where $\mathcal{O}((\mathscr{G}_i^* - \hat{\mathscr{G}}_i)^2)$ is a higher order term of the Taylor series. By following [100, Lemma 1], this higher order term is bounded as

$$\left\|\mathcal{O}((\mathscr{G}_i^* - \hat{\mathscr{G}}_i)^2)\right\| \le 2 + \frac{1}{\beta}\|B_i\| b_{\Phi_{ei}} \left\|\tilde{W}_i\right\|. \tag{7.55}$$

Based on (7.54), we rewrite (7.53) as

$$\begin{aligned} \Delta \hat{u}_{i_b} - \Delta u_{i_b}^* &= \beta(\tanh(\mathscr{G}_i^*) - \tanh\left(\hat{\mathscr{G}}_i\right)) + \epsilon_{\Delta u_i^*} \\ &= -\frac{1}{2}(1 - \tanh^2(\hat{\mathscr{G}}_i)) B_i^\top \nabla \Phi_i^\top \tilde{W}_i + \beta \mathcal{O}((\mathscr{G}_i^* - \hat{\mathscr{G}}_i)^2) + \epsilon_{\Delta u_i^*}. \end{aligned} \tag{7.56}$$

Then, by combining (7.55) with (7.56), and given that $\left\|1 - \tanh^2(\hat{\mathscr{G}}_i)\right\| \le 2$, $\left\|\Delta \hat{u}_{i_b} - \Delta u_{i_b}^*\right\|^2$ in (7.51) follows

$$\begin{aligned} \left\|\Delta \hat{u}_{i_b} - \Delta u_{i_b}^*\right\|^2 &\le 3\beta^2 \left\|\mathcal{O}((\mathscr{G}_i^* - \hat{\mathscr{G}}_i)^2)\right\|^2 + 3\left\|\epsilon_{\Delta u_i^*}\right\|^2 + 3\left\|-\frac{1}{2}(1 - \tanh^2(\hat{\mathscr{G}}_i)) B_i^\top \nabla \Phi_i^\top \tilde{W}_i\right\|^2 \\ &\le 6\|B_i\|^2 b_{\Phi_{ei}}^2 \left\|\tilde{W}_i\right\|^2 + 12\beta^2 + \frac{3}{4}\|B_i\|^2 b_{\epsilon_{ei}}^2 + 12\beta\|B_i\| b_{\Phi_{ei}} \left\|\tilde{W}_i\right\|. \end{aligned} \tag{7.57}$$

Based on (7.34), (7.40), Assumption 7.1, and the fact that $\|W_i^*\| \le b_{W_i^*}$, $\left\|\tanh^{-1}(\Delta u_{i_b}^*/\beta)\right\|^2$ in (7.51) follows

$$\begin{aligned} \left\|\tanh^{-1}\left(\Delta u_{i_b}^*/\beta\right)\right\|^2 &= \left\|\frac{1}{4\beta^2} \nabla V_i^{*\top} B_i B_i^\top \nabla V_i^*\right\| \\ &\le \frac{1}{4\beta^2}\|B_i\|^2 b_{\Phi_{ei}}^2 b_{W_i^*}^2 + \frac{1}{4\beta^2} b_{\epsilon_{ei}}^2 \|B_i\|^2 + \frac{1}{2\beta^2}\|B_i\|^2 b_{\Phi_{ei}} b_{\epsilon_{ei}} b_{W_i^*}. \end{aligned} \tag{7.58}$$

Using (7.57) and (7.58), (7.51) reads

$$\begin{aligned} &-2\beta \tanh^{-1}\left(\Delta u_{i_b}^*/\beta\right)(\Delta \hat{u}_{i_b} - \Delta u_{i_b}^*) \le \frac{1}{4}\|B_i\|^2 b_{\Phi_{ei}}^2 b_{W_i^*}^2 \\ &+ \frac{1}{4} b_{\epsilon_{ei}}^2 \|B_i\|^2 + \frac{1}{2}\|B_i\|^2 b_{\Phi_{ei}} b_{\epsilon_{ei}} b_{W_i^*} + 6\|B_i\|^2 b_{\Phi_{ei}}^2 \left\|\tilde{W}_i\right\|^2 \\ &+ 12\beta^2 + \frac{3}{4}\|B_i\|^2 b_{\epsilon_{ei}}^2 + 12\beta\|B_i\| b_{\Phi_{ei}} \left\|\tilde{W}_i\right\|. \end{aligned} \tag{7.59}$$

Substituting (7.59) into (7.50), finally the first term $\dot{L}_{i_1}$ follows

$$\begin{aligned}\dot{L}_{i_1} \leq & - e_i^\top Q_i e_i - (\bar{\xi}_{oi}^2 - \xi_i^\top \xi_i) - \left[\beta \tanh^{-1}\left(\Delta u_{i_b}^*/\beta\right) + \xi_i\right]^2 \\ & + \frac{3}{4}\|B_i\|^2 b_{\Phi_{ei}}^2 b_{W_i^*}^2 + \frac{3}{4} b_{\epsilon_{ei}}^2 \|B_i\|^2 + \frac{3}{2}\|B_i\|^2 b_{\Phi_{ei}} b_{\epsilon_{ei}} b_{W_i^*} + 12\beta^2 + \frac{3}{4}\|B_i\|^2 b_{\epsilon_{ei}}^2 \\ & + 6\|B_i\|^2 b_{\Phi_{ei}}^2 \left\|\tilde{W}_i\right\|^2 + 12\beta \|B_i\| b_{\Phi_{ei}} \left\|\tilde{W}_i\right\|.\end{aligned} \tag{7.60}$$

As for the second term $\dot{L}_W = \frac{1}{2}\tilde{W}_i^\top \Gamma_i^{-1} \tilde{W}_i$, based on (7.44) and Theorem 1 in our previous work [52], it follows

$$\dot{L}_{i_2} \leq -\tilde{W}_i^\top \mathcal{Y}_i \tilde{W}_i + \tilde{W}_i^\top \epsilon_{\tilde{W}_i}. \tag{7.61}$$

where $\mathcal{Y}_i = \sum_{l=1}^{P_i} k_{e_i} Y_{i_l} Y_{i_l}^\top \in \mathbb{R}^{N_i \times N_i}$, and $\epsilon_{\tilde{W}_i} = -k_{t_i} Y_i \epsilon_{h_i} - \sum_{l=1}^{P_i} k_{e_i} Y_{i_l} \epsilon_{h_{il}} \in \mathbb{R}^{N_i}$. The boundness of $Y_i$ and $\epsilon_{h_i}$ results in bounded $\epsilon_{\tilde{W}_i}$. Thus, there exists $\bar{\epsilon}_{\tilde{W}_i} \in \mathbb{R}^+$ such that $\left\|\epsilon_{\tilde{W}_i}\right\| \leq \bar{\epsilon}_{\tilde{W}_i}$. According to Assumption 7.2, $\mathcal{Y}_i$ is positive definite. Thus, (7.61) could be rewritten as

$$\dot{L}_{i_2} \leq -\lambda_{\min}(\mathcal{Y}_i) \left\|\tilde{W}_i\right\|^2 - \bar{\epsilon}_{\tilde{W}_i} \left\|\tilde{W}_i\right\|. \tag{7.62}$$

Finally, as for $\dot{L}_i$, substituting (7.60) and (7.61) into (7.47), we get

$$\dot{L}_i \leq -\mathcal{A}_i - \mathcal{B}_i \left\|\tilde{W}_i\right\|^2 + \mathcal{C}_i \left\|\tilde{W}_i\right\| + \mathcal{D}_i, \tag{7.63}$$

where $\mathcal{A}_i = e_i^\top Q_i e_i + (\bar{\xi}_{oi}^2 - \xi_i^\top \xi_i) + \left[\beta \tanh^{-1}\left(\Delta u_{i_b}^*/\beta\right) + \xi_i\right]^2$, $\mathcal{B}_i = \lambda_{\min}(\mathcal{Y}_i) - 6\|B_i\|^2 b_{\Phi_{ei}}^2$, $\mathcal{C}_i = 12\beta\|B_i\| b_{\Phi_{ei}} + \bar{\epsilon}_{\tilde{W}_i}$, and $\mathcal{D}_i = \frac{3}{4}\|B_i\|^2 b_{\Phi_{ei}}^2 b_{W_i^*}^2 + \frac{3}{2} b_{\epsilon_{ei}}^2 \|B_i\|^2 + \frac{3}{2}\|B_i\|^2 b_{\Phi_{ei}} b_{\epsilon_{ei}} b_{W_i^*} + 12\beta^2$. Let the parameters be chosen such that $\mathcal{B}_i > 0$. Since $\mathcal{A}_i$ is positive definite, the above Lyapunov derivative (7.63) is negative if

$$\left\|\tilde{W}_i\right\| > \frac{\mathcal{C}_i}{2\mathcal{B}_i} + \sqrt{\frac{\mathcal{C}_i^2}{4\mathcal{B}_i^2} + \frac{\mathcal{D}_i}{\mathcal{B}_i}}. \tag{7.64}$$

