DATA-BASED REINFORCEMENT LEARNING: APPROXIMATE OPTIMAL CONTROL FOR UNCERTAIN NONLINEAR SYSTEMS

By

PATRYK DEPTUŁA

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2019

To my parents Bożena and Tadeusz Deptuła, and to my sister Aleksandra Deptuła, who have provided invaluable encouragement, support, and love throughout my journey.

ACKNOWLEDGMENTS

I would like to thank Dr. Warren E. Dixon, who has provided me with continuous support, guidance, and patience throughout this journey. His motivation and drive encouraged me to work harder and accomplish more than I have before, thus allowing me to grow on both an individual and intellectual level. I would like to extend my gratitude to my committee members Dr. Carl Crane, Dr. Matthew Hale, and Dr. Jacob Hammer for their oversight and recommendations throughout this journey. I would also like to thank my prior professors from my undergraduate university who motivated me to pursue a Doctorate of Philosophy. Last but not least, thank you to my family and friends who constantly support me and encourage me not to limit myself.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

DATA-BASED REINFORCEMENT LEARNING: APPROXIMATE OPTIMAL CONTROL FOR UNCERTAIN NONLINEAR SYSTEMS

By

Patryk Deptuła

May 2019

Chair: Warren E. Dixon
Major: Mechanical Engineering

The last two decades have witnessed an influx of autonomous systems including: unmanned aerial vehicles (UAVs), autonomous underwater vehicles (AUVs), and autonomous land vehicles. Since such systems are subject to physical, energetic, or mission constraints, there is motivation for optimality. However, ensuring credibility or reliability in autonomous systems is challenging because often times the systems and the environment they operate in are prone to uncertainties. Even in the absence of dynamic or environmental uncertainties, an analytical optimal policy can not be determined.

Reinforcement learning (RL) has become a popular tool for investigating optimal control problems because it enables a cognitive agent to learn a desirable behavior based on interactions with its environment. Recent advances in approximate dynamic programming (ADP) provide a means to use RL to generate forward-in-time (online) controllers for systems. Specifically, advances in model-based reinforcement learning (MBRL) such as regional MBRL (R-MBRL) and state-following (StaF) kernel based techniques have enabled ADP to be implemented on hardware. Unlike traditional methods such as R-MBRL that aim to approximate the value function over a large compact set, the StaF kernel approach aims to approximate the value function in a local neighborhood of the state that travels within a compact set and is computationally efficient compared to the R-MBRL method. This dissertation investigates such computational

issues within a RL-ADP framework as a means to learn the infinite-horizon approximate optimal controllers and value functions (i.e. minimized cost function under optimal controller) for systems with various constraints.

In Chapter 3, a regulation problem for a control-affine system is solved online using the StaF kernel and a R-MBRL method to approximate the value function. While the StaF method used is computationally efficient, it loses the information about the value function in a region once the system state leaves that region. However, R-MBRL is able to approximate the value function over a predefined region irrespective of the current state. In this chapter, the value function is approximated using a state-dependent convex combination of the StaF-based and the R-MBRL-based approximations. When the state enters a neighborhood containing the origin, the value function transitions from being approximated by the StaF approach to the R-MBRL approach.

Motivated by the local approximation nature of the StaF method, a regulation problem is considered in Chapter 4 for a control-affine nonlinear autonomous agent in the presence of dynamic avoidance regions. The StaF-based ADP method is implemented to approximate the value function in a local neighborhood of the agent. By performing local approximations, prior knowledge of avoidance region locations is not required. Because having knowledge about the number of avoidance regions or their dynamics is limiting, an extension is provided to alleviate this assumption.

Unlike the single agent path-planning problem, where policies are developed to drive a single agent to a goal location, there may be instances where the goal is to direct multiple agents to a goal. In Chapter 5, an indirect regulation problem is considered for two agents, an influencing agent and a roaming agent. The roaming agent is not motivated to go a desired location; hence, the influencing agent is used to guide the roaming agent to the goal through inter-agent interaction. An indirect approach is used where a virtual state and input are designed that aim to regulate the roaming agent. The influencing agent tracks the virtual signals, and thus intercepts and regulates

the roaming agent. A data-based system parameter estimation technique is used to learn both agent dynamics and the computationally efficient StaF method is used to approximate the optimal policy and value function online.

Using a game-based formulation, Chapter 6 considers the indirect regulation problem for two agents. Compared to the indirect virtual signal development in Chapter 5, two error systems are developed. The first signal is developed such that the influencing agent intercepts the roaming agent, the second signal is developed such that the influencing agent regulates the roaming to the goal location. The problem is formulated as a minimax differential game where the influencing agent's policy is the minimizer while the roaming agents worst-case policy is the maximizer. The StaF approximation method is used in an actor-critic-disturber approach to estimate the optimal influencing agent policy, the worst-case roaming agent disturbing policy, and value function.

A Lyapunov-based stability analysis is used in this dissertation to show convergence of the closed-loop systems and weight estimation errors. In addition, the performance of the developed strategies are shown through simulation and experimental validation.

CHAPTER 1
INTRODUCTION

1.1 Background

As we seek to engineer more intelligent systems, opportunities to optimize them in real-time, while ensuring stability and desirable performance in the presence of uncertainty arise. Optimal control is a method which associates a cost with control actions and has been applied to a wide range of applications including: engineering systems, financial markets, and medical technologies. Being able to learn the optimal, or most efficient, control policies that minimize the cost of a control action involves the fundamental challenge of exploration versus exploitation. In autonomous systems, optimal behavior is necessary for efficient task execution. Moreover, many challenges exist for real-time navigation in uncertain environments. To operate safely in an uncertain environment, an autonomous agent must be able to identify and react to possible collisions and other uncertainties. In practice, challenges result from limitations in computational resources, sensing, communication, and mobility.

Path-planning strategies can be divided into global and local planners [2]. Global planners seek the best trajectory by using models of the entire environment and are computed before a mission begins (cf. [3–6]). Local or reactive methods plan only a few time steps forward based on limited knowledge using sensory data and have the advantage of providing optimal feedback if the agent is forced off of its original path. When developing optimal control policies for autonomous agents, it is necessary to consider the agent's dynamics and the operating environment. Such operating conditions present significant guidance and control challenges since agents may be required to avoid or track strict avoidance regions, which may not be initially known. Due to the dynamic nature of the environment, online feedback-based control/guidance algorithms with online learning and adaptation capabilities are essential for re-planning and execution.

Constrained optimization methods can be used to generate guidance/control laws for agents operating in complex environments; however, agents often exhibit nonlinear dynamics and navigate in environments with uncertain dynamics or constraints, which makes the determination of analytical solutions to constrained optimization problems difficult. Traditional guidance/control solutions exploit numerical methods [3, 7–10]. In results such as [7, 9], the Hamilton-Jacobi-Bellman (HJB) equation is solved offline to generate a feedback strategy, while in [8] the HJB is approximated across a discretized state-space by way of offline Dynamic Programming (DP) with online interpolation. In [10], viscosity solutions for the Hamilton-Jacobi-Isaacs (HJI) are numerically computed offline to generate safe feedback controllers. In [3], a pseudospectral method is developed where the cost function and dynamics are discretized and solved across each phase of the control problem. Afterwards, the phases are connected using boundary conditions of each phase. However, in numerical nonlinear optimization problems difficulties arise as the dimension of the system increases due to the associated computational complexity. Furthermore, in many instances numerical methods do not consider uncertainty in the dynamics or environment, and are ill-suited for dynamically changing environments as new guidance/control solutions would need to be recalculated offline in the event of a change in the environment. This motivates the use of approximate optimal control methods that use parametric function approximation techniques capable of approximating the solution to the HJB online (cf. [1, 11–21]).

Further complicating the task of optimal path-planning are agent actuator constraints and state constraints (e.g., static or mobile avoidance regions) often present en route to an objective. Certain avoidance regions may be undiscovered until they fall into a given detection range. The concept of avoidance control was introduced in [22] for two-player pursuit-evasion games. In [23, 24] navigation functions where constructed to unify path planning with lower level feedback control. Results such as [25–27] utilize navigation functions for multi-agent systems for collision avoidance and network

connectivity. The results in [28–30] consider collision avoidance in multi-agent systems with limited sensing by using bounded avoidance functions in the controller which are only active when agents are within a defined sensing radius. Nonetheless, the results in [25–30] do not consider optimal controllers, and in [25–27] control constraints are not considered. In [31], unbounded avoidance functions are incorporated with explicitly computed optimal controllers for cooperative avoidance for multi-agent systems. A computationally efficient parametric method was utilized in [32] to develop an approximate optimal online path planner with static obstacle avoidance which are not known a priori until sensed. However, the development in [32] used a transitioning controller which switched between the approximate controller and a robust controller when the obstacles where sensed. In [33–35], sets of feasible states along with safe controllers are computed using the results in [10] by developing a differential game between two players. Despite such advances, the results in [31] rely on explicitly computed controllers, which are unknown when the optimal value function is unknown, while the results in [10,33–35] rely on numerical techniques, which tend to be computationally intensive.

While path-planning problems for single agent systems focus on developing strategies to guide an agent to some goal, multi-agent regulation problems consist of active agents steering other agents (roaming, target, or follower agents) to a desired location. Specifically, in such problems, multi-agent systems may be cooperative, where the agents aim to reach the same goal, or non-cooperative, where the agents aim to reach different goals such as pursuit-evasion or reach-avoid games. Game theory is concerned with the analysis of strategies players can take based on particular conditions [36]. Pursuit-evasion games are problems motivated by predator-prey scenarios, where strategies for either evading or pursing agents are calculated using differential game theory (cf. [37–49] and references therein). Several different approaches have been considered in pursuit-evasion games. For instance, works such as [44] consider multi-player pursuit-evasion capture conditions, and cooperative control strategies are

calculated for pursuing agents to capture evading agents. Results such as [45] develop escape strategies for evading agents using mathematical frameworks based on Apollonius circles. Results such as [46] consider pursing agents with uncertain speeds and calculate the strategies for the pursuing agent, while escape strategies are selected for evading agents based on how pursuing agents are approaching them. Results such as [43] and [50] consider reach-avoid games to determine strategies (i.e., winning and losing regions) computed numerically using level set methods to solve the Hamilton Jacobi Isaacs (HJI) equation. While the aforementioned and related literature provide foundational strategies for pursuit-evasion games, most results assume simple or known dynamics and generally only consider the problem in two-dimensions. In addition, assumptions about the knowledge of the opposing players strategies are required to gain an advantage. Moreover, such games are only focused at finding strategies for either capturing or evading aspects of the game.

While traditional pursuit-evasion problems focus on either the trapping or fleeing aspects of the game, a different class of problems, called herding, focus on directing uncontrolled agents to a goal location and have been investigated in results such as [42, 51–57]. Unlike pursuit-evasion problems, in indirect herding problems, the influencing agent must pursue a roaming agent while also escorting it to a desired location through an inter-agent interaction.[1] The results in [55] and [54] approach the indirect regulation (also known as indirect herding) problem via a switched-systems approach where the influencing agent switches between target agents. In [54], a robust sliding mode approach is used to compensate for worst case uncertainties of the target agent dynamics. Compared to [54], the result in [55] uses an adaptive control approach

[1] The agent is called roaming instead of evading since it may not pursue an optimal strategy or necessarily seek to escape from the pursing agent. However, the result is still valid if the roaming agent pursues strategies that may be optimal in the sense of the game formulation.

where a data-based parameter estimation method called integral concurrent learning (ICL) is used to learn the linearly parametrized (LP) target dynamics by storing input-output data (cf. [58]). In [56] and [57], a forcing function based on geometric constraints is used to develop a controller for a group of agents that regulate other agents indirectly by forming an arc and forcing the targets to the desired location. Although major advancements have been made in multi-player problems, results such as [44–46] generally consider point mass systems and know the form of the agent dynamics, while results such as [54–57] rely on explicitly designed controllers for the influencing agent based on the target dynamics and do not consider optimality.