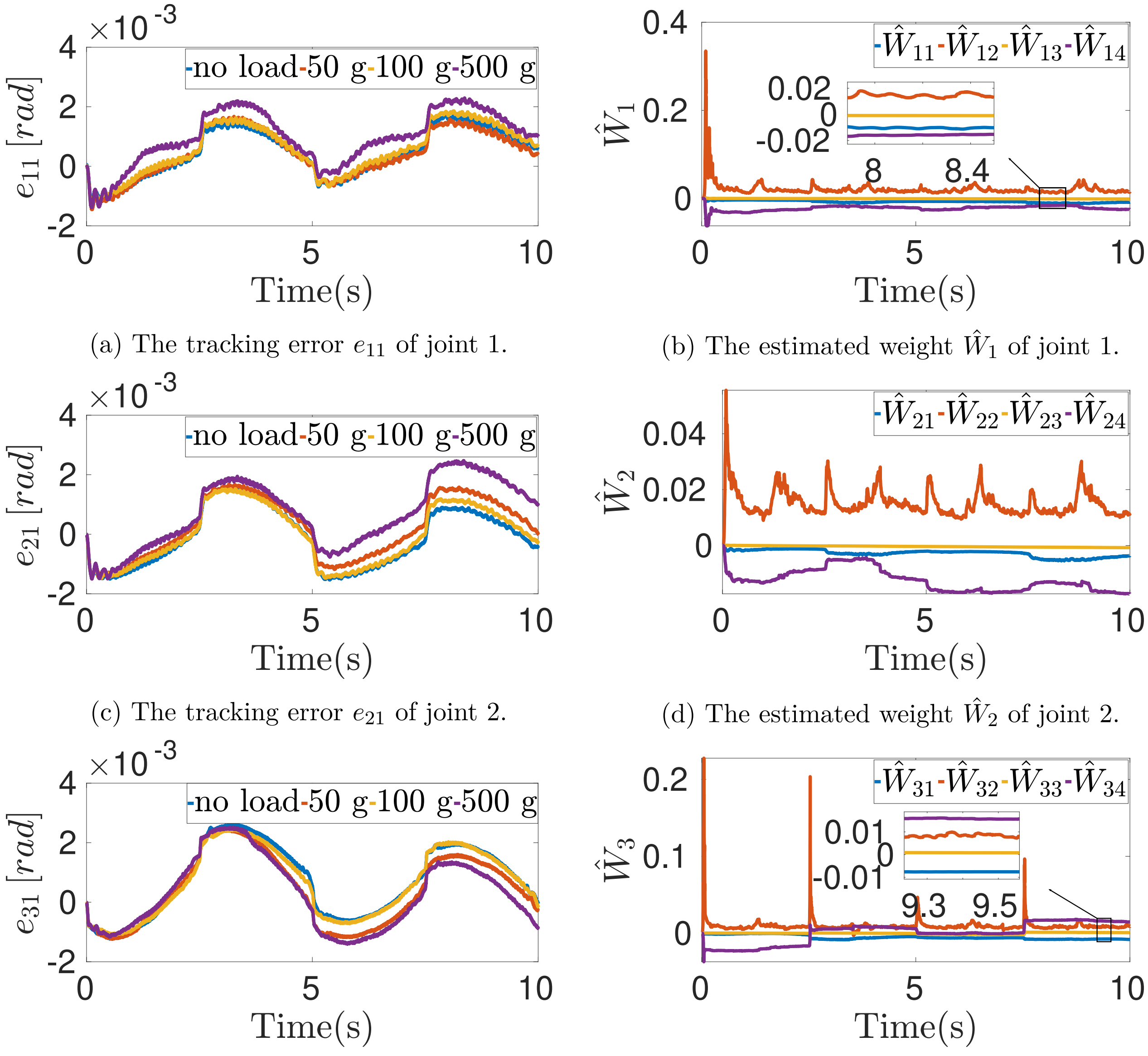
Thus, the weight learning error of the critic agent converges to the residual set

$$\tilde{\Omega}_{\tilde{W}_i} = \left\{\tilde{W}_i \middle| \left\|\tilde{W}_i\right\| \leq \frac{\mathcal{C}_i}{2\mathcal{B}_i} + \sqrt{\frac{\mathcal{C}_i^2}{4\mathcal{B}_i^2} + \frac{\mathcal{D}_i}{\mathcal{B}_i}}\right\}. \tag{7.65}$$

□

## 7.5 Experimental Validation

This section experimentally validates the efficiency of our proposed tracking control scheme on a 3-DoF robot manipulator (see Figure A.1 in Appendix A.1). Note that the common ADP based tracking control scheme [20] is impractical to conduct the experimental validation presented here. The tracking control scheme developed in [20] requires one 12-D augmented system [20]. It is not trivial to pick a suitable high-dimensional activation function to accomplish the accurate approximation of the value function of the constructed 12-D augmented system. Even though a high-dimensional activation function is available, the realtime performance of the corresponding weight update law is poor for practical experiments.



(a) The tracking error $e_{11}$ of joint 1.

(b) The estimated weight $\hat{W}_1$ of joint 1.

(c) The tracking error $e_{21}$ of joint 2.

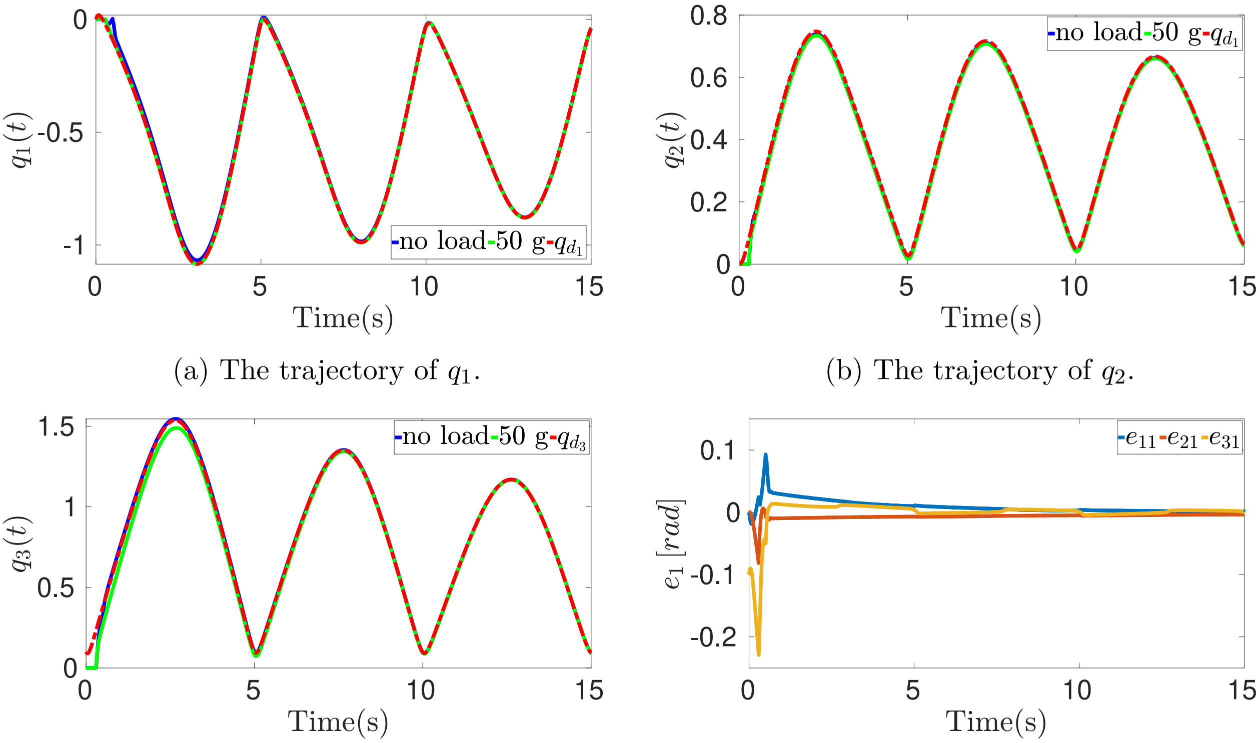
(d) The estimated weight $\hat{W}_2$ of joint 2.

(e) The tracking error $e_{31}$ of joint 3.

(f) The estimated weight $\hat{W}_3$ of joint 3.

Figure 7.2: The trajectories of the tracking error $e_{i_1}$ and the estimated weight $\hat{W}_i$ under different payloads, $i = 1, 2, 3$.

During our experiment, the measured angular position is numerically differentiated to compute the angular velocity and acceleration of the robot [118], [133]. The robot manipulator is driven to track the desired trajectory $x_d = [q_d^\top, \dot{q}_d^\top]^\top \in \mathbb{R}^6$, where $q_d = (1 + \sin(\frac{t}{2} - \frac{\pi}{2}))k_p \in \mathbb{R}^3$. To simulate varying tasks, we set $k_p = [0.3, 0.6, 1]^\top$ for $t \in [0, 5)$, and $k_p = [0.2, 0.5, 0.8]^\top$ for $t \in [5, 10]$. Our developed approach adopts the 4-D activation function $\Phi_i(e_i) = [e_{i_1}^2, e_{i_2}^2, e_{i_1}e_{i_2}, e_{i_2}^3]^\top$ for the $i$-th decoupled subsystem, $i = 1, 2, 3$. The utilized low-dimensional activation function $\Phi_i(e_i)$ in a fixed structure exemplify the scalability and practicability. Given the sampling rate is 1kHz, accordingly, we choose the delay time as $L = 0.001$s. The simulation parameters for subsystems 1-3 are set as: $Q_i = \text{diag}(300, 40000)$, $\bar{c}_i = 200$, $\Gamma_i = \text{diag}(100, 4, 0.1, 16)$, $k_{t_i} = 0.2$, $k_{e_i} = 0.01$, $P_i = 10$, $k_{i_1} = 8$, $k_{i_2} = 8$, $i = 1, 2, 3$;



(a) The trajectory of $q_1$.

(b) The trajectory of $q_2$.

(c) The trajectory of $q_3$.

(d) The trajectory of $e_1$ (no load case).

Figure 7.3: The joint space trajectories under different payloads.

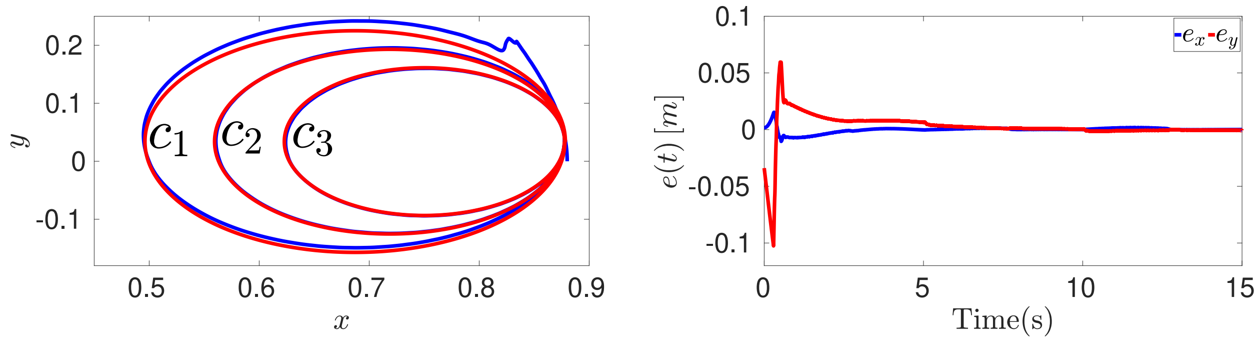
and $\beta = 0.1$, $\bar{g}_1 = 40$, $\bar{g}_2 = 46$, and $\bar{g}_3 = 54$.

The trajectories of $e_{i_1}$, $i = 1, 2, 3$ under different payloads (installed to the end effector of the robot manipulator) are displayed in Figure 7.2a, Figure 7.2c, and Figure 7.2e, respectively. It is shown that our developed tracking control scheme efficiently tracks the desired trajectory $x_d$ with a satisfying tracking precision and robustness against varying payloads. The parallel training results (the 500g payload case is displayed for demonstration) of $\hat{W}_i$ are displayed in Figure 7.2b, Figure 7.2d, and Figure 7.2f. We obtain the desired weight convergence for each subsystem using the realtime and experience data together. This validates the realtime learning performance of our developed weight update law (7.44) even for a high-dimensional system.

To further show the superiority of our developed tracking control scheme under different tasks, we drive the end effector of the robot manipulator to track three different reference circles in task space sequentially. Circle 1: center $c_1 = (0.68, 0.05)$ and radius $r_1 = 0.2$; Circle 2: center $c_2 = (0.72, 0.05)$ and radius $r_2 = 0.16$; Circle 3: center $c_3 = (0.75, 0.05)$ and radius $r_3 = 0.12$. We use the Robotics toolbox [144] to conduct the inverse kinematics calculation to get the associated joint space trajectories of Circles 1-3, which are inputs of our proposed tracking control scheme. The required kinematics information to conduct the inverse kinematics calculation is referred to Appendix A.1. More details of experimental settings are referred to Table 7.1. The associated tracking trajectories in joint space and task space are displayed in Figure 7.3 and Figure 7.4, respectively. The satisfying tracking performance validates the efficiency of our developed tracking control scheme.

Table 7.1: The parameter settings for the robot manipulator.

| | |
|---|---|
| **Initial value conditions** | $x(0) = [0,0,0,0]^\top$, $u(0) = [0,0]^\top$,<br>$\bar{g}_1 = 14$, $\bar{g}_2 = 32$, $\bar{g}_3 = 80$,<br>$k_{i_1} = 8$, $k_{i_2} = 8$, $\hat{W}_i(0) = 0_{4\times 1}$, $i = 1,2,3$ |
| **Cost function parameters** | $Q_1 = \mathrm{diag}(16,10)$, $Q_2 = \mathrm{diag}(18,10)$,<br>$Q_3 = \mathrm{diag}(0.2,0.1)$, $\beta = 1$, $\bar{c}_i = 4$ , $i = 1,2,3$ |
| **Weight learning parameters** | $k_{t_i} = 0.1$, $k_{e_i} = 0.1$, $P_i = 10$, $i = 1,2,3$<br>$\Gamma_1 = 0.01\,\mathrm{diag}(I_{1\times 4})$, $\Gamma_2 = 0.03\,\mathrm{diag}(I_{1\times 4})$,<br>$\Gamma_3 = 0.01\,\mathrm{diag}(I_{1\times 4})$. |



(a) The trajectory of the robot end effector.

(b) The trajectory of the tracking error.

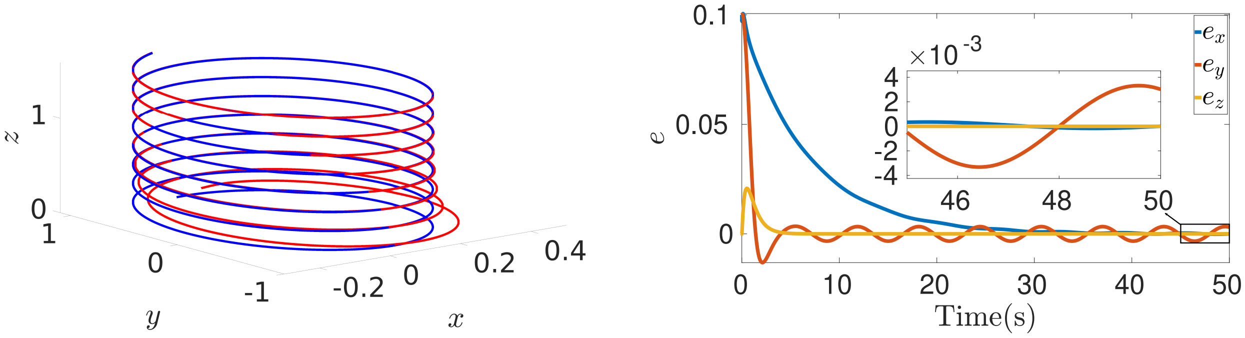
Figure 7.4: The task space trajectories of the circle tracking scenario.

## 7.6 Numerical Simulation

This section further certifies the effectiveness of our proposed tracking control scheme under a high-dimensional quadrotor tracking task. The quadrotor [145] is driven to track the desired spiral reference trajectory $x_r = [\frac{3}{10\sin(t)}, \cos(t), \frac{t}{10\pi}, 0]^\top \in \mathbb{R}^3$, $t \in [0,50]$. The associated parameter settings to conduct the numerical simulations are referred to Table 7.2. The detailed procedures to decouple the 6-DoF quadrotor into 6 subsystems are referred to Appendix A.2. For subsystems 1-6, we adopt the same activation functions used in Section 7.5. As displayed in Figure 7.5, we obtain a satisfying tracking performance via our developed approach.

Table 7.2: The parameter settings of a quadrotor OTCP.

| | |
|---|---|
| **Initial value conditions** | $\xi(0) = [0.1,1.1,0]^\top$, $\eta(0) = [0,0,0]^\top$, $u(0) = [0,0,0.5]^\top$<br>$\bar{g}_i = 300$, $i = 1,2,3$; $\bar{g}_i = 60000$, $i = 4,5,6$,<br>$k_{i_1} = 3$, $k_{i_2} = 3$, $\hat{W}_i(0) = 0_{4\times 1}$, $i = 1,\cdots,6$. |
| **Cost function parameters** | $Q_i = \mathrm{diag}(1,1)$, $\bar{c}_i = 4$,<br>$\beta = 0.1$, $i = 1,\cdots,6$. |
| **Weight learning parameters** | $k_{t_i} = 1$, $k_{e_i} = 0.01$, $P_i = 6$,<br>$\Gamma_i = 0.01\,\mathrm{diag}(I_{1\times 4})$, $i = 1,\cdots,6$. |



(a) The position trajectory.

(b) The position tracking error.

Figure 7.5: The position tracking performance of a quadrotor.