Herding-based problems using DP to find optimal strategies for pursuing agents to capture and regulate evading agents to goal locations have been investigated. Results such as [42, 52, 53] compute optimal policies for pursuing agents regulating multiple evading agents with known point mass dynamics. Specifically, the result in [52] uses the Sparse Nonlinear Optimizer (SNOPT) algorithm in [59, 60] to compute numerical solutions offline. The works in [42, 53] use DP and shortest algorithms over a finite graph to determine offline optimal policies that a pursing agent can take to drive an evading agent to a goal location. Although results such as [42, 52, 53] provide in-roads for optimal herding, the computational complexity associated with a large number of states renders the problems infeasible for online implementation, the agents are assumed to take simple one-step discrete actions over a finite grid, and generally the dynamics of the agents are known. Such results require numerical solutions, which can be computationally expensive for high dimensional systems, and do not consider system uncertainties; hence, the use of parametric methods, such as neural-networks (NNs), to yield computationally efficient approximate optimal controllers online is motivated.

Designing optimal controllers for uncertain nonlinear systems is difficult because the solution to the HJB (or HJI) is generally unknown. Approximate dynamic programming is a popular method which has been successfully used in deterministic autonomous control-affine systems to develop approximately optimal solutions (cf. [11, 61–63]). RL-based ADP is based on the strategy of approximating the solution to the HJB via NN representations (cf., [11–14, 17, 18, 20, 21, 64–69]). Analytical representations of the optimal controllers can be derived by incorporating the control in the cost function. For systems without input constraints a quadratic penalty on the control input is used, while for systems with input constraints, a non-quadratic cost function can be used to yield a bounded approximate optimal controller (cf., [15–17]). Moreover, utilizing parametric approximation methods, ADP approximates the value function, which is the solution to the HJB, and is used to compute the online forward-in-time optimal policy; however, ADP has two inherent significant challenges.

One challenge for dynamic programming methods is the curse of dimensionality because a large number (i.e., exponential growth with the number of states) of basis functions is generally required to obtain a sufficient approximation over a large region. For general nonlinear systems, generic basis functions, such as Gaussian radial basis functions (RBF), polynomials, or universal kernel functions are used to approximate the value function. One limitation of these generic approximation methods is that they only ensure an approximation over a compact neighborhood of the origin. Once outside the compact set, the approximation tends to either grow or decay depending on the selected functions. Consequently, in the absence of domain knowledge, a large number of basis functions, and hence, a large number of unknown parameters, is required for value function approximation. A recent advancement in ADP utilizes computationally efficient StaF kernel basis functions for local approximation of the value function around the current state, thereby reducing the number of basis functions required for sufficient value function approximation [1, 32, 70]. Unlike traditional ADP approximation methods

which use a large number of basis functions, the recent StaF method performs local approximations of the value function with a reduced number (i.e., linear growth) of basis functions. Although the StaF approximation method is efficient, it trades global optimality for computational efficiency.

Another challenge for ADP methods is that, unlike traditional adaptive controllers, the ideal weights of the NN must be exactly learned (i.e., system identification) to approximate the optimal controller. The rate at which the value function approximation error decays, which results when the approximate NN weights converge to the ideal weights, is determined by the richness of the data used for learning. In traditional adaptive control and ADP methods such as [64] and [71] richness of the data correlates to the amount of excitation in the system, resulting in the so-called persistence of excitation (PE) condition. Typically excitation is introduced by adding an exploration signal to the control input (cf., [65, 66, 72–75]). However, for general nonlinear systems, there is no currently known way to ensure the PE condition is satisfied a priori, even with the added probing noise, and no way to verify if the condition is satisfied online. Moreover, because the addition of the exploration signal causes undesirable oscillations and noise, hardware implementation of traditional ADP techniques is challenging. Motivated by this issue, data-driven techniques such as simulation of experience and experience replay aim to relax the PE assumption by utilizing concurrent learning (CL), where data richness is characterized by the eigenvalues of a history stack, which unlike the PE condition can be verified online (cf., [1, 15, 17, 18, 21, 65, 75–80]). Specifically, CL collects pairs of input/output data and stores them in an evolving history stack during task execution. The input/output pairs can then be used, along with methods such as [77, 81] to manage the size and composition of the history stack, to perform system identification assuming the derivative of the highest order states is available or numerically generated. ICL removes the need to measure the derivative of the highest order terms by including an integral of the terms in the history stack. Specifically,

in [82, 83], ICL is formulated so that the dynamics are integrated over a finite window; hence, the formulation includes both a finite difference and an integrated function. However, such an approach requires numerical techniques to evaluate the integrals, which can cause errors to accumulate if large integration buffers are used or in the presence of measurement noise. Another approach which uses an initial excitation (IE) condition that can be verified online for guaranteed parameter identification via an integral-like update law is presented in [84]. Unlike CL and ICL, the integral-like method in [84] uses a continuous input-output signal and does not rely on a history stack of sampled data to ensure sufficient learning.

Despite the challenges associated with ADP, numerous results have been developed using differential game formulations using ADP (cf. [12, 62, 85–89]). However, all of these results solve the multi-player problem by generating controllers and directly controlling each agent. Exceptions include the innovative work in [90] and [91]. Specifically, an ADP-based backstepping approach is developed in [90] and [91] for a class of known strict-feedback nonlinear systems containing a one-dimensional input. In [91], for each individual step of the backstepping approach, a virtual control is obtained using the Sontag formula [92] which is equivalent to the optimal control. While in [90], a quadratic term is injected into the optimal value function of each backstepping instance, and the mismatch between the quadratic term and unknown optimal value function is approximated using NNs. Despite such progress, results such as [90] and [91] both assume exact model knowledge of the agent dynamics and require the strict PE condition to be satisfied.

1.2 Outline of the dissertation

In Chapter 2, the infinite-horizon optimal control problem is introduced and the approach to approximating the value function is discussed. In addition, the data-based system identification method ICL is introduced and briefly discussed.

In Chapter 3, a novel framework is developed to merge local and regional value function approximation methods to yield an online optimal control method that is computationally efficient and simultaneously accurate over a specified critical region of the state-space. This chapter is motivated by the ability of R-MBRL such as [18] to approximate the value function over a predefined region and the computational efficiency of the StaF method [1] in approximating the value function locally along the state trajectory. Instead of generating an approximation of the value function over the entire operating region, which would be computationally expensive, the operating domain is separated into two regions: a closed set A, containing the origin, where a regional approximation method is used to approximate the value function, and the complement of A, where the StaF method is used to approximate the value function. Using a switching based approach to combine regional and local approximations would inject discontinuities to the system and result in a non-smooth value function which would introduce discontinuities in the control signal. To overcome this challenge, a state varying convex combination of the two approximation methods is used to ensure a smooth transition from the StaF to the R-MBRL approximation as the state enters the closed convex set containing the origin. Once the state enters this region, R-MBRL regulates the state to the origin. The developed result is generalized to allow for the use of any R-MBRL method. A Lyapunov-based stability analysis is performed to provide insights into how the estimates should be designed to combine StaF and R-MBRL while also preserving stability. The analysis of the closed-loop systems with the smoothly switching approximation guarantees uniformly ultimately bounded (UUB) convergence. Numerical simulations are performed to demonstrate the performance of the developed method.

In Chapter 4, an approximate optimal feedback-based motion planner is developed that considers input and state constraints with mobile avoidance regions. The developed method differs from results such as [32] and other path planners in that it tackles the

challenge of avoiding dynamic avoidance regions within the path-planning strategy without switching between controllers. Since the StaF method uses local approximations, it does not require knowledge of uncertainties in the state-space outside an approximation window. Local approximations of the StaF kernel method enable it to handle certain situations such as approaching avoidance regions, not known a priori, in addition to state and system constraints. Because the avoidance regions become coupled with the agent in the HJB, their states need to be considered when approximating the value function. Hence, a basis is given for each region which is zero outside of the sensing radius but is active when the avoidance region is sensed. In some applications, knowing the weights for an avoidance region may provide useful information, as the approximation of the value function can be improved every time the region is encountered. To prevent collision, a penalizing term is added to the cost function which guarantees that the agent stays outside of the avoidance regions. A Lyapunov-based stability analysis is presented and guarantees UUB convergence while also ensuring that the agent remains outside of the avoidance regions. Since knowledge about the number of avoidance regions and their dynamics is limiting. An extension is provided to alleviate this Assumption. Experimental results are provided to illustrate the performance of the developed method.

Chapter 5 investigates an approximately optimal indirect regulation problem for two agents. Unlike typical pursuit-evasion problems or the aforementioned ADP results, the influencing agent (often referred to as the pursuer) aims to intercept and regulate a roaming agent to a goal location. Since there is no direct input for the roaming agent, the goal is for the influencing agent to direct the roaming agent by the use of an uncertain interaction dynamic. Moreover, the influencing agent state may be non-affine in the roaming agent dynamics; hence, a virtual state is introduced, whose time-derivative is the virtual input. The virtual state aims to approximately optimally regulate the roaming to a goal location. The influencing agent's input is then designed to approximately optimally minimize the mismatch between the influencing agent's

actual versus virtual state. Specifically, the contribution of this result is to approximately optimally regulate an agent to a goal location through an uncertain interaction with a controlled pursuing agent. The approximate optimal pursuer does not require exact model knowledge of either the agent dynamic or the interaction dynamic, and does not assume a policy for the roaming agent.

The developed approach is model-based and an actor-critic-identifier [61] strategy is employed, where the adaptive estimates must converge to the actual parameters to yield the optimal policy. To alleviate the need for physical excitation of the system to satisfy the PE condition, the work in this chapter uses ICL to identify both the pursuing and roaming agent uncertainties. Unlike results such as [54] and [55], the drift dynamics of both agents are assumed to be unknown, and an approximately optimal control strategy is developed. Compared to results such as [44–46], the agent dynamics in this chapter are considered to be nonlinear and uncertain and the developed technique does not rely on numerical methods. Compared to results such as [42, 52, 53], the policy for the influencing agent is calculated online and does not use one-step discrete actions. Moreover, while the results in [42, 52, 53] tend to be computationally inefficient due to the curse of dimensionality associated with DP, the strategy in this work uses the computationally efficient StaF function approximation approach in a continuous space problem formulation. Furthermore, unlike the preliminary result in [93], this work includes redefined error signals. Moreover, a Lyapunov-based stability analysis is included which ensures that the closed-loop system is UUB. A simulation and an experimental study are included to demonstrate the performance of the developed strategy.

In Chapter 6, an adaptive approximately optimal indirect regulation approach is investigated for a single influencing agent that pursues and steers a single roaming agent to a goal location (unknown to the roaming agent) using principles from differential game theory. The approach in Chapter 5 does not consider additional inputs, such

as exogenous disturbances, on the roaming agent. In this chapter, the roaming agent dynamics are considered to be modeled as a combination of uncertain drift dynamics and a worst-case disturbing input which aims to maximize the prescribed cost functional. Two error systems are developed to facilitate the pursuit and regulation objectives. The problem is formulated as an infinite-horizon minimax problem and the solution to HJI equation is approximated using the computationally efficient StaF method. Compared to Chapter 5, streamlined error systems and redeveloped control development eliminate the need to consider a virtual state. In addition, system uncertainties not captured during system identification, can be included as part of the roaming agent's disturbing policy. Unlike the indirect regulation results such as [54–57, 94], the result in this chapter considers optimality and approximates the worst-case roaming agent disturbing policy while the influencing agent still achieves the objective. Compared to pursuit-evasion games such as [45, 46, 48, 49], the developed result aims to track and intercept an agent and indirectly steer it to a desired goal location despite uncertainties in the dynamics of both agents. Unlike [42, 52, 53], the result is computed online. A Lyapunov-based stability analysis is provided to show UUB convergence. Two simulations are included to demonstrate the performance of the developed approach. One simulation considers the roaming agent to have deterministic dynamics with uncertainties which are not modeled, while the second simulation considers the roaming agent with a random disturbance. Both simulations show that the influencing agent is able to sufficiently intercept and steer the roaming agent to the goal location.