## 7.7 Summary

This chapter develops a time-delayed data informed RL based tracking control scheme to address the limitations of existing RL based solutions to the OTCP. Through the decoupled control and time-delayed data, the investigated OTCP of a high-dimensional system is divided into multiple sub-OTCPs of incremental subsystems. Then, the sub-OTCPs are transformed into sub-RORCPs that are approximately solved by a parallel critic learning structure. The proposed tracking control scheme is developed with rigorous theoretical analysis of system stability and weight convergence. The experimental and numerical simulation results validate that our proposed model-free tracking control strategy could be applied to high-dimensional systems with the flexibility of different tracking tasks.

Departing from available solutions to the OTCP, our developed tracking control scheme settles the curse of complexity problem in value function approximation from a decoupled way, circumvents the learning inefficiency regarding varying desired trajectories by avoiding introducing a reference trajectory dynamics into the learning process, and requires neither an accurate nor identified dynamics through the time delay estimation technique. However, the input saturation is not addressed in the current work, which remains our future work. Besides, the effectiveness and the superiority of the parallel critic learning structure will be validated under a high-dimensional system.

# 8 Conclusion and Future Directions

This dissertation presents our works to empower autonomous systems with resilience to safely interact with unforeseen environments and guaranteed or even optimal performance to accomplish given tasks. The techniques from the control and learning communities are used interchangeably to develop methods with theoretical guarantees and generalization to uncertain scenarios.

## 8.1 Conclusion

### 8.1.1 Part I Conclusion

Our works in Part I are mainly rooted in the control community, concerning domains including system identification, adaptive control, robust control, and backstepping. Chapter 2 safely improves the tracking performance of one partially unknown robot manipulators based on BLF, TF-CL, and backstepping. Chapter 3 explicitly considers the control-level attainable performance bound into the planning level and uses time-delayed data to achieve the model-free (kinematics-free and/or dynamics-free) control. Thereby, the provable safe execution of autonomous systems suffering from uncertainties and disturbances is realized. Chapter 4 presents an integrated perception and control approach that utilizes instantaneous local sensory data to stimulate safe feedback control strategies with fast adaptation to diverse uncertain environments without building a global map.

The less information on dynamics and environment that is required, the stronger the robustness. Chapter 2 accomplishes online identification of the partially unknown dynamics using the physical structure. Chapter 3 relaxes the model structure by exploiting time-delayed signals to achieve model-free control. The methods developed in Chapter 2 and Chapter 3 both require a nominal understating of the operating environment given that the inputs (i.e., safe reference trajectories) to our designed tracking controllers are planned using prior-developed maps. Chapter 4 moves one step further to remove the requirement of an accurate map by directly coupling perceptional signals with control inputs. Regarding safety issues, Chapter 2 and Chapter 3 confine states into safe regions. Chapter 4 ensures safety by enforcing controlled plants to stay away from unsafe regions. In summary, Part I utilizes informative data (historical data in Chapter 2, time-delayed data in Chapter 3, and perceptual data in Chapter 4 in particular) to refresh the traditional control methods. The resulting learning-based control strategies in Chapters 2-4 safely improve the closed-loop performance despite uncertainties.

### 8.1.2 Part II Conclusion

In Part II, we attempt to achieve the safe approximate optimal control under uncertainty via RL based optimization framework. RL provides designers with avenues to solve the optimal control problem of continuous time nonlinear systems. We additionally embed the RL based optimal control with robustness and safety guarantees. Chapter 5 proposes an off-policy risk-sensitive RL based control framework to jointly optimize task performance and constraint satisfaction in a disturbed environment. Chapter 6 presents a new formulation for model-free robust optimal regulation of continuous-time nonlinear systems. The proposed RL based approach utilizes measured input-state data to allow the design of the approximate optimal incremental control strategy, stabilizing the controlled system incrementally under model uncertainties, environmental disturbances, and input saturation. Chapter 7 develops one time-delayed data informed RL based approximate optimal tracking control strategy. Departing from available solutions to the optimal tracking control problem, our developed tracking control scheme settles the curse of complexity problem, circumvents the learning inefficiency, and requires neither an accurate nor identified dynamics.

The nominal knowledge and an assumed disturbance bound is utilized in Chapter 5 to present the closed-loop stability and safety under uncertainty. A further step is taken in Chapter 6 and Chapter 7 where time-delayed signals are used to represent the controlled plant in an incremental form. The resulting model-free incremental system facilitates model-free control and also provides a mathematical form of dynamics for the stability analysis. Towards the safe control, Chapter 5 studies the safe optimization problem via introducing the risk-sensitive input and state penalty terms into the value function. Chapter 6 utilizes a safety filter to enforce the safety constraint satisfaction after the learning process. Chapter 7 investigates the safe operation problem through a systematic view. An approximate optimal tracking control strategy with high tracking accuracy and strong robustness drives autonomous systems to track the planned safe reference trajectory precisely even in a disturbed environment. In summary, Part II builds on the RL based optimization framework and enables the approximate optimal control strategy to cope with uncertain and unsafe scenarios.

## 8.2 Future Directions

The topic investigated in this dissertation intertwines multiple disciplines. Potential future research directions are listed as follows.

**Revisit the data quality.** The techniques developed in Part I and Part II are data-hungry. For example, the realtime and historical data are required in CL to ensure the parameter convergence, and the time-delayed data are needed in IADP to achieve model-free control. The efficacy of the data-based techniques is heavily dependent on perfect measurement data. Current works assume available perfect measurements without considering potential missing data, process and observation noises. Considering practicability, expanding our developed approaches to imperfect measurements is essential.

**Explore deep neural network and deep reinforcement learning.** Current works adopt the linear function approximator to facilitate the theoretical analysis of the system stability and weight convergence. However, the utilized simple architecture lacks the generality to complex and high-dimensional problems. The solution is to turn to the nonlinear function approximator – deep neural network. Deep reinforcement learning gets state-of-the-art performance on control tasks due to the powerful approximation ability of the deep neural network, which fully captures the complexities and nonlinear proprieties of the investigated problems. However, the accompanying problem is that conducting theoretical analysis based on this complicated representation scheme is nontrivial. It remains to explore novel algorithm structures and training strategies to learn deep neural network based control policies with theoretical guarantees.

**Bridge the gap between planning and control.** To accomplish the safe autonomous operation, common approaches often firstly plan a safe desired trajectory that a tracking controller then follows. However, the planned collision-free trajectory does not imply actual safe execution given the tracking controllers' inefficiency caused by model uncertainties and/or environmental disturbances. The deviation between the actual execution trajectory and the planned safe trajectory might result in unsafe scenarios. It is promising to use tools from the control and learning communities to solve the safe autonomous operation problem in an end-to-end way to avoid gaps among different levels.

**Inform reinforcement learning algorithms with theoretical guarantees.** This dissertation mainly focuses on the control field and utilises RL to support the controller design. It remains to investigate how to exploit theoretical-analysis tools and available knowledge in the control field to improve the transparency and interpretability of RL algorithms. An exciting point is to preserve the promising exploration ability of RL while meeting specific theoretical guarantees during the online learning process.

**Competitive game among different algorithms.** Our works attempt to empower autonomous systems with intelligence to survive in an uncertain environment. Different strategies are developed. Besides, there also exist works that uses different algorithms to achieve the same goal. Then, the question arise, which algorithm is more efficient to solve certain problems? Except from comparing different algorithms using common performance indexes such as time, computational resources, one interesting comparison is to design a properly competitive game for different algorithms. In competitive games, such as autonomous racing game and soccer game, intelligent robots driven by different algorithms would compete with each other to check which kind of algorithm is more efficient.

A

# Background Information

## A.1 Dynamics and Kinematics of Robot Manipulator

This section provides detailed dynamics and kinematics knowledge of the 3-DoF robot manipulator (see Figure A.1) used for the experimental validations of our developed approaches.

This 3-DoF robot manipulator is created by Chair of Automatic Control Engineering (LSR), Technical University of Munich (TUM). The manipulator is confined in the horizontal plane and actuated by 3 Maxon torque motors with a turn ration of 1:100. The incremental encoders offer the joint position measurements with a resolution of 2000. The sensors and actuators are connected with the computer using a peripheral component interconnect (PCI) communication card. The executable algorithm is created by MATLAB 2017a in Ubuntu 14.04 LTS with the first-order Euler solver at the sampling rate of 1kHz.

### Dynamics Information

The E-L equation of the 3-DoF robot manipulator follows

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + F\dot{q} = \tau, \tag{A.1}$$

where $q = [q_1, q_2, q_3]^\top \in \mathbb{R}^3$, $\dot{q} = [\dot{q}_1, \dot{q}_2, \dot{q}_3]^\top \in \mathbb{R}^3$, $M(q) = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{12} & m_{22} & m_{23} \\ m_{13} & m_{23} & m_{33} \end{bmatrix} \in \mathbb{R}^{3\times 3}$, $C(q,\dot{q})\dot{q} = [N_1, N_2, N_3]^\top \in \mathbb{R}^3$, and $F = \begin{bmatrix} f_1 & 0 & 0 \\ 0 & f_2 & 0 \\ 0 & 0 & f_3 \end{bmatrix} \in \mathbb{R}^{3\times 3}$. For brevity, $f_1 = f_2 = f_3 = f$ is assumed for the viscous friction. Note that the robot manipulator is confined in the horizontal plane. Thus, the gravity term is omitted in (A.1).