In Chapter 7, conclusions are provided for each chapter. The contributions of each chapter are highlighted, with approaches taken and attributes. In addition, based on some limitations, possible extensions of the work in this dissertation are provided.

CHAPTER 2
PRELIMINARIES

2.1 Notation

Throughout the dissertation, $\mathbb{R}$ denotes the set of real numbers, $\mathbb{Z}$ denotes the set of integers, and $\mathbb{N}$ denotes the set of natural numbers. The sets of real n-vectors and $n \times m$ matrices are denoted by $\mathbb{R}^n$ and $\mathbb{R}^{n\times m}$, respectively. The set of numbers numbers greater than or equal to $a \in \mathbb{R}$ and strictly greater than a, are denoted by the subscripts $\geq$ and $>$, respectively. The $n \times n$ identity matrix and column vector of ones of dimension j are denoted by I_n and 1_j, respectively. The $n \times m$ matrix of zeros and ones is denoted by $0_{n\times m}$ and $1_{n\times m}$, respectively. The n-dimensional vectors of zeros and ones are denoted the notation 0_n, $0_{n\times 1}$, 1_n, and $1_{n\times 1}$, respectively. The partial derivative of h with respect to the state x is denoted by $\nabla h(x, y, \ldots)$. The notation $(\cdot)^T$ denotes the transpose of a matrix or vector. For a vector $\xi \in \mathbb{R}^m$, the notation $\mathrm{Tanh}(\xi) \in \mathbb{R}^m$ and $\mathrm{sgn}(\xi) \in \mathbb{R}^m$ are defined as $\mathrm{Tanh}(\xi) \triangleq [\tanh(\xi_i), \ldots, \tanh(\xi_m)]^T$ and $\mathrm{sgn}(\xi) \triangleq [\mathrm{sgn}(\xi_i), \ldots, \mathrm{sgn}(\xi_m)]^T$, respectively, where $\tanh(\cdot)$ denotes the hyperbolic tangent function and $\mathrm{sgn}(\cdot)$ denotes the signum function. The vectorization operator of a matrix $A = [a_1, a_2, \ldots, a_m] \in \mathbb{R}^{n\times m}$ is denoted by $\mathrm{vec}(A) \triangleq \left[a_1^T, a_2^T, \ldots, a_m^T\right]^T$, where $a_i \in \mathbb{R}^n$ denotes the i^{th}-column of the matrix A. The notation $\lambda_{\min}\{\cdot\}$, and $\lambda_{\max}\{\cdot\}$ denote the minimum and maximum eigenvalues, respectively. The notations $U[a, b]\, 1_n$ and $U[a, b]\, 1_{n\times m}$ denote a $n-$dimensional vector and $n \times m$ matrix, respectively, with elements selected from a uniform distribution on $[a, b]$. The notation $\overline{\|(\cdot)\|}$ is defined as $\overline{\|(\cdot)\|} \triangleq \sup_{\xi\in B_\zeta} \|(\cdot)\|$, and $\tilde{(\cdot)}$ denotes the estimation error defined as $\tilde{(\cdot)} \triangleq (\cdot) - \hat{(\cdot)}$ for an estimate $\hat{(\cdot)}$.

2.2 Problem Formulation

The focus of this dissertation is to obtain approximate optimal control policies for uncertain nonlinear systems. Specifically, the goal throughout the dissertation is to find

a controller $u : \mathbb{R}_{\geq t_0} \to \mathbb{R}^m$ which minimizes the cost functional

$$J(x, u) \triangleq \int_{t_0}^{\infty} r(x(\tau), u(\tau)) \, d\tau, \tag{2–1}$$

while regulating the system states $x : \mathbb{R}_{\geq t_0} \to \mathbb{R}^n$ to the origin under the control-affine dynamic constraints

$$\dot{x}(t) = F(x(t)) + G(x(t)) u(t), \tag{2–2}$$

with initial condition $x(t_0) = x_0 \in \mathbb{R}^n$, where $F : \mathbb{R}^n \to \mathbb{R}^n$ are drift dynamics, and $G : \mathbb{R}^n \to \mathbb{R}^{n \times m}$ is a locally Lipschitz continuous and bounded control effectiveness matrix. The system dynamics are assumed to satisfy the following assumption.

Assumption 2.1. The drift dynamics F and control effectiveness matrix G in (2–2) are locally Lipschitz and continuous. Moreover, the control effectiveness matrix is bounded, and the system in (2–2) is observable and controllable.

In the cost functional (2–1), $r : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}_{\geq 0}$ is the instantaneous user-defined positive-definite (PD) cost taking a form such as

$$r(x, u) = Q(x) + \Psi(u),$$

where $Q : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ and $\Psi : \mathbb{R}^m \to \mathbb{R}_{\geq 0}$ are PD functions penalizing the state and control actions.

2.3 Exact Solution

Provided the time-horizon is infinite, and the drift dynamics F, control effectiveness matrix G, and state penalty function Q are time-invariant (i.e., stationary), then the optimal control policy is also stationary feedback policy $u(t) = \pi(x(t))$ for some function $\pi : \mathbb{R}^n \to \mathbb{R}^m$.

Definition 2.1. [90]: Let $\Omega \subseteq \mathbb{R}^n$ be a set containing the origin. A control policy $\pi(x(t))$ is said to be admissible with respect to (2–2) in Ω, i.e. $\pi(x(t)) \in U(\Omega) \subset \mathbb{R}^m$, if $\pi(x(t))$ is continuous on Ω with $\pi(0) = 0_{m \times 1}$, $\pi(x(t))$ stabilizes (2–2) on Ω, and $V(x(t)) \triangleq J(x(t), \pi(x(t)))$ is finite.

Moreover, the infinite-horizon scalar function for the optimal solution, called the value function, is denoted by $V^* : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ and is expressed as

$$V^*(x) = \inf_{u(\tau)\in U|\tau\in\mathbb{R}_{\geq t}} \int_t^\infty r\left(\phi^u\left(\tau; t, x\right), u\left(\tau\right)\right) d\tau, \tag{2–3}$$

where $U \in \mathbb{R}^m$ is the set of admissible control actions and the notation $\phi^u(t; t_0, x_0)$ denotes the trajectory of the system in (2–2) under the controller u with initial condition $x_0 \in \mathbb{R}^n$ and initial time $t_0 \in \mathbb{R}_{\geq 0}$. The value function in (2–3) is characterized by the optimality condition, also known as the HJB equation, given as

$$0 = \nabla V^*\left(x\right)\left(F\left(x\right) + G\left(x\right)u^*\left(x\right)\right) + r\left(x, u^*\left(x\right)\right), \tag{2–4}$$

with $V\left(0\right) = 0$, which holds for all $x \in \mathbb{R}^n$ for an optimizing admissible policy $u^* : \mathbb{R}^n \to \mathbb{R}^m$ determined as

$$u^*\left(x\right) = \arg\min_u \left(\nabla V^*\left(x\right)\left(F\left(x\right) + G\left(x\right)u^*\left(x\right)\right) + r\left(x, u^*\left(x\right)\right)\right). \tag{2–5}$$

2.4 Value Function Approximation

The value function in (2–3) is an uncertain, and potentially, highly nonlinear function. Moreover, the HJB in (2–4) and policy in (2–5) depend on the gradient of the value function, but the value function is unknown. Hence, it is difficult to implement an optimal policy on an autonomous agent. However, using function approximation tools, such as parametric methods (i.e., neural networks), the value function can be approximated on a compact set. Moreover, let $\chi \subset \mathbb{R}^n$ be a compact set. Regional ADP approaches such as [11–15, 18, 62, 68, 69, 78] approximate the value function as

$$V^*\left(x\right) = W^T\sigma\left(x\right) + \epsilon\left(x\right), \tag{2–6}$$

where $W \in \mathbb{R}^L$ are unknown weights, $\sigma : \chi \to \mathbb{R}^L$ is a bounded vector of continuously differentiable nonlinear basis functions such that $\sigma(0) = 0$ and $\nabla\sigma(0) = 0$, and $\epsilon : \chi \to \mathbb{R}$ is the continuously differentiable and bounded function reconstruction error. The value

function in (2–6) is approximated using stationary basis function that represent the entire operating domain. However, for a large operating region, a large number of basis (and unknowns) is needed to sufficiently approximate the value function, which can be computationally expensive. Unlike regional value function approximation methods, local methods such as [1, 95, 96] approximate the value function inside a local neighborhood of the agent's state. Specifically, using StaF kernels centered at x such that, the value function can be evaluated at $y \in \overline{B_r(x)}$, where $\overline{B_r(x)}$ is a compact neighborhood around the agent's state such that

$$V^*(y) = W^T(x)\,\sigma(y, c(x)) + \epsilon(x, y), \tag{2–7}$$

where $W : \chi \to \mathbb{R}^L$ is the continuously differentiable state-dependent ideal StaF weight function, $\sigma : \chi \times \chi \to \mathbb{R}^L$ is the bounded vector of continuously differentiable nonlinear kernels, and $\epsilon : \chi \times \chi \to \mathbb{R}$ is the continuously differentiable function approximation error. Using the value function representation in (2–7), a reduced number of basis and unknowns is required to sufficiently approximate the value function. Specifically, [1, 95] have shown that a minimum of $n+1$ StaF basis is required for an n-dimensional system. Moreover, both value function approximations have their advantages and disadvantages; the regional approximation in (2–6) is computationally expensive but provides a global approximation of the value function, while the local approximation in (2–7) trades global optimality for computational efficiency, which motivates its use in real-time on hardware. Because both approximations provide a sufficient approximation of the value function, the work in Chapter 3 uses a combination of (2–6) and (2–7) to approximate the value function, and also discusses the advantages and disadvantages of each method.

Since the ideal weights in the value function are unknown, parametric estimates are substituted to get estimated versions of the value function and control policy. Specifically, a critic weight estimate $\hat{W}_c : \mathbb{R}_{\geq t_0} \to \mathbb{R}^L$ is substituted into the value function to get an estimated value function $\hat{V} : \mathbb{R}^L \times \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$, and an actor weight estimate

$\hat{W}_a : \mathbb{R}_{\geq t_0} \to \mathbb{R}^L$ in substituted into the policy to get an estimated implementable policy $\hat{u} : \mathbb{R}^L \times \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^m$. When the value function and policy estimates are substituted into (2–4), a residual $\delta : \mathbb{R}^L \times \mathbb{R}^L \times \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is formed, called the Bellman error (BE), and is represented as

$$\delta\left(y, x, \hat{W}_c, \hat{W}_a\right) \triangleq \nabla V\left(y, x, \hat{W}_c\right)\left(F\left(y\right) + G\left(y\right)\hat{u}\left(y, x, \hat{W}_a\right)\right) + r\left(y, \hat{u}\left(y, x, \hat{W}_a\right)\right). \tag{2–8}$$

The representation in (2–8) is based on the StaF kernel function approximation in (2–7). When using the approximation in (2–6), the BE in (2–8) is redefined as

$$\delta\left(x, \hat{W}_c, \hat{W}_a\right) \triangleq \nabla V\left(x, \hat{W}_c\right)\left(F\left(x\right) + G\left(x\right)\hat{u}\left(x, \hat{W}_a\right)\right) + r\left(x, \hat{u}\left(x, x, \hat{W}_a\right)\right). \tag{2–9}$$

The aim of the critic and actor weight estimates is to find weights which minimize the BE for all $x \in \mathbb{R}^n$.

2.5 System Identification

The aim in ADP is to find weights online which minimize the BE in (2–8). If the system is known, the BE in (2–8) can be evaluated at the current state $x\left(t\right)$ and time t to yield an instantaneous BE $\delta_t\left(t\right) \triangleq \delta\left(x\left(t\right), x\left(t\right), \hat{W}_c\left(t\right), \hat{W}_a\left(t\right)\right)$. The issue with (2–8) is that the BE depends on the system drift dynamics F, which may be unknown. Using the universal function approximation property [97, 98], the uncertain drift dynamics can be represented as

$$F\left(y\right) = F\left(y, \theta\right) \triangleq \theta^T Y\left(y\right) + \varepsilon\left(y\right), \tag{2–10}$$

where $\theta \in \mathbb{R}^{p \times n}$ is an unknown bounded weight, $Y : \mathbb{R}^n \to \mathbb{R}^p$ is a known basis, and $\varepsilon : \mathbb{R}^n \to \mathbb{R}^n$ is an unknown bounded function approximation error. Since, the weight θ is unknown, an estimate $\hat{\theta} : \mathbb{R}_{\geq t_0} \to \mathbb{R}^p$ is used to get estimate drift dynamics $\hat{F}\left(y, \hat{\theta}\right)$. The goal is to find update laws that drive the estimates $\hat{\theta}$ to the true weights θ. Various methods can be employed to learn learn the system uncertainties (cf.,

[58, 76, 81, 84, 99, 100]). For example, the ICL approach in [58] can be used to estimate the system uncertainties.

To show the development of the update laws for $\hat{\theta}$ using ICL, let $\Delta t_\theta \in \mathbb{R}_{>0}$ denote an integration time-window. Substituting (2–10) into (2–2) and integrating over the time $t_i \in [\Delta t_\theta, t]$, the system dynamics can be represented as $x(t_i) - x(t_i - \Delta t_\theta) = \theta^T \mathcal{S}_i + \mathcal{G}_i + \mathcal{E}_i$, where $\mathcal{S}_i \triangleq \int_{t_i - \Delta t_\theta}^{t_i} Y(x(\tau))\, d\tau$, $\mathcal{G}_i \triangleq \int_{t_i - \Delta t_\theta}^{t_i} G(x(\tau))\, u(\tau)\, d\tau$, and $\mathcal{E}_i \triangleq \int_{t_i - \Delta t_\theta}^{t_i} \varepsilon(x(\tau))\, d\tau$. To alleviate the need to inject a probing signal to satisfy the stringent PE condition, a least-square based estimate update law can be designed such as

$$\dot{\hat{\theta}}(t) = k_\theta \Gamma_\theta(t) \sum_{i=1}^{M} \mathcal{S}_i \left(x^T(t_i) - x^T(t_i - \Delta t_\theta) - \mathcal{G}_i^T - \mathcal{S}_i^T \hat{\theta}(t) \right), \tag{2–11}$$

$$\dot{\Gamma}_\theta(t) = \beta_\theta \Gamma_\theta(t) - k_\theta \Gamma_\theta(t) \sum_{i=1}^{M} \mathcal{S}_i \mathcal{S}_i^T \Gamma_\theta(t), \tag{2–12}$$

where $k_\theta, \beta_\theta \in \mathbb{R}_{>0}$ are user defined gains, and $M \in \mathbb{Z}_{>0}$ is the number of data points collected in the history stack.

Assumption 2.2. There exists a finite time $T_1 \in \mathbb{R}_{>0}$ and constant $\lambda_1 \in \mathbb{R}_{>0}$ such that for all $t \geq T_1$, $\lambda_1 I_p \leq \sum_{i=1}^{M} \mathcal{S}_i \mathcal{S}_i^T$.

Moreover, compared to the stringent PE condition, the time T_1 can be measured online as data is gathered. In addition, provided $\lambda_{\min}\left\{\Gamma_\theta^{-1}(t_0)\right\} > 0$ and Assumption 2.1 is satisfied, using similar arguments to [101, Corollary 4.3.2], Γ_θ satisfies $\underline{\Gamma}_\theta I_p \leq \Gamma_\theta(t) \leq \overline{\Gamma}_\theta I_p$, where $\underline{\Gamma}_\theta, \overline{\Gamma}_\theta \in \mathbb{R}_{>0}$.

Using the estimated system dynamics $\hat{F}\left(y, \hat{\theta}(t)\right)$, the BE in (2–8) can be rewritten as

$$\hat{\delta}\left(y, x, \hat{W}_c, \hat{W}_a, \hat{\theta}\right) \triangleq \nabla V\left(y, x, \hat{W}_c\right)\left(\hat{F}\left(y, \hat{\theta}\right) + G(y)\,\hat{u}\left(y, x, \hat{W}_a\right)\right) + r\left(y, \hat{u}\left(y, x, \hat{W}_a\right)\right). \tag{2–13}$$

The critic performs one-step updates to the critic weight estimates based on either the instantaneous experience, quantified by the squared error

$$\hat{\delta}\left(x\left(t\right),x\left(t\right),\hat{W}_{c}\left(t\right),\hat{W}_{a}\left(t\right),\hat{\theta}\left(t\right)\right)^{2}+\sum_{k=1}^{N}\hat{\delta}\left(x_{k}\left(t\right),x\left(t\right),\hat{W}_{c}\left(t\right),\hat{W}_{a}\left(t\right),\hat{\theta}\left(t\right)\right)^{2}, \quad \text{(2–14)}$$

or the cumulative experience, quantified by the integral square error

$$\begin{aligned} E\left(t\right) = & \int_{t_{0}}^{t}\left(\hat{\delta}\left(x\left(\tau\right),x\left(\tau\right),\hat{W}_{c}\left(\tau\right),\hat{W}_{a}\left(\tau\right),\hat{\theta}\left(\tau\right)\right)^{2}\right)d\tau \\ & +\int_{t_{0}}^{t}\left(\sum_{k=1}^{N}\hat{\delta}\left(x_{k}\left(\tau\right),x\left(\tau\right),\hat{W}_{c}\left(\tau\right),\hat{W}_{a}\left(\tau\right),\hat{\theta}\left(\tau\right)\right)^{2}\right)d\tau \end{aligned} \quad \text{(2–15)}$$

using a steepest descent based update law. In (2–14) and (2–15), $\{x_{k}:\mathbb{R}^{n}\times\mathbb{R}_{\geq t_{0}}\rightarrow\mathbb{R}^{n}\}_{k=1}^{N}$ are off-policy trajectories selected by the critic to alleviate the need to inject a probing signal into the system in order to facilitate learning. Specifically, the critic is designed to minimize (2–14) or (2–15), while the actor is designed to follow the critic. Provided Assumption 2.1 and specific gain conditions for the critic and actor update laws are satisfied, convergence of $x\left(t\right)$, $\tilde{W}_{c}\left(t\right)$, $\tilde{W}_{a}\left(t\right)$, and $\tilde{\theta}\left(t\right)$ to a neighborhood of zero can be established using the candidate Lyapunov function

$$V_{L}\left(x,\tilde{W}_{c},\tilde{W}_{a},\tilde{\theta},t\right)\triangleq V^{*}\left(x\right)+\frac{1}{2}\tilde{W}_{c}^{T}\Gamma_{c}^{-1}\left(t\right)\tilde{W}_{c}+\frac{1}{2}\tilde{W}_{a}^{T}\Gamma_{a}^{-1}\tilde{W}_{a}+\text{tr}\left(\tilde{\theta}^{T}\Gamma_{\theta}^{-1}\left(t\right)\tilde{\theta}\right)$$

when the system in (2–2) uses the policy $u\left(t\right)=\hat{u}\left(x\left(t\right),x\left(t\right)\hat{W}_{a}\left(t\right)\right)$, where $\Gamma_{c}:\mathbb{R}_{\geq t_{0}}\rightarrow\mathbb{R}^{L\times L}$ is a positive-definite least-squares gain matrix, $\Gamma_{a}\in\mathbb{R}^{L\times L}$ a positive definite gain, and $\tilde{\left(\cdot\right)}$ denotes the weight estimation errors defined in Section 2.1.

CHAPTER 3
APPROXIMATE DYNAMICS PROGRAMMING: COMBINING REGIONAL AND LOCAL STATE FOLLOWING APPROXIMATIONS

In this chapter, a novel framework is developed to merge local and regional value function approximation methods to yield an online optimal control method that is computationally efficient and simultaneously accurate over a specified critical region of the state-space. While the StaF method used in Chapters 4, 5, and 6 is computationally efficient, it lacks memory, i.e. the information about the value function in a region is lost once the system state leaves that region. To maintain an accurate approximation of the value function near the goal state (i.e., the origin), the developed method uses R-MBRL in A, which contains the origin; the weights are learned based on selected points in that set and the value function does not have to be re-learned once the state leaves this neighborhood. The developed architecture is motivated by the observation that in many applications such as station keeping of marine craft, like in [102], accurate approximation of the value function in a neighborhood of the goal state can improve the performance of the closed-loop system near the goal-state.

Since the StaF method uses state-dependent centers, the unknown optimal weight are themselves also state-dependent, which makes analyzing stability difficult. To add to the technical challenge, using a convex combination of R-MBRL and StaF results in a complex representation of the value function and resulting Bellman error. To provide insights into how to combine StaF and R-MBRL while also preserving stability, the estimates are designed using a Lyapunov-based stability analysis. The analysis of the closed-loop systems with the smoothly switching approximation guarantees UUB convergence. The performance of the developed method is illustrated through numerical simulations. Simulations are provided for a two-state system with a known value function as well as three, six, and ten-state systems with unknown value functions to illustrate the scalability of the method in terms of computational time, cost, and final RMS error. Comparisons with [1] and [18] illustrate the advantage of the developed method. The

results show that the optimal choice of the approximation method depends on several factors.

Notation

In the following chapter, the notation $(\cdot)^o$ denotes an arbitrary variable of the set which the variable belongs to. The notation $G_{\nabla F}, G_{\nabla F \nabla K}, G_F, G_{FK}$, and $G_{\nabla FK}$ is defined as $G_{\nabla F} \triangleq \nabla F g R^{-1} g^T \nabla F^T$, $G_{\nabla F \nabla K} \triangleq \nabla F g R^{-1} g^T \nabla K^T$, $G_F \triangleq F g R^{-1} g^T F^T$, $G_{FK} \triangleq F g R^{-1} g^T K^T$, and $G_{\nabla FK} \triangleq \nabla F g R^{-1} g^T K^T$ respectively, where F and K denote arbitrary functions.

3.1 Problem Formulation

Consider a control affine nonlinear dynamical system

$$\dot{x}(t) = f\big(x(t)\big) + g\big(x(t)\big)u(t), \tag{3–1}$$

where $x : \mathbb{R}_{\geq t_0} \to \mathbb{R}^n$ denotes the system state, $f : \mathbb{R}^n \to \mathbb{R}^n$ denotes the drift dynamics, $g : \mathbb{R}^n \to \mathbb{R}^{n \times m}$ denotes the control effectiveness, and $u : \mathbb{R}_{\geq t_0} \to \mathbb{R}^m$ denotes the control input.