Each element of the inertial matrix $M(q)$ reads

$$\begin{aligned} m_{11} &= p_1 + p_2 c_{23} + p_3 c_2 + p_4 c_3 \\ m_{12} &= p_5 + p_6 c_{23} + p_7 c_2 + p_4 c_3 \\ m_{13} &= p_8 + p_6 c_{23} + p_9 c_3 \\ m_{22} &= p_5 + p_{10} c_3 \\ m_{33} &= p_8 \end{aligned}$$

where $c_2 = \cos(q_2)$, $c_3 = \cos(q_3)$, and $c_{23} = \cos(q_2 + q_3)$.

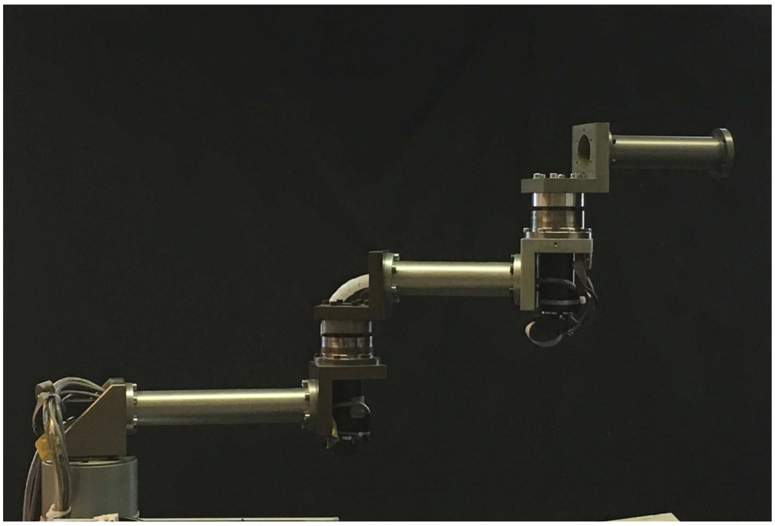


Figure A.1: The 3-DoF robot manipulator for experimental validations.

The explicit forms of $N_1$, $N_2$ and $N_3$ follows

$$\begin{aligned} N_1 =& -p_2(s_{23}\dot{q}_1\dot{q}_3 + s_{23}\dot{q}_2\dot{q}_3 + s_{23}\dot{q}_1\dot{q}_2) - p_3 s_2\dot{q}_1\dot{q}_2 - p_4(s_3\dot{q}_1\dot{q}_3 + s_3\dot{q}_2\dot{q}_3) \\ &- p_6(s_{23}\dot{q}_2^2 + s_{23}\dot{q}_3^2) - p_7 s_2\dot{q}_2^2 - p_9 s_3\dot{q}_3^2 \\ N_2 =& p_7 s_2\dot{q}_1^2 + p_6 s_{23}\dot{q}_1^2 - p_9 s_3\dot{q}_3^2 - p_4 s_3\dot{q}_1\dot{q}_3 - p_4 s_3\dot{q}_2\dot{q}_3 \\ N_3 =& p_9(s_3\dot{q}_1^2 + s_3\dot{q}_2^2) + p_6 s_{23}\dot{q}_1^2 + p_4 s_3\dot{q}_1\dot{q}_2 \end{aligned}$$

where $s_2 = \sin q_2$, $s_3 = \sin q_3$, and $s_{23} = \sin(q_2 + q_3)$.

### Kinematics Information

The Cartesian space position $p = [x, y]^\top \in \mathbb{R}^2$ of the robot manipulator's end-effector reads

$$p = h(q), \tag{A.2}$$

where $h(q) : \mathbb{R}^3 \to \mathbb{R}^2$ is the forward kinematics. The explicit form of $h(q)$ follows

$$h(q) = \begin{bmatrix} l_1 c_1 + l_2 c_{12} + l_3 c_{123} \\ l_1 s_1 + l_2 s_{12} + l_3 s_{123} \end{bmatrix}, \tag{A.3}$$

where $l_1 = 0.3$ m, $l_2 = 0.24$ m, $l_3 = 0.34$ m are lengths of joint 1, joint 2 and joint 3; and $c_{123} = \cos(q_1 + q_2 + q_3)$, $s_{123} = \sin(q_1 + q_2 + q_3)$.

## A.2 Incremental Subsystems of Quadrotor

This section presents the detailed procedures to decouple a 6-DoF quadrotor into 6 incremental subsystems.

Let $\zeta = [x, y, z]^\top \in \mathbb{R}^3$, and $\eta = [\phi, \theta, \psi]^\top \in \mathbb{R}^3$ represent the absolute linear position and Euler angles defined in the inertial frame, respectively. The E-L equation of a quadrotor follows (see [77])

$$m\ddot{\zeta} + mgI_z = RT_B \tag{A.4a}$$

$$J(\eta)\ddot{\eta} + C(\eta,\dot{\eta})\dot{\eta} = \tau_B, \tag{A.4b}$$

where $m \in \mathbb{R}^+$ denotes the mass of the quadrotor; $g \in \mathbb{R}^+$ is the gravity constant; $I_z = [0,0,1]^\top$ represents a column vector; $T_B = [0,0,T]^\top \in \mathbb{R}^3$, where $T \in \mathbb{R}$ is the thrust in the direction of the body $z$-axis; $\tau_B = [\tau_\phi, \tau_\theta, \tau_\psi]^\top \in \mathbb{R}^3$ denotes the torques in the direction of the corresponding body frame angles; $R$, $J(\eta)$, $C(\eta,\dot{\eta}) \in \mathbb{R}^{3\times 3}$ represent the rotation matrix, Jacobian matrix, and Coriolis term, respectively. Their explicit forms and values are referred to [77].

Expanding the translational dynamics (A.4a) yields

$$\begin{aligned} \ddot{x} &= \frac{1}{m}T(C_\psi S_\theta C_\phi + S_\psi S_\phi) \\ \ddot{y} &= \frac{1}{m}T(S_\psi S_\theta C_\phi - C_\psi S_\phi) \\ \ddot{z} &= -g + \frac{1}{m}TC_\theta C_\phi, \end{aligned} \tag{A.5}$$

where $C_{(\cdot)}$ and $S_{(\cdot)}$ denote $\cos(\cdot)$ and $\sin(\cdot)$, respectively.

Introducing pseudo controls $u_1 = T(C_\psi S_\theta C_\phi + S_\psi S_\phi)$, $u_2 = T(S_\psi S_\theta C_\phi - C_\psi S_\phi)$, and $u_3 = TC_\theta C_\phi$, and denoting $x_{11} = x$, $x_{12} = \dot{x}$, $x_{21} = y$, $x_{22} = \dot{y}$, $x_{31} = z$, $x_{32} = \dot{z}$, we finally decouple the transnational dynamics (A.4a) into the following three subsystems

$$\dot{x}_{11} = x_{12},\ \dot{x}_{12} = \frac{1}{m}u_1 \tag{A.6a}$$

$$\dot{x}_{21} = x_{22},\ \dot{x}_{22} = \frac{1}{m}u_2 \tag{A.6b}$$

$$\dot{x}_{31} = x_{32},\ \dot{x}_{32} = -g + \frac{1}{m}u_3. \tag{A.6c}$$

Following the same procedures (7.2)–(7.7) clarified in Section 7.2, we get three subsystems for the rotational dynamics (A.4b):

$$\dot{x}_{41} = x_{42},\ \dot{x}_{42} = -\frac{H_1}{J_{11}} + \frac{1}{J_{11}}u_4 \tag{A.7a}$$

$$\dot{x}_{51} = x_{52},\ \dot{x}_{52} = -\frac{H_2}{J_{22}} + \frac{1}{J_{22}}u_5 \tag{A.7b}$$

$$\dot{x}_{61} = x_{62},\ \dot{x}_{32} = -\frac{H_3}{J_{33}} + \frac{1}{J_{33}}u_6, \tag{A.7c}$$

where $H_i = \sum_{j=1,j\neq i}^{3} J_{ij}\ddot{\eta}_j + C_i\dot{\eta}_j \in \mathbb{R}$, $i = 1,2,3$; $u_4 = \tau_\phi$, $u_5 = \tau_\theta$, and $u_6 = \tau_\psi$.