Assumption 3.1. Both f and g are assumed to be locally Lipschitz continuous. Furthermore, $f(0) = 0$, and $\nabla f : \mathbb{R}^n \to \mathbb{R}^{n \times n}$ is continuous.

In the following, the notation $\phi^u(t; t_0, x_0)$ denotes the trajectory of the system in (3–1) under the controller u with initial condition $x_0 \in \mathbb{R}^n$ and initial time $t_0 \in \mathbb{R}_{\geq 0}$. The objective is to solve the infinite-horizon optimal regulation problem, i.e. find a control policy u online to minimize the cost functional

$$J(x, u) \triangleq \int_{t_0}^{\infty} r(x(\tau), u(\tau)) d\tau, \tag{3–2}$$

while regulating the system states to the origin under the dynamic constraint (3–1). In (3–2), $r : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}_{\geq 0}$ denotes the instantaneous cost defined as

$$r(x^o, u^o) \triangleq x^{oT} Q x^o + u^{oT} R u^o, \tag{3–3}$$

for all $x^o \in \mathbb{R}^n$ and $u^o \in \mathbb{R}^m$, where $R \in \mathbb{R}^{m\times m}$ and $Q \in \mathbb{R}^{n\times n}$ are constant PD matrices and the matrix Q can be bounded as $\underline{q}\|x^o\|^2 \leq x^{oT}Qx^o \leq \overline{q}\|x^o\|^2$.

The infinite-horizon scalar value function for the optimal solution, i.e. the function which maps each state to the total cost-to-go, denoted by $V^* : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$, can be expressed as

$$V^*(x^o) = \inf_{u(\tau)\in U|\tau\in\mathbb{R}_{\geq t}} \int_t^\infty r\left(\phi^u\left(\tau; t, x^o\right), u\left(\tau\right)\right) d\tau, \tag{3–4}$$

where $U \subset \mathbb{R}^m$ is the action space. The optimal value function is characterized by the corresponding HJB equation

$$\nabla V^*(x^o)\big(f(x^o) + g(x^o)u^*(x^o)\big) + r\big(x^o, u^*(x^o)\big) = 0, \tag{3–5}$$

with the boundary condition $V(0) = 0$, where $u^* : \mathbb{R}^n \to \mathbb{R}^m$ is the optimal control policy which can be determined from (3–5) as

$$u^*(x^o) \triangleq -\frac{1}{2}R^{-1}g^T(x^o)\big(\nabla V^*(x^o)\big)^T. \tag{3–6}$$

Using (3–6), the open-loop HJB in (3–5) can be expressed in a closed-loop form as

$$-\frac{1}{4}\nabla V^*\left(x^o\right) g\left(x^o\right) R^{-1} g^T\left(x^o\right)\left(\nabla V^*\left(x^o\right)\right)^T + \nabla V^*\left(x^o\right) f\left(x^o\right) + x^{oT}Qx^o = 0. \tag{3–7}$$

The analytical expression in (3–6) requires knowledge of the optimal value function which is the solution to the HJB in (3–5), but since the analytical solution for the HJB is generally infeasible to compute, an approximation of the solution is sought.

3.2 Combining Regional and Local State Following Approximations

Traditional approaches to approximating the value function establish the approximation over the entire state-space. When implementing the approximation online, traditional methods spend computational resources approximating the value function in regions where the state may not enter. The StaF method reduces the computational

efforts of the approximation problem by approximating the value function in a moving neighborhood of the state.

A drawback of the StaF method is that it does not establish an approximation of the value function in regions where the state will travel in the future; the StaF method only approximates the value function at the *current* position of the state. In general, it is difficult to provide a perfect prediction of the future state of an uncertain nonlinear system. However, since convergence to the origin is the goal of regulation problems, approximating the function in a neighborhood around the origin is well motivated.

The operating domain χ of the state is segregated into two sets, the set A, which is a closed compact set containing the origin, and the set $B = \chi \setminus A$. Two different approximation strategies will be used over A and B. Various R-MBRL methods can be used to approximate the value function inside A. For the set B, the StaF method is employed since there are large regions of B that the state does not visit for the regulation problem. Thus, the value function is approximated by the StaF method when the state is in B and some R-MBRL method is used when the state is in A. A regional approximation method is also used to approximate the value function in the set $A' = \{x \in \chi : d(x, A) \leq \ell\}$ (also known as an *inflation* of A), where $d(x, A) = \inf\{d(x, y) : y \in A\}$ and $\ell \in \mathbb{R}_{>0}$ is a constant, and approximation of the value function over the transition region $A' \setminus A$ will be a state dependent convex combination of the two controllers.

Let $\hat{V}_1(x)$ denote the approximation of the value function over A' using the R-MBRL method, and denote $\hat{V}_2(x)$ as the StaF approximation of the value function over B. The resulting approximation of the value function over χ will then be $\hat{V}(x) = \lambda(x)\hat{V}_1(x) + (1 - \lambda(x))\hat{V}_2(x)$, where $\lambda : \chi \to [0, 1]$ such that $\lambda(x) = 1$ when $x \in A$ and $\lambda(x) = 0$ when $x \in \chi \setminus A' \subset B$. If $\epsilon > 0$ and $|\hat{V}_1(x) - V^*(x)| < \epsilon$ over A' and $|\hat{V}_2(x) - V^*(x)| < \epsilon$ over B, then $|\hat{V}(x) - V^*(x)| < \epsilon$ for all $x \in \chi$, since $\hat{V}$ is a convex combination of $\hat{V}_1$ and $\hat{V}_2$ over the transition region $A' \setminus A \subset B$.

The following analysis is agnostic with respect to the compact set A and the transition function λ. However, the transition function λ should be a continuously differentiable compactly supported function such that $\|\nabla\lambda(x^o)\| \leq \overline{\nabla\lambda}$, where $\overline{\nabla\lambda} \in \mathbb{R}_{>0}$. An example of such a function is

$$\lambda(x) = \begin{cases} 1, & x \in A, \\ \frac{1}{2}\left[1 + \cos(\pi \frac{d(x,A)}{\ell})\right], & x \in A' \setminus A, \\ 0, & x \notin A'. \end{cases} \tag{3–8}$$

Examples of A for which λ is continuously differentiable include $[-1,1]^n$ as well as $\overline{B_1(0)} = \{y \in \mathbb{R}^n : \|y\| \leq 1\}$.

3.3 Value Function Approximation

The value function V^* evaluated at x^o using StaF kernels centered at $y^o \in \overline{B_r(x^o)}$ can be represented using a convex combination as

$$V^*(x^o) = \lambda(x^o)W_1^T\sigma(x^o) + \big(1 - \lambda(x^o)\big)W_2^T(y^o)\phi\big(x^o, c(y^o)\big) + \epsilon(x^o, y^o). \tag{3–9}$$

In (3–9), $\sigma : \chi \to \mathbb{R}^P$ is a bounded vector of continuously differentiable nonlinear basis functions such that $\sigma(0) = 0$ and $\nabla\sigma(0) = 0$, $\phi\big(x^o, c(y^o)\big) = \big[k(x^o, c_1(y^o), \ldots, k(x^o, c_L(y^o)\big]^T$ where $k : \chi \times \chi^L \to \mathbb{R}^L$ is a strictly positive definite continuously differentiable kernel, $W_1 \in \mathbb{R}^P$ is a constant ideal R-MBRL weight vector which is upper-bounded by a known positive constant $\overline{W}_1$ such that $\|W_1\| \leq \overline{W}_1$ (cf., [18, 21, 78, 98, 103]). Furthermore, $W_2 : \chi \to \mathbb{R}^L$ is the continuously differentiable ideal local StaF weight function which changes with the state dependent centers, and $\epsilon : \chi \to \mathbb{R}$ is the continuously differentiable function reconstruction error such that $\sup_{x^o \in \chi,\, y^o \in \overline{B_r(x^o)}} |\epsilon(x^o, y^o)| \leq \overline{\epsilon}$ and $\sup_{x^o \in \chi,\, y^o \in \overline{B_r(x^o)}} |\nabla\epsilon(x^o, y^o)| \leq \overline{\nabla\epsilon}$.

The subsequent analysis is based on an approximation of the value function and optimal policy, evaluated at x^o using StaF kernels centered at $y^o \in \overline{B_r(x^o)}$, expressed as

$$\hat{V}(x^o, y^o, \hat{W}_{1c}, \hat{W}_{2c}) = \lambda(x^o)\hat{W}_{1c}^T\sigma(x^o) + (1 - \lambda(x^o))\hat{W}_{2c}^T\phi(x^o, c(y^o)), \tag{3–10}$$

and

$$\hat{u}(x^o, y^o, \hat{W}_{1a}, \hat{W}_{2a}) = -\frac{1}{2}R^{-1}g^T(x^o)\Big(\lambda(x^o)\nabla\sigma^T(x^o)\hat{W}_{1a} + (1 - \lambda(x^o))\nabla\phi^T(x^o, c(y^o))\hat{W}_{2a} + \nabla\lambda^T(x^o)\big(\sigma^T(x^o)\hat{W}_{1a} - \phi^T(x^o, c(y^o))\hat{W}_{2a}\big)\Big). \tag{3–11}$$

In (3–10) and (3–11), $\hat{W}_{1a}$, $\hat{W}_{1c} \in \mathbb{R}^P$ and $\hat{W}_{2a}$, $\hat{W}_{2c} \in \mathbb{R}^L$ are weight estimates for the ideal weight vectors W_1 and $W_2(y^o)$ respectively, and λ denotes the transition function introduced in Section 3.2. In an approximate actor-critic based solution, the optimal value function V^* and control policy u^* in (3–5) are replaced by their respective estimates $\hat{V} : \chi \times \mathbb{R}^L \times \mathbb{R}^P \to \mathbb{R}$ and $\hat{u} : \chi \times \mathbb{R}^L \times \mathbb{R}^P \to \mathbb{R}^m$. This results in a residual error $\delta : \mathbb{R}^n \times \mathbb{R}^L \times \mathbb{R}^L \times \mathbb{R}^P \times \mathbb{R}^P \to \mathbb{R}$ called the BE which is defined as

$$\delta\big(x^o, y^o, \hat{W}_{1c}, \hat{W}_{2c}, \hat{W}_{1a}, \hat{W}_{2a}\big) \triangleq \nabla\hat{V}\big(x^o, y^o, \hat{W}_{1c}, \hat{W}_{2c}\big)\big(f(x^o) + g(x^o)\hat{u}(x^o, y^o, \hat{W}_{1a}, \hat{W}_{2a})\big) + r\big(x^o, \hat{u}(x^o, y^o, \hat{W}_{1a}, \hat{W}_{2a})\big). \tag{3–12}$$

Motivated by classical ADP solutions which aim to find a set of weights so that the BE is zero $\forall x^o \in \mathbb{R}^n$, to solve the optimal control problem, the critics and actors aim to find a set of weights that minimize the BE $\forall x^o \in \mathbb{R}^n$.