The aforementioned procedures (A.5)-(A.7) allow us to get 6 subsystems. Then, we design controllers to drive the quadrotor (A.4) to track the predefined reference trajectory $x_r = [x_d, y_d, z_d, \psi_d]^\top$. Note that after the explicit values of pseudo controls $u_1$, $u_2$, and $u_3$ are gotten, we obtain the trust $T$, and reference angles $\phi_d$, $\theta_d$ as

$$T = \sqrt{u_1^2 + u_2^2 + u_3^2} \tag{A.8}$$

$$\phi_d = \arctan\left(\frac{u_1 S_\psi - u_2 C_\psi}{\sqrt{(u_1 C_\psi + u_2 S_\psi)^2 + u_3^2}}\right), \ \ \phi_d \in (-\frac{\pi}{2}, \frac{\pi}{2}) \tag{A.9}$$

$$\theta_d = \arctan\left(\frac{u_1 C_\psi + u_2 S_\psi}{u_3}\right), \ \ \theta_d \in (-\frac{\pi}{2}, \frac{\pi}{2}). \tag{A.10}$$

# Bibliography

[1] F. Blanchini and S. Miani, *Set-theoretic methods in control.* Springer, 2008.

[2] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2016.

[3] M. Jankovic, "Robust control barrier functions for constrained stabilization of nonlinear systems," *Automatica*, vol. 96, pp. 359–367, 2018.

[4] K. P. Tee, S. S. Ge, and E. H. Tay, "Barrier lyapunov functions for the control of output-constrained nonlinear systems," *Automatica*, vol. 45, no. 4, pp. 918–927, 2009.

[5] K. P. Tee and S. S. Ge, "Control of nonlinear systems with full state constraint using a barrier lyapunov function," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, 2009, pp. 8618–8623.

[6] J. Villadsen and M. L. Michelsen, *Solution of differential equation models by polynomial approximation.* Prentice-Hall Englewood Cliffs, NJ, 1978.

[7] A. Zygmund, *Trigonometric series.* Cambridge university press, 2002.

[8] N. K. Bary, *A treatise on trigonometric series.* Elsevier, 2014.

[9] L. Schumaker, *Spline functions: basic theory.* Cambridge University Press, 2007.

[10] S. N. Kumpati and P. Kannan, "Identification and control of dynamical systems using neural networks," *IEEE Transactions on neural networks*, vol. 1, no. 1, pp. 4–27, 1990.

[11] F. L. Lewis, K. Liu, and A. Yesildirek, "Neural net robot controller with guaranteed tracking performance," *IEEE Transactions on Neural Networks*, vol. 6, no. 3, pp. 703–715, 1995.

[12] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction.* MIT press, 2018.

[13] P. Kormushev, S. Calinon, and D. G. Caldwell, "Reinforcement learning in robotics: Applications and real-world challenges," *Robotics*, vol. 2, no. 3, pp. 122–148, 2013.

[14] W. Koch, R. Mancuso, R. West, and A. Bestavros, "Reinforcement learning for uav attitude control," *ACM Transactions on Cyber-Physical Systems*, vol. 3, no. 2, pp. 1–21, 2019.

[15] S. Kuutti, R. Bowden, Y. Jin, P. Barber, and S. Fallah, "A survey of deep learning applications to autonomous vehicle control," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[16] B. Recht, "A tour of reinforcement learning: The view from continuous control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 2, pp. 253–279, 2019.

[17] L. Buşoniu, T. de Bruin, D. Tolić, J. Kober, and I. Palunko, "Reinforcement learning for control: Performance, stability, and deep approximators," *Annual Reviews in Control*, vol. 46, pp. 8–28, 2018.

[18] H. K. Khalil and J. W. Grizzle, *Nonlinear systems*. Prentice hall Upper Saddle River, NJ, 2002.

[19] 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.

[20] R. Kamalapurkar, H. Dinh, S. Bhasin, and W. E. Dixon, "Approximate optimal trajectory tracking for continuous-time nonlinear systems," *Automatica*, vol. 51, pp. 40–48, 2015.

[21] K. G. Vamvoudakis, D. Vrabie, and F. L. Lewis, "Online adaptive algorithm for optimal control with integral reinforcement learning," *International Journal of Robust and Nonlinear Control*, vol. 24, no. 17, pp. 2686–2710, 2014.

[22] B. Kiumarsi, K. G. Vamvoudakis, H. Modares, and F. L. Lewis, "Optimal and autonomous control using reinforcement learning: A survey," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 6, pp. 2042–2062, 2017.

[23] Z. Ni, H. He, and J. Wen, "Adaptive learning in tracking control based on the dual critic network design," *IEEE transactions on neural networks and learning systems*, vol. 24, no. 6, pp. 913–928, 2013.

[24] 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, pp. 82–92, 2013.

[25] Y. Li, Y. Liu, and S. Tong, "Observer-based neuro-adaptive optimized control of strict-feedback nonlinear systems with state constraints," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[26] H. Zhang, L. Cui, X. Zhang, and Y. Luo, "Data-driven robust approximate optimal tracking control for unknown general nonlinear systems using adaptive dynamic programming method," *IEEE Transactions on Neural Networks*, vol. 22, no. 12, pp. 2226–2236, 2011.

[27] J. Boedecker, J. T. Springenberg, J. Wülfing, and M. Riedmiller, "Approximate realtime optimal control based on sparse gaussian process models," in *2014 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, 2014, pp. 1–8.

[28] J. Sun and C. Liu, "Disturbance observer-based robust missile autopilot design with full-state constraints via adaptive dynamic programming," *Journal of the Franklin Institute*, vol. 355, no. 5, pp. 2344–2368, 2018.

[29] W. Huang, G. Li, K.-L. Tan, and J. Feng, "Efficient safe-region construction for moving top-k spatial keyword queries," in *Proceedings of the 21st ACM international conference on Information and knowledge management*, 2012, pp. 932–941.

[30] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin, "A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games," *IEEE Transactions on automatic control*, vol. 50, no. 7, pp. 947–957, 2005.

[31] E. F. Camacho and C. B. Alba, *Model predictive control*. Springer science & business media, 2013.

[32] J. Nocedal and S. Wright, *Numerical optimization*. Springer Science & Business Media, 2006.

[33] S. Singh, B. Landry, A. Majumdar, J.-J. Slotine, and M. Pavone, "Robust feedback motion planning via contraction theory," *The International Journal of Robotics Research*, 2019.

[34] C. Ho, J. Patrikar, R. Bonatti, and S. Scherer, "Adaptive safety margin estimation for safe real-time replanning under time-varying disturbance," *arXiv preprint arXiv:2110.03119*, 2021.

[35] B. Anderson, "Exponential stability of linear equations arising in adaptive identification," *IEEE Transactions on Automatic Control*, vol. 22, no. 1, pp. 83–88, 1977.

[36] S. Boyd and S. S. Sastry, "Necessary and sufficient conditions for parameter convergence in adaptive control," *Automatica*, vol. 22, no. 6, pp. 629–639, 1986.

[37] S. B. Roy and S. Bhasin, "Robustness analysis of initial excitation based adaptive control," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, 2019, pp. 7055–7062.

[38] S. S. Ge, C. C. Hang, T. H. Lee, and T. Zhang, *Stable adaptive neural network control*. Springer Science & Business Media, 2013.

[39] Y. Yang, D.-W. Ding, H. Xiong, Y. Yin, and D. C. Wunsch, "Online barrier-actor-critic learning for h-$\infty$ control with full-state constraints and input saturation," *Journal of the Franklin Institute*, 2019.

[40] M. He, "Data-driven approximated optimal control for chemical processes with state and input constraints," *Complexity*, vol. 2019, 2019.

[41] J. Na, B. Wang, G. Li, S. Zhan, and W. He, "Nonlinear constrained optimal control of wave energy converters with adaptive dynamic programming," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 10, pp. 7904–7915, 2018.

[42] M. Abu-Khalaf, J. Huang, and F. L. Lewis, *Nonlinear H2/H-Infinity Constrained Feedback Control: A Practical Design Approach Using Neural Networks*. Springer Science & Business Media, 2006.

[43] M. Athans and P. L. Falb, *Optimal control: an introduction to the theory and its applications*. Courier Corporation, 2013.

[44] W. B. Powell, *Approximate Dynamic Programming: Solving the curses of dimensionality*. John Wiley & Sons, 2007.

[45] R. Kamalapurkar, J. A. Rosenfeld, and W. E. Dixon, "Efficient model-based reinforcement learning for approximate online optimal control," *Automatica*, vol. 74, pp. 247–258, 2016.

[46] B. A. Finlayson, *The method of weighted residuals and variational principles*. SIAM, 2013.

[47] W. Zhao, H. Liu, and F. L. Lewis, "Robust formation control for cooperative underactuated quadrotors via reinforcement learning," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[48] C. Li, F. Liu, Y. Wang, and M. Buss, "Concurrent learning-based adaptive control of an uncertain robot manipulator with guaranteed safety and performance," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2021.

[49] C. Li, Z. Zhang, A. Nesrin, Q. Liu, F. Liu, and M. Buss, "Instantaneous local control barrier function: An online learning approach for collision avoidance," *arXiv preprint arXiv:2106.05341*, 2021.

[50] C. Li, F. Liu, Z. Zhou, and M. Buss, "Off-policy risk-sensitive reinforcement learning based constrained robust optimal control," *arXiv preprint arXiv:2006.05681*, 2020.