3.4 Online Learning

At a given time instant t, the Bellman error $\delta_t : \mathbb{R}_{\geq 0} \to \mathbb{R}$ is evaluated as

$$\delta_t(t) \triangleq \delta\big(x(t), x(t), \hat{W}_{1c}(t), \hat{W}_{2c}(t), \hat{W}_{1a}(t), \hat{W}_{2a}(t)\big), \tag{3–13}$$

where $\hat{W}_{1c}$, $\hat{W}_{1a}$, and $\hat{W}_{2c}$, $\hat{W}_{2a}$, denote estimates of the critic and actor weights for the R-MBRL approximation method and StaF approximation method, respectively, at time t. Furthermore, $x(t)$ denotes the state of the system in (3–1) when starting from initial time

t_0 and initial state x_0 under the influence of the state feedback controller

$$u(t) = \hat{u}\big(x(t), x(t), \hat{W}_{1a}(t), \hat{W}_{2a}(t)\big). \tag{3–14}$$

The BE is extrapolated to unexplored areas of the state space to learn via simulation of experience (cf. [1, 18]). The critic $\hat{W}_{1c}$ selects sample points $\{x_i \in A' | i = 1, \cdots, N\}$ based on prior information about the desired behavior of the system, i.e. selected about the origin, and evaluates a form of the BE, $\delta_{1t,i} : \mathbb{R}_{\geq t_0} \to \mathbb{R}$. Similarly, sample trajectories $\{x_j(x(t), t) \in B_r(x(t)) \big| j = 1, 2, \ldots, M\}$ that follow the current state $x(t)$ are selected so that the StaF critic $\hat{W}_{2c}$ evaluates another extrapolated form of the BE $\delta_{2t,j} : \mathbb{R}_{\geq t_0} \to \mathbb{R}$. The extrapolated BEs are expressed as

$$\delta_{1t,i}(t) = \hat{W}_{1c}^T(t)\omega_{\nabla\sigma i}(t) + r\big(x_i, \hat{u}_i(t)\big), \tag{3–15}$$

$$\delta_{2t,j}(t) = \hat{W}_{2c}^T(t)\omega_{\nabla\phi j}(t) + r\big(x_j(x(t), t), \hat{u}_j(t)\big), \tag{3–16}$$

where

$$\omega_{\nabla\sigma i}(t) \triangleq \nabla\sigma(x_i)\Big(f(x_i) + g(x_i)\hat{u}_i(t)\Big),$$
$$\omega_{\nabla\phi j}(t) \triangleq \nabla\phi\big(x_j(x(t), t), c(x(t))\big)\left(f(x_j(x(t), t)) + g(x_j(x(t), t))\hat{u}_j(t)\right),$$

and

$$\hat{u}_i(t) = -\frac{1}{2}R^{-1}g^T(x_i)\nabla\sigma(x_i)^T\hat{W}_{1a}(t),$$
$$\hat{u}_j(t) = -\frac{1}{2}R^{-1}g^T\big(x_j(x(t), t)\big)\nabla\phi\big(x_j(x(t), t), c(x(t))\big)^T\hat{W}_{2a}(t).$$

3.4.1 Regional Update Laws

The BE and extrapolated BE in (3–13) and (3–15), respectively, contain R-MBRL actor and critic estimates, $\hat{W}_{1a}$ and $\hat{W}_{1c}$. Various approximation methods could be used to evaluate the BE in A. See [1, 18, 32, 67, 78] for examples of R-MBRL actor and critic update laws.

3.4.2 Local Update Laws

While the state is not in the local domain of A', the StaF critic uses the BEs in (3–13) and (3–16) to improve the estimate of $\hat{W}_{2c}$. Specifically, the StaF critic can be designed using the recursive least-squares update law

$$\dot{\hat{W}}_{2c}(t) = -k_{c1}\Gamma_2(t)\frac{\omega_{\nabla\phi}(t)}{\rho_2(t)}\delta_t(t) - \frac{k_{c2}}{M}\Gamma_2(t)\sum_{j=1}^{M}\frac{\omega_{\nabla\phi j}(t)}{\rho_{2j}(t)}\delta_{2t,j}(t), \tag{3–17}$$

$$\dot{\Gamma}_2(t) = \beta_2\Gamma_2(t) - k_{c1}\Gamma_2(t)\frac{\omega_{\nabla\phi}(t)\omega_{\nabla\phi}^T(t)}{\rho_2^2(t)}\Gamma_2(t) - \frac{k_{c2}}{M}\Gamma_2(t)\sum_{j=1}^{M}\frac{\omega_{\nabla\phi j}(t)\omega_{\nabla\phi j}^T(t)}{\rho_{2j}^2(t)}\Gamma_2(t) \tag{3–18}$$

where $\Gamma_2(t_0) = \Gamma_{2o}$ and $\Gamma_2(t)$ is the least-squares learning gain matrix, $k_{c1},\ k_{c2}, \in \mathbb{R}_{\geq 0}$ are constant adaptation gains, $\beta_2 \in \mathbb{R}_{\geq 0}$ is a constant forgetting factor, $\rho_2(t) \triangleq 1 + \gamma_2\omega_{\nabla\phi}^T(t)\omega_{\nabla\phi}(t)$, $\rho_{2j}(t) \triangleq 1 + \gamma_2\omega_{\nabla\phi j}^T(t)\omega_{\nabla\phi j}(t)$, and $\gamma_2 \in \mathbb{R}_{\geq 0}$ is a constant positive gain. In (3–18)

$$\begin{aligned}\omega_{\nabla\phi}(t) \triangleq &\Big(\left(1 - \lambda(x(t))\right)\nabla\phi\left(x(t), c(x(t))\right) \Big) + \phi\big(x(t), c(x(t))\big)\nabla\lambda(x(t))\Big) \\ &\times \Big(f(x(t)) + g(x(t))\hat{u}\big(x(t), x(t), \hat{W}_{1a}(t), \hat{W}_{2a}(t)\big)\Big)\end{aligned} \tag{3–19}$$

is an instantaneous regressor matrix. The StaF actor update law is given by

$$\begin{aligned}\dot{\hat{W}}_{2a}(t) = &-k_{a1}\big(\hat{W}_{2a}(t) - \hat{W}_{2c}(t)\big) - k_{a2}\hat{W}_{2a}(t) + \frac{k_{c1}G_{\nabla\phi}^T(t)\hat{W}_{2a}(t)\omega_{\nabla\phi}^T(t)}{4\rho_2(t)}\hat{W}_{2c}(t) \\ &+ \frac{k_{c2}}{4M}\sum_{j=1}^{M}\frac{G_{\nabla\phi j}^T(t)\hat{W}_{2a}(t)\omega_{\nabla\phi j}^T(t)}{\rho_{2j}(t)}\hat{W}_{2c}(t),\end{aligned} \tag{3–20}$$

where $k_{a1},\ k_{a2} \in \mathbb{R}$ are positive constant adaptation gains and $G_{\nabla\phi}(t) \triangleq \nabla\phi\big(x(t), c(x(t))\big)g(x(t))R^{-1}g^T(x(t))\nabla\phi^T\big(x(t), c(x(t))\big)$.

Remark 3.1. In typical BE extrapolation approaches, the extrapolated BEs $\delta_{1t,i},\ \delta_{2t,j}$ and controls $\hat{u}_i(t),\ \hat{u}_j(t)$ take similar forms to the actual BE δ_t and control $u\left(t\right)$, respectively, with the exception of using extrapolated states. However, the extrapolated BEs and

extrapolated inputs in this work take a different form compared to the true BE and control. The goal is to approximate the ideal weight W_1 irrespective of the system state and W_2 in a region around the state, therefore the extrapolated BEs do not rely on a convex combination in the transition region $A' \setminus A$. Furthermore, when the state is in $B = \chi \setminus A$, only BE extrapolation is used in A' to approximate the weight W_1. Hence, the developed method is fundamentally different from the approach in [1] and [18].

3.5 Stability Analysis

For notational brevity, unless otherwise specified, time dependence is suppressed in subsequent equations, trajectories, and definitions. The approach in this chapter was generalized to allow the use of any model-based approximation method in A. However, to facilitate the following analysis a certain structure is given to the R-MBRL update laws. Without a loss of generality, let the R-MBRL update laws take a similar form to the StaF update laws in (3–17), (3–18), and (3–20). The R-MBRL update laws contain the extrapolated regressor $\omega_{\nabla\sigma i}$ defined in Section 3.4, where the regressor $\omega_{\nabla\sigma}$ is defined as

$$\omega_{\nabla\sigma} \triangleq \Big(\lambda\left(x\right)\nabla\sigma\left(x\right) + \sigma\left(x\right)\nabla\lambda\left(x\right)\Big)\Big(f\left(x\right) + g\left(x\right)u\Big), \tag{3–21}$$

where the BE δ_t is defined in (3–22), and the extrapolated BE $\delta_{1t,i}$ is defined in (3–23). The constant gains for the R-MBRL update laws are $\eta_{c1},\, \eta_{c2} \in \mathbb{R}_{\geq 0}$. The R-MBRL least-squares learning gain matrix is $\Gamma_1(t)$ with a forgetting factor $\beta_1 \in \mathbb{R}_{\geq 0}$, and with normalizing factors $\rho_1(t) \triangleq 1 + \gamma_1\omega_{\nabla\sigma}^T(t)\omega_{\nabla\sigma}(t)$, $\rho_{1i}(t) \triangleq 1 + \gamma_1\omega_{\nabla\sigma i}^T(t)\omega_{\nabla\sigma i}(t)$ where $\gamma_1 \in \mathbb{R}_{\geq 0}$ is a constant positive gain.

To facilitate the analysis, let $\tilde{W}_{1a} \triangleq W_1 - \hat{W}_{1a}$, $\tilde{W}_{1c} \triangleq W_1 - \hat{W}_{1c}$, $\tilde{W}_{2a} \triangleq W_2 - \hat{W}_{2a}$, and $\tilde{W}_{2c} \triangleq W_2 - \hat{W}_{2c}$ denote the weight estimation errors. Unmeasurable forms of the BEs in (3–13), (3–15), and (3–16) can be written as

$$\delta_t = \delta_{t1} + \delta_{t2} + \delta_{t3}, \tag{3–22}$$

where

$$\delta_{t1} = -\omega_{\nabla\sigma}^T \tilde{W}_{1c} + \frac{1}{4}\lambda^2 \tilde{W}_{1a}^T G_{\nabla\sigma} \tilde{W}_{1a} + \Delta_1,$$
$$\delta_{t2} = -\omega_{\nabla\phi}^T \tilde{W}_{2c} + \frac{1}{4}(1-\lambda)^2 \tilde{W}_{2a}^T G_{\nabla\phi} \tilde{W}_{2a} + \Delta_2,$$
$$\begin{aligned}\delta_{t3} = {} & \frac{1}{2}(1-\lambda)\Big(\lambda \tilde{W}_{2a}^T G_{\nabla\phi\nabla\sigma} \tilde{W}_{1a} + \tilde{W}_{1a}^T \sigma G_{\nabla\lambda\nabla\phi} \tilde{W}_{2a} - \frac{1}{2}\tilde{W}_{2a}^T \phi G_{\nabla\lambda\nabla\phi} \tilde{W}_{2a}\Big) \\ & + \frac{1}{4}\Big(\tilde{W}_{1a}^T \sigma G_{\nabla\lambda} \sigma^T \tilde{W}_{1a} - 2\tilde{W}_{2a}^T \phi G_{\nabla\lambda} \sigma^T \tilde{W}_{1a} + \tilde{W}_{2a}^T \phi G_{\nabla\lambda} \phi^T \tilde{W}_{2a}\Big) \\ & + \frac{1}{2}\lambda\Big(\tilde{W}_{1a}^T \sigma G_{\nabla\lambda\nabla\sigma} \tilde{W}_{1a} - \tilde{W}_{2a}^T \phi G_{\nabla\lambda\nabla\sigma} \tilde{W}_{1a}\Big) + \Delta_3\end{aligned}$$

and

$$\begin{aligned}\delta_{1t,i} &= -\omega_{\nabla\sigma i}^T \tilde{W}_{1c} + \frac{1}{4}\tilde{W}_{1a}^T G_{\nabla\sigma i} \tilde{W}_{1a} + \Delta_{1i} \\ \delta_{2t,j} &= -\omega_{\nabla\phi j}^T \tilde{W}_{2c} + \frac{1}{4}\tilde{W}_{2a}^T G_{\nabla\phi j} \tilde{W}_{2a} + \Delta_{2j},\end{aligned} \tag{3–23}$$

where the functions $\Delta_1, \Delta_2, \Delta_3, \Delta_{1i}, \Delta_{2j} : \mathbb{R}^n \to \mathbb{R}$ are uniformly bounded over χ such that the bounds $\left\{\overline{\|\Delta_k\|} \middle| k = 1,2,3\right\}$, $\overline{\|\Delta_{1i}\|}$, and $\overline{\|\Delta_{2j}\|}$ decrease with decreasing $\overline{\|\nabla\epsilon\|}$ and $\overline{\|\nabla W\|}$.

Using the R-MBRL and StaF update laws, the system states x and selected states x_i and x_j are assumed to satisfy the following inequalities.

Assumption 3.2. There exists a positive constant $T \in \mathbb{R}_{\geq 0}$ such that

$$\underline{c}_1 I_P \leq \int_t^{t+T} \left(\frac{\omega_{\nabla\sigma}(\tau)\,\omega_{\nabla\sigma}^T(\tau)}{\rho_1^2(\tau)}\right) d\tau,\ \forall t \in \mathbb{R}_{\geq t_0},$$
$$\underline{c}_2 I_P \leq \inf_{t \in \mathbb{R}_{\geq t_0}} \left(\frac{1}{N}\sum_{i=1}^{N} \frac{\omega_{\nabla\sigma i}(t)\,\omega_{\nabla\sigma i}^T(t)}{\rho_{1i}^2(t)}\right),$$
$$\underline{c}_3 I_P \leq \frac{1}{N}\int_t^{t+T} \left(\sum_{i=1}^{N} \frac{\omega_{\nabla\sigma i}(\tau)\,\omega_{\nabla\sigma i}^T(\tau)}{\rho_{1i}^2(\tau)}\right) d\tau,\ \forall t \in \mathbb{R}_{\geq t_0},$$
$$\underline{b}_1 I_L \leq \int_t^{t+T} \left(\frac{\omega_{\nabla\phi}(\tau)\,\omega_{\nabla\phi}^T(\tau)}{\rho_1^2(\tau)}\right) d\tau,\ \forall t \in \mathbb{R}_{\geq t_0},$$

$$\underline{b}_2 I_L \leq \inf_{t \in \mathbb{R}_{\geq t_0}} \left(\frac{1}{M} \sum_{j=1}^{M} \frac{\omega_{\nabla \phi j}(t)\, \omega_{\nabla \phi j}^T(t)}{\rho_{2j}^2(t)} \right),$$

$$\underline{b}_3 I_L \leq \frac{1}{M} \int_t^{t+T} \left(\sum_{j=1}^{M} \frac{\omega_{\nabla \phi j}(\tau)\, \omega_{\nabla \phi j}^T(\tau)}{\rho_{2j}^2(\tau)} \right) d\tau, \ \forall t \in \mathbb{R}_{\geq t_0},$$

where $\{\underline{c}_k | k = 1, 2, 3\}$, $\{\underline{b}_k | k = 1, 2, 3\} \in \mathbb{R}_{\geq 0}$ are nonnegative constants, and at least one of the constants from each set is strictly positive.

Remark 3.2. Assumption 3.2 requires the regressors $\omega_{\nabla \sigma}$, $\omega_{\nabla \phi}$ or $\omega_{\nabla \sigma i}$, $\omega_{\nabla \phi j}$ to be persistently exciting. The regressors $\omega_{\nabla \sigma}$ and $\omega_{\nabla \phi}$ are completely determined by the state x and weights $\hat{W}_{1a}$ and $\hat{W}_{2a}$. Typically, to ensure that $\underline{c}_1, \underline{b}_1 > 0$, meaning $\omega_{\nabla \sigma}$ and $\omega_{\nabla \phi}$ are persistently excited, a probing signal is added to the control input. However, this introduces undesired oscillations in the system and produces noisy signals in the response. In addition, as the system and state converge to the origin, excitation will usually vanish. Hence, it is difficult to ensure that $\underline{c}_1, \underline{b}_1 > 0$. On the other hand, $\omega_{\nabla \sigma i}$ and $\omega_{\nabla \phi j}$ are dependent on x_i and x_j, which are designed independent of the system state x. In fact, $\omega_{\nabla \sigma i}$ is designed based on the desired behavior of the system, i.e. regulate the states to the origin. Therefore, without the need of a probing signal, $\underline{c}_2$ and $\underline{b}_2$ can be made strictly positive by selecting a sufficient number of extrapolated sample states in both regions of the state space, or if x_i and x_j contain enough frequencies then $\underline{c}_3$, $\underline{b}_3$ become strictly positive.[1]

Let a candidate Lyapunov function $V_L : \mathbb{R}^{n+2L+2P} \times \mathbb{R}_{\geq 0} \to \mathbb{R}$ be defined as

$$V_L(Z, t) = V^*(x) + \frac{1}{2} \tilde{W}_{1c}^T \Gamma_1^{-1}(t) \tilde{W}_{1c} + \frac{1}{2} \tilde{W}_{2c}^T \Gamma_2^{-1}(t) \tilde{W}_{2c} + \frac{1}{2} \tilde{W}_{1a}^T \tilde{W}_{1a} + \frac{1}{2} \tilde{W}_{2a}^T \tilde{W}_{2a},$$

[1] Typical results in ADP require excitation along the system trajectory (cf. [12, 64–66, 75, 78, 79]), which may potentially cause the system to go unstable. However, in this result, virtual excitation can be used without injecting destabilizing dither signals into the system. The sample trajectories x_i and x_j can be designed to contain enough frequencies if they are selected to follow a highly oscillatory trajectory or are chosen from a sampling distribution such as a normal or uniform distribution.

where V^* is the unknown, positive, continuously differentiable optimal value function, and

$$Z = \left[x^T,\ \tilde{W}_{1a}^T,\ \tilde{W}_{1c}^T,\ \tilde{W}_{2a}^T,\ \tilde{W}_{2c}^T \right]^T.$$

The least-squares update laws which take the form of (3–18) ensure that the least-squares gain matrices satisfy [101, Corollary 4.3.2]

$$\underline{\Gamma_1} I_P \leq \Gamma_1(t) \leq \overline{\Gamma_1} I_P, \tag{3–24}$$

$$\underline{\Gamma_2} I_L \leq \Gamma_2(t) \leq \overline{\Gamma_2} I_L, \tag{3–25}$$

provided the minimum eigenvalues $\lambda_{\min}\left\{\Gamma_{1o}^{-1}\right\}$, $\lambda_{\min}\left\{\Gamma_{2o}^{-1}\right\} > 0$ and Assumption 3.2 holds (see [1]).

Since the optimal value function V^* is positive definite, using [104, Lemma 4.3], the candidate Lyapunov function V_L can be bounded as

$$\underline{\nu_l}\left(\|Z^o\|\right) \leq V_L\left(Z^o, t\right) \leq \overline{\nu_l}\left(\|Z^o\|\right) \tag{3–26}$$

for all $t \in \mathbb{R}_{\geq t_0}$ and for all $Z^o \in \mathbb{R}^{n+2L+2P}$, where $\underline{\nu_l}, \overline{\nu_l} : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ in (3–26) are class $\mathcal{K}$ functions. To facilitate the analysis, let $\underline{c}, \underline{b} \in \mathbb{R}_{>0}$ be constants defined as

$$\underline{c} \triangleq \frac{\beta_1}{2\overline{\Gamma}_1 \eta_{c2}} + \frac{\underline{c_2}}{2}, \qquad \underline{b} \triangleq \frac{\beta_2}{2\overline{\Gamma}_2 k_{c2}} + \frac{\underline{b_2}}{2}. \tag{3–27}$$

Let $\nu_l : \mathbb{R}_{\geq 0}$ be a class $\mathcal{K}$ function such that

$$\nu_l(||Z||) \leq \frac{q}{2}\|x\|^2 + \frac{(k_{a1}+k_{a2})}{8}\left\|\tilde{W}_{2a}\right\|^2 + \frac{k_{c2}\underline{b}}{8}\left\|\tilde{W}_{2c}\right\|^2 + \frac{(\eta_{a1}+\eta_{a2})}{8}\left\|\tilde{W}_{1a}\right\|^2 + \frac{\eta_{c2}\underline{c}}{8}\left\|\tilde{W}_{1c}\right\|^2. \tag{3–28}$$

Theorem 3.1. *Provided Assumption 3.2 is satisfied and the control gains are selected sufficiently large (see Appendix A.1), then the controller in (3–11) along with the R-MBRL and StaF update laws taking the form of (3–17)-(3–20) ensure that the state x and weight estimation errors $\tilde{W}_{1a}$, $\tilde{W}_{1c}$, $\tilde{W}_{2a}$, and $\tilde{W}_{2c}$ are uniformly ultimately bounded.*[2]

Proof. The time-derivative of the Lyapunov function is

$$\dot{V}_L = \dot{V}^* - \tilde{W}_{1c}^T \Gamma_1^{-1} \dot{\hat{W}}_{1c} + \tilde{W}_{2c}^T \Gamma_2^{-1} (\dot{W}_2 - \dot{\hat{W}}_{2c}) - \tilde{W}_{1a}^T \dot{\hat{W}}_{1a} + \tilde{W}_{2a} (\dot{W}_2 - \dot{\hat{W}}_{2a}) + \frac{1}{2} \tilde{W}_{1c}^T \dot{\Gamma}_1^{-1} \tilde{W}_{1c} + \frac{1}{2} \tilde{W}_{2c}^T \dot{\Gamma}_2^{-1} \tilde{W}_{2c}. \tag{3–29}$$

Using the fact that $\dot{W}_2 = \nabla W_2 (x) (f (x) + g (x) u)$, $\dot{V}^* = \nabla V^* (x) (f (x) + g (x) u)$, and adding and subtracting $\nabla V^* g (x) u^*$, (3–29) is expressed as

$$\begin{aligned}
\dot{V}_L = {} & \nabla V^* (f + g u^*) + \nabla V^* g (u - u^*) - \tilde{W}_{1c}^T \Gamma_1^{-1} (t) \dot{\hat{W}}_{1c} \\
& + \tilde{W}_{2c}^T \Gamma_2^{-1} (t) \nabla W_2 (f + g u) - \tilde{W}_{2c}^T \Gamma_2^{-1} (t) \dot{\hat{W}}_{2c} - \tilde{W}_{1a}^T \dot{\hat{W}}_{1a} \\
& + \tilde{W}_{2a}^T \nabla W_2 (f + g u) - \tilde{W}_{2a}^T \dot{\hat{W}}_{2a} - \frac{1}{2} \tilde{W}_{1c}^T \Gamma_1^{-1} \dot{\Gamma}_1 \Gamma_1^{-1} \tilde{W}_{1c} \\
& - \frac{1}{2} \tilde{W}_{2c}^T \Gamma_2^{-1} \dot{\Gamma}_2 \Gamma_2^{-1} \tilde{W}_{2c}.
\end{aligned}$$

Using (3–5), substituting in (3–6), (3–14), and the gradient of (3–9), the Lyapunov function derivative is further expressed as

$$\begin{aligned}
\dot{V}_L = {} & -x^T Q x - u^{*T} R u^* - \tilde{W}_{1c}^T \Gamma_1^{-1} (t) \dot{\hat{W}}_{1c} - \tilde{W}_{2c}^T \Gamma_2^{-1} (t) \dot{\hat{W}}_{2c} \\
& + \frac{1}{2} \lambda \nabla V^* G \nabla \sigma^T \tilde{W}_{1a} + \frac{1}{2} \nabla V^* G \nabla \lambda^T \sigma^T \tilde{W}_{1a} \\
& + \frac{1}{2} (1 - \lambda) \nabla V^* G \nabla \phi^T \tilde{W}_{2a} - \frac{1}{2} \nabla V^* G \nabla \lambda^T \phi^T \tilde{W}_{2a} \\
& + \frac{1}{2} (1 - \lambda) \nabla V^* G \nabla W_2^T \phi + \frac{1}{2} \nabla V^* G \nabla \epsilon^T
\end{aligned}$$