[51] C. Li, Y. Wang, F. Liu, and M. Buss, "Off policy risk sensitive reinforcement learning based optimal tracking control with prescribe performances," *arXiv preprint arXiv:2009.00476*, 2020.

[52] C. Li, Y. Wang, F. Liu, Q. Liu, and M. Buss, "Model-free incremental adaptive dynamic programming based approximate robust optimal regulation," *International Journal of Robust and Nonlinear Control*, 2022.

[53] ——, "Model-free incremental adaptive dynamic programming based approximate robust optimal regulation," *arXiv preprint arXiv:2105.01698*, 2021.

[54] C. Li, Y. Wang, F. Liu, and M. Buss, "Time-delayed data informed reinforcement learning for approximate optimal tracking control," *arXiv preprint arXiv:2110.15237*, 2021.

[55] P. Holmes, S. Kousik, B. Zhang, *et al.*, "Reachable sets for safe, real-time manipulator trajectory design," *arXiv preprint arXiv:2002.01591*, 2020.

[56] M. M. Schill, "Hybrid system stabilization and robot motion planning for robust catching," Ph.D. dissertation, Technische Universität München, 2019.

[57] F. L. Lewis, D. M. Dawson, and C. T. Abdallah, *Robot manipulator control: theory and practice*. CRC Press, 2003.

[58] H. Sadeghian, L. Villani, M. Keshmiri, and B. Siciliano, "Task-space control of robot manipulators with null-space compliance," *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 493–506, 2013.

[59] M. Saveriano and D. Lee, "Learning barrier functions for constrained motion planning with dynamical systems," *arXiv preprint arXiv:2003.11500*, 2020.

[60] C. P. Bechlioulis and G. A. Rovithakis, "Robust adaptive control of feedback linearizable mimo nonlinear systems with prescribed performance," *IEEE Transactions on Automatic Control*, vol. 53, no. 9, pp. 2090–2099, 2008.

[61] D. P. BERTSEKAS, "Approximate dynamic programming," 2012.

[62] I. M. Mitchell, "Comparing forward and backward reachability as tools for safety analysis," in *International Workshop on Hybrid Systems: Computation and Control*, 2007, pp. 428–443.

[63] G. Welch and G. Bishop, *An introduction to the kalman filter*, 1995.

[64] G. Tao, *Adaptive control design and analysis*. John Wiley & Sons, 2003.

[65] G. V. Chowdhary, "Concurrent learning for convergence in adaptive control without persistency of excitation," Ph.D. dissertation, Georgia Institute of Technology, 2010.

[66] A. Parikh, R. Kamalapurkar, and W. E. Dixon, “Integral concurrent learning: Adaptive control with parameter convergence without pe or state derivatives,” *arXiv preprint arXiv:1512.03464*, 2015.

[67] W. He, H. Huang, and S. S. Ge, “Adaptive neural network control of a robotic manipulator with time-varying output constraints,” *IEEE transactions on cybernetics*, vol. 47, no. 10, pp. 3136–3147, 2017.

[68] S. Kousik, P. Holmes, and R. Vasudevan, “Safe, aggressive quadrotor flight via reachability-based trajectory design,” in *Dynamic Systems and Control Conference*, vol. 59162, 2019, V003T19A010.

[69] S. Kousik, S. Vaskov, F. Bu, M. Johnson-Roberson, and R. Vasudevan, “Bridging the gap between safety and real-time performance in receding-horizon trajectory design for mobile robots,” *The International Journal of Robotics Research*, vol. 39, no. 12, pp. 1419–1469, 2020.

[70] S. Liu, M. Watterson, K. Mohta, *et al.*, “Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-d complex environments,” *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1688–1695, 2017.

[71] A. Majumdar and R. Tedrake, “Funnel libraries for real-time robust feedback motion planning,” *The International Journal of Robotics Research*, vol. 36, no. 8, pp. 947–982, 2017.

[72] E. D. Sontag and Y. Wang, “On characterizations of the input-to-state stability property,” *Systems & Control Letters*, vol. 24, no. 5, pp. 351–359, 1995.

[73] E. D. Sontag, “Smooth stabilization implies coprime factorization,” *IEEE transactions on automatic control*, vol. 34, no. 4, pp. 435–443, 1989.

[74] X. Jin, “Adaptive fixed-time control for mimo nonlinear systems with asymmetric output constraints using universal barrier functions,” *IEEE Transactions on Automatic Control*, vol. 64, no. 7, pp. 3046–3053, 2018.

[75] S. M. LaValle, *Planning algorithms.* Cambridge university press, 2006.

[76] P. Zhao, A. Lakshmanan, K. Ackerman, A. Gahlawat, M. Pavone, and N. Hovakimyan, “Tube-certified trajectory tracking for nonlinear systems with robust control contraction metrics,” *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5528–5535, 2022.

[77] T. Luukkonen, “Modelling and control of quadcopter,” *Independent research project in applied mathematics, Espoo*, vol. 22, p. 22, 2011.

[78] H. Ji, W. Shang, and S. Cong, “Adaptive synchronization control of cable-driven parallel robots with uncertain kinematics and dynamics,” *IEEE Transactions on Industrial Electronics*, vol. 68, no. 9, pp. 8444–8454, 2020.

[79] F.-Y. Wang, H. Zhang, and D. Liu, “Adaptive dynamic programming: An introduction,” *IEEE computational intelligence magazine*, vol. 4, no. 2, pp. 39–47, 2009.

[80] D. Enns, D. Bugajski, R. Hendrick, and G. Stein, “Dynamic inversion: An evolving methodology for flight control design,” *International Journal of control*, vol. 59, no. 1, pp. 71–91, 1994.

[81] X. Xu, “Constrained control of input–output linearizable systems using control sharing barrier functions,” *Automatica*, vol. 87, pp. 195–201, 2018.

[82] A. D. Ames, K. Galloway, K. Sreenath, and J. W. Grizzle, “Rapidly exponentially stabilizing control lyapunov functions and hybrid zero dynamics,” *IEEE Transactions on Automatic Control*, vol. 59, no. 4, pp. 876–891, 2014.

[83] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise,” in *Kdd*, vol. 96, 1996, pp. 226–231.

[84] J. O. Rawlings, S. G. Pantula, and D. A. Dickey, *Applied regression analysis: a research tool*. Springer Science & Business Media, 2001.

[85] P. W. Holland and R. E. Welsch, “Robust regression using iteratively reweighted least-squares,” *Communications in Statistics-theory and Methods*, vol. 6, no. 9, pp. 813–827, 1977.

[86] W. Xiao and C. Belta, “High order control barrier functions,” *IEEE Transactions on Automatic Control*, 2021.

[87] W. Xiao, G. C. Cassandras, and C. Belta, “Safety-critical optimal control for autonomous systems,” *Journal of Systems Science and Complexity*, vol. 34, no. 5, pp. 1723–1742, 2021.

[88] S. Castro, *Mobile robotics simulation toolbox*, MathWorks, 2019. [Online]. Available: `https://github.com/mathworks-robotics/mobile-robotics-simulation-toolbox`.

[89] T. Coleman, M. A. Branch, and A. Grace, “Optimization toolbox,” *For Use with MATLAB. User’s Guide for MATLAB 5, Version 2, Relaese II*, 1999.

[90] M. Abu-Khalaf and F. L. Lewis, “Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network hjb approach,” *Automatica*, vol. 41, no. 5, pp. 779–791, 2005.

[91] F. Lin, R. D. Brandt, and J. Sun, “Robust control of nonlinear systems: Compensating for uncertainty,” *International Journal of Control*, vol. 56, no. 6, pp. 1453–1459, 1992.

[92] D. Liu, S. Xue, B. Zhao, B. Luo, and Q. Wei, “Adaptive dynamic programming for control: A survey and recent advances,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2020.

[93] F. Lin and R. D. Brandt, “An optimal control approach to robust control of robot manipulators,” *IEEE Transactions on robotics and automation*, vol. 14, no. 1, pp. 69–77, 1998.

[94] Y. Shen, W. Stannat, and K. Obermayer, “Risk-sensitive markov control processes,” *SIAM Journal on Control and Optimization*, vol. 51, no. 5, pp. 3652–3672, 2013.

[95] A. Heydari and S. N. Balakrishnan, “Finite-horizon control-constrained nonlinear optimal control using single network adaptive critics,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 1, pp. 145–157, 2012.

[96] 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.

[97] K. Zhang, R. Su, H. Zhang, and Y. Tian, "Adaptive resilient event-triggered control design of autonomous vehicles with an iterative single critic learning framework," *IEEE transactions on neural networks and learning systems*, 2021.

[98] L. Wang, E. A. Theodorou, and M. Egerstedt, "Safe learning of quadrotor dynamics using barrier certificates," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 2460–2465.

[99] R. Courant and D. Hilbert, *Methods of Mathematical Physics.* Vol I. Wiley (Interscience), New York, 1953.

[100] X. Yang, D. Liu, H. Ma, and Y. Xu, "Online approximate solution of hji equation for unknown constrained-input nonlinear continuous-time systems," *Information Sciences*, vol. 328, pp. 435–454, 2016.

[101] R. Kamalapurkar, P. Walters, and W. E. Dixon, "Model-based reinforcement learning for approximate optimal regulation," in *Control of Complex Systems*, Elsevier, 2016, pp. 247–273.

[102] L. R. G. Carrillo and K. G. Vamvoudakis, "Deep-learning tracking for autonomous flying systems under adversarial inputs," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 2, pp. 1444–1459, 2019.

[103] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," *arXiv preprint arXiv:1511.05952*, 2015.

[104] S. J. Leon, I. Bica, and T. Hohn, *Linear algebra with applications.* Prentice Hall Upper Saddle River, NJ, 1998.

[105] W. Fedus, P. Ramachandran, R. Agarwal, *et al.*, "Revisiting fundamentals of experience replay," *arXiv preprint arXiv:2007.06700*, 2020.

[106] X. Yang and H. He, "Event-triggered robust stabilization of nonlinear input-constrained systems using single network adaptive critic designs," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2018.

[107] V. Nevistić and J. A. Primbs, "Constrained nonlinear optimal control: A converse hjb approach," 1996.

[108] T. C. Hsia and L. Gao, "Robot manipulator control using decentralized linear time-invariant time-delayed joint controllers," in *IEEE International Conference on Robotics and Automation*, 1990, pp. 2070–2075.

[109] K. Youcef-Toumi and S.-T. Wu, "Input/output linearization using time delay control," *Journal of dynamic systems, measurement, and control*, vol. 114, no. 1, pp. 10–19, 1992.

[110] S. Tong, K. Sun, and S. Sui, "Observer-based adaptive fuzzy decentralized optimal control design for strict-feedback nonlinear large-scale systems," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 2, pp. 569–584, 2017.

[111] B. Kiumarsi and F. L. Lewis, "Actor–critic-based optimal tracking for partially unknown nonlinear discrete-time systems," *IEEE transactions on neural networks and learning systems*, vol. 26, no. 1, pp. 140–151, 2014.

[112] S. Formentin, S. Garatti, G. Rallo, and S. M. Savaresi, “Robust direct data-driven controller tuning with an application to vehicle stability control,” *International Journal of Robust and Nonlinear Control*, vol. 28, no. 12, pp. 3752–3765, 2018.

[113] S. Sastry, *Nonlinear systems: analysis, stability, and control.* Springer Science & Business Media, 2013.

[114] M. Jin, J. Lee, P. H. Chang, and C. Choi, “Practical nonsingular terminal sliding-mode control of robot manipulators for high-accuracy tracking control,” *IEEE Transactions on Industrial Electronics*, vol. 56, no. 9, pp. 3593–3601, 2009.

[115] G. F. Franklin, J. D. Powell, and M. L. Workman, *Digital control of dynamic systems.* Addison-wesley Reading, MA, 1998.

[116] S. Bhasin, R. Kamalapurkar, H. T. Dinh, and W. E. Dixon, “Robust identification-based state derivative estimation for nonlinear systems,” *IEEE Transactions on Automatic Control*, vol. 58, no. 1, pp. 187–192, 2012.

[117] A. Levant, “Robust exact differentiation via sliding mode technique,” *Automatica*, vol. 34, no. 3, pp. 379–384, 1998.

[118] P. H. Chang and J. H. Jung, “A systematic method for gain selection of robust pid control for nonlinear plants of second-order controller canonical form,” *IEEE Transactions on Control Systems Technology*, vol. 17, no. 2, pp. 473–483, 2008.

[119] W. Wang and Z. Gao, “A comparison study of advanced state observer design techniques,” in *Proceedings of the 2003 American Control Conference, 2003*, vol. 6, 2003, pp. 4754–4759.

[120] Y. Zhou, E.-J. v. Kampen, and Q. Chu, “Nonlinear adaptive flight control using incremental approximate dynamic programming and output feedback,” *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 2, pp. 493–496, 2016.

[121] ——, “Incremental model based online dual heuristic programming for nonlinear adaptive control,” *Control Engineering Practice*, vol. 73, pp. 13–25, 2018.

[122] Y. Zhou, E.-J. Van Kampen, and Q. Chu, “Incremental model based online heuristic dynamic programming for nonlinear adaptive tracking control with partial observability,” *Aerospace Science and Technology*, vol. 105, p. 106 013, 2020.

[123] P. Acquatella, E. van Kampen, and Q. P. Chu, “Incremental backstepping for robust nonlinear flight control,” *Proceedings of the EuroGNC*, vol. 2013, 2013.

[124] P. Simplicio, M. Pavel, E. Van Kampen, and Q. Chu, “An acceleration measurements-based approach for helicopter nonlinear flight control using incremental nonlinear dynamic inversion,” *Control Engineering Practice*, vol. 21, no. 8, pp. 1065–1077, 2013.

[125] W.-H. Chen, J. Yang, L. Guo, and S. Li, “Disturbance-observer-based control and related methods—an overview,” *IEEE Transactions on Industrial Electronics*, vol. 63, no. 2, pp. 1083–1095, 2015.

[126] Y. Shtessel, C. Edwards, L. Fridman, and A. Levant, *Sliding mode control and observation.* Springer, 2014.

[127] H. Flanders, “Differentiation under the integral sign,” *The American Mathematical Monthly*, vol. 80, no. 6, pp. 615–627, 1973.

[128] W. C. Stirling, *Satisficing Games and Decision Making: with applications to engineering and computer science*. Cambridge University Press, 2003.

[129] D. Liu and Q. Wei, “Policy iteration adaptive dynamic programming algorithm for discrete-time nonlinear systems,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 3, pp. 621–634, 2013.

[130] J. Si and Y.-T. Wang, “Online learning control by association and reinforcement,” *IEEE Transactions on Neural networks*, vol. 12, no. 2, pp. 264–276, 2001.

[131] K. G. Vamvoudakis and F. L. Lewis, “Online solution of nonlinear two-player zero-sum games using synchronous policy iteration,” *International Journal of Robust and Nonlinear Control*, vol. 22, no. 13, pp. 1460–1483, 2012.

[132] D. Liu, X. Yang, D. Wang, and Q. Wei, “Reinforcement-learning-based robust controller design for continuous-time uncertain nonlinear systems subject to input constraints,” *IEEE transactions on cybernetics*, vol. 45, no. 7, pp. 1372–1385, 2015.

[133] T. S. Hsia, “A new technique for robust control of servo systems,” *IEEE Transactions on Industrial Electronics*, vol. 36, no. 1, pp. 1–7, 1989.

[134] F. Lewis, S. Jagannathan, and A. Yesildirak, *Neural network control of robot manipulators and non-linear systems*. CRC press, 2020.

[135] R. Kamalapurkar, L. Andrews, P. Walters, and W. E. Dixon, “Model-based reinforcement learning for infinite-horizon approximate optimal tracking,” *IEEE transactions on neural networks and learning systems*, vol. 28, no. 3, pp. 753–758, 2016.

[136] Y. Liu, W. Sun, and H. Gao, “High precision robust control for periodic tasks of linear motor via b-spline wavelet neural network observer,” *IEEE Transactions on Industrial Electronics*, 2021.

[137] Y. Li, K. Sun, and S. Tong, “Observer-based adaptive fuzzy fault-tolerant optimal control for siso nonlinear systems,” *IEEE transactions on cybernetics*, vol. 49, no. 2, pp. 649–661, 2018.

[138] T. Beckers, J. Umlauft, D. Kulic, and S. Hirche, “Stable gaussian process based tracking control of lagrangian systems,” in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, 2017, pp. 5180–5185.

[139] J. Na, G. Herrmann, and K. G. Vamvoudakis, “Adaptive optimal observer design via approximate dynamic programming,” in *2017 American Control Conference (ACC)*, 2017, pp. 3288–3293.

[140] B. Zhao and D. Liu, “Event-triggered decentralized tracking control of modular reconfigurable robots through adaptive dynamic programming,” *IEEE Transactions on Industrial Electronics*, vol. 67, no. 4, pp. 3054–3064, 2019.

[141] F. Luo, B. Zhao, and D. Liu, “Event-triggered decentralized optimal fault tolerant control for mismatched interconnected nonlinear systems through adaptive dynamic programming,” *Optimal Control Applications and Methods*, 2021.

[142] R. Kamalapurkar, B. Reish, G. Chowdhary, and W. E. Dixon, “Concurrent learning for parameter estimation using dynamic state-derivative estimators,” *IEEE Transactions on Automatic Control*, vol. 62, no. 7, pp. 3594–3601, 2017.

[143] H. Modares and F. L. Lewis, “Optimal tracking control of nonlinear partially-unknown constrained-input systems using integral reinforcement learning,” *Automatica*, vol. 50, no. 7, pp. 1780–1792, 2014.

[144] P. Corke, “Robotics toolbox,” *Obtained from Peter O. Corke site: http://www. peter-corke. com/Robotics% 20Toolbox. html*, 2002.

[145] M. Greiff, “Modelling and control of the crazyflie quadrotor for aggressive and autonomous flight by optical flow driven state estimation,” 2017.