[2] Results such as [105] could potentially be used to achieve an asymptotic convergence to the origin, but the additional feedback to eliminate the residual error would deviate from the optimal policy.

$$
\begin{aligned}
&+ \tilde{W}_{2c}^T \Gamma_2^{-1}(t) \nabla W_2 f - \frac{1}{2} \tilde{W}_{2c}^T \Gamma_2^{-1}(t) \nabla W_2 g R^{-1} g^T \left(\lambda \nabla \sigma^T + \nabla \lambda^T \sigma^T \right) \hat{W}_{1a} \\
&- \frac{1}{2} \tilde{W}_{2c}^T \Gamma_2^{-1}(t) \nabla W_2 g R^{-1} g^T \left((1-\lambda) \nabla \phi^T - \nabla \lambda^T \phi^T \right) \hat{W}_{2a} \\
&- \tilde{W}_{1a}^T \dot{\hat{W}}_{1a} - \tilde{W}_{2a}^T \dot{\hat{W}}_{2a} - \frac{1}{2} \tilde{W}_{1c}^T \Gamma_1^{-1} \dot{\Gamma}_1 \Gamma_1^{-1} \tilde{W}_{1c} \\
&+ \tilde{W}_{2a}^T \nabla W_2 f - \frac{1}{2} \tilde{W}_{2a}^T \nabla W_2 g R^{-1} g^T \left(\lambda \nabla \sigma^T + \nabla \lambda^T \sigma^T \right) \hat{W}_{1a} \\
&- \frac{1}{2} \tilde{W}_{2a}^T \nabla W_2 g R^{-1} g^T \left((1-\lambda) \nabla \phi^T \hat{W}_{2a} - \nabla \lambda^T \phi^T \right) \hat{W}_{2a} \\
&- \frac{1}{2} \tilde{W}_{2c}^T \Gamma_2^{-1} \dot{\Gamma}_2 \Gamma_2^{-1} \tilde{W}_{2c}.
\end{aligned}
$$

Substituting in the update laws (3–17), (3–18), (3–20), and the expressions for the R-MBRL update laws, then using Assumption 3.2 along with (3–24) and (3–25), segregating terms, completing the squares, and using Young's inequalities yields

$$
\begin{aligned}
\dot{V}_L \leq & -\underline{q} \|x\|^2 - \frac{(k_{a1}+k_{a2})}{4} \|\tilde{W}_{2a}\|^2 - \frac{(\eta_{a1}+\eta_{a2})}{4} \|\tilde{W}_{1a}\|^2 - \frac{k_{c2}\underline{b}}{4} \|\tilde{W}_{2c}\|^2 - \frac{\eta_{c2}\underline{c}}{4} \|\tilde{W}_{1c}\|^2 \\
& - \left(\frac{k_{c2}\underline{b}}{4} - \frac{\vartheta_2}{2\underline{\Gamma}_2} - \frac{\vartheta_{10}}{2} - \frac{k_{c1}(\vartheta_5+\vartheta_6+\vartheta_7)}{4\sqrt{\gamma_2}} \right) \|\tilde{W}_{2c}\|^2 - \left(\frac{k_{a1}+k_{a2}}{4} - B_1 \right) \|\tilde{W}_{2a}\|^2 \\
& - \left(\frac{\eta_{c2}\underline{c}}{4} - \frac{\eta_{c1}(\vartheta_5+\vartheta_6+\vartheta_7)}{4\sqrt{\gamma_1}} - \frac{\vartheta_{10}}{2} \right) \|\tilde{W}_{1c}\|^2 - \left(\frac{\eta_{a1}+\eta_{a2}}{4} - B_2 \right) \|\tilde{W}_{1a}\|^2 \\
& - \begin{bmatrix} \left\|\tilde{W}_{1a}\right\| & \left\|\tilde{W}_{1c}\right\| \end{bmatrix} \begin{bmatrix} \frac{(\eta_{a1}+\eta_{a2})}{4} & -B_3 \\ -B_3 & \frac{\eta_{c2}\underline{c}}{4} \end{bmatrix} \begin{bmatrix} \left\|\tilde{W}_{1a}\right\| \\ \left\|\tilde{W}_{1c}\right\| \end{bmatrix} \\
& - \begin{bmatrix} \left\|\tilde{W}_{2a}\right\| & \left\|\tilde{W}_{2c}\right\| \end{bmatrix} \begin{bmatrix} \frac{(k_{a1}+k_{a2})}{4} & -B_4 \\ -B_4 & \frac{k_{c2}\underline{b}}{4} \end{bmatrix} \begin{bmatrix} \left\|\tilde{W}_{2a}\right\| \\ \left\|\tilde{W}_{2c}\right\| \end{bmatrix} + \iota,
\end{aligned} \tag{3–30}
$$

where $\sqrt{\gamma}_1$ and $\sqrt{\gamma_2}$ result from the fact that $\left\| \frac{\omega_{\nabla\sigma}}{\rho_1} \right\| \leq \frac{1}{2\sqrt{\gamma_1}}$ and $\left\| \frac{\omega_{\nabla\phi}}{\rho_2} \right\| \leq \frac{1}{2\sqrt{\gamma_2}}$. In (3–30) the terms $\underline{c}$ and $\underline{b}$ are defined in (3–27), and the terms B_1, B_2, B_3, and B_4 are defined as

$$
\begin{aligned}
B_1 &\triangleq \vartheta_1 + \frac{\vartheta_4 \overline{\|W_2\|}}{2\sqrt{\gamma_2}} + \frac{\vartheta_2}{2} + \left(\frac{k_{c1}}{2\sqrt{\gamma_2}} + \frac{\eta_{c1}}{2\sqrt{\gamma_1}} \right) \vartheta_5 \|\tilde{W}_{2a}\|^2 + \frac{k_{c1}}{2\sqrt{\gamma_2}} \vartheta_6 \|\tilde{W}_{1a}\|^2, \\
B_2 &\triangleq \frac{\vartheta_3 \overline{W_1}}{\sqrt{\gamma_1}} + \frac{1}{2} \left(\frac{1}{\underline{\Gamma}_2} + 1 \right) \vartheta_2 + \left(\frac{\eta_{c1}}{2\sqrt{\gamma_1}} + \frac{k_{c1}}{2\sqrt{\gamma_2}} \right) \vartheta_7 \|\tilde{W}_{1a}\|^2 + \frac{\eta_{c1}}{2\sqrt{\gamma_1}} \vartheta_6 \|\tilde{W}_{2a}\|^2,
\end{aligned}
$$

$$B_3 \triangleq \frac{1}{2}\left(\eta_{a1} + \frac{\vartheta_3}{2\sqrt{\gamma_1}}\overline{W_1}\right),$$
$$B_4 \triangleq \frac{1}{2}\left(k_{a1} + \vartheta_9\right),$$

and the constants $\iota, \{\vartheta_i | i = 1, \ldots 12\} \in \mathbb{R}_{>0}$ are defined in Appendix A.1. Using (3–28) and (A–1)-(A–4) time derivative in (3–30) can be upper bounded as

$$\dot{V}_L \leq -\nu_l(\|Z\|), \quad \forall \|Z\| > \nu_l^{-1}(\iota). \tag{3–31}$$

Using (3–31), (A–5), and (3–26), Theorem 4.18 in [104] can be invoked to conclude that all trajectories $Z(t)$ that satisfy $\|Z(t_0)\| \leq \overline{\nu_l}^{-1}\left(\underline{\nu_l}(\zeta)\right)$, remain bounded for all $t \in \mathbb{R}_{\geq 0}$ and satisfy $\limsup_{t\to\infty} \|Z(t)\| \leq \underline{v_l}^{-1}\left(\overline{v_l}\left(v_l^{-1}(\iota)\right)\right)$. □

3.6 Simulation

3.6.1 Two-State Dynamical System

To demonstrate the performance of the developed ADP method for a nonlinear system with a known value function, simulation results for a two-state dynamical system are provided. The simulation is performed for the control affine system given in (3–1) where $x^o = [x_1^o, x_2^o]^T$,

$$f(x^o) = \begin{bmatrix} -x_1^o + x_2^o \\ -\frac{1}{2}x_1^o - \frac{1}{2}x_2^o\left(1 - (cos(2x_1^o) + 2)^2\right) \end{bmatrix}, \text{ and } g(x^o) = \begin{bmatrix} 0 \\ cos(2x_1^o) + 2 \end{bmatrix}. \tag{3–32}$$

The control objective is to minimize the cost functional in (3–2) with the instantaneous cost in (3–3) and the weighting matrices being $Q = I_2$ and $R = 1$. The optimal value function, $V^*(x^o)$, and optimal control policy, $u^*(x^o)$, for these particular dynamics and cost function are known to be $V^*(x^o) = \frac{1}{2}x_1^{o2} + x_2^{o2}$ and $u^*(x^o) = -\left(cos(2x_1^o) + 2\right)x_2^o$ respectively (cf. [64]). The regions A and A' are selected as circles around the origin such that $A = \overline{B_{1.5}(0)} \triangleq \{x^o : \|x^o\| \leq 1.5\}$ and $A' = \overline{B_{2.5}(0)} \triangleq \{x^o : \|x^o\| \leq 2.5\}$,

respectively. The transition function $\lambda(x^o)$ is selected to be (3–8) with $\ell = 1.0$ as discussed in Section 3.2.

To simulate the developed technique, the MBRL approach from [18] is used to learn the value function in A. The MBRL basis function vector for value function approximation in the set A' is selected as $\sigma(x^o) = [x_1^{o2}, x_1^o x_2^o, x_2^{o2}]^T$, with thirteen uniformly distributed points selected in A' for BE extrapolation. To approximate the value function in $B = \chi \setminus A$, the StaF basis function vector is selected as $\phi\big(x^o, c(x^o)\big) = \big[x^{oT} c_1(x^o), x^{oT} c_2(x^o), x^{oT} c_3(x^o)\big]^T$ where $c_i(x^o) = x^o + d_i$ for $i = 1, 2, 3$. The centers of the StaF kernels are selected as $d_1 = 0.25 \cdot [0, 1]^T$, $d_2 = 0.25 \cdot [-0.886, -0.5]^T$, and $d_2 = 0.25 \cdot [-0.886, 0.5]^T$. To ensure sufficient excitation in B, a single trajectory $x_j^o : \mathbb{R}_{t \geq t_0} \to \mathbb{R}^n$ is selected for BE extrapolation such that at each time instant t, $x_j^o(t)$ is selected at random from a uniform distribution over a $\nu(x^o(t)) \times \nu(x^o(t))$ square centered at the current state $x^o(t)$ where $\nu(x^o(t)) = \frac{x^{oT} x^o + 0.01}{1 + x^{oT} x^o}$. The initial conditions for the system at $t_0 = 0$ are

$$x(0) = [-10, 10]^T, \ \hat{W}_{1c}(0) = \hat{W}_{1a}(0) = 2 \times 1_3, \ \hat{W}_{2c}(0) = \hat{W}_{2a}(0) = 0.3 \times 1_3,$$
$$\Gamma_1(0) = 350 \times I_3, \ \Gamma_2(0) = 50 \times I_3.$$

The gains for the MBRL update laws are selected as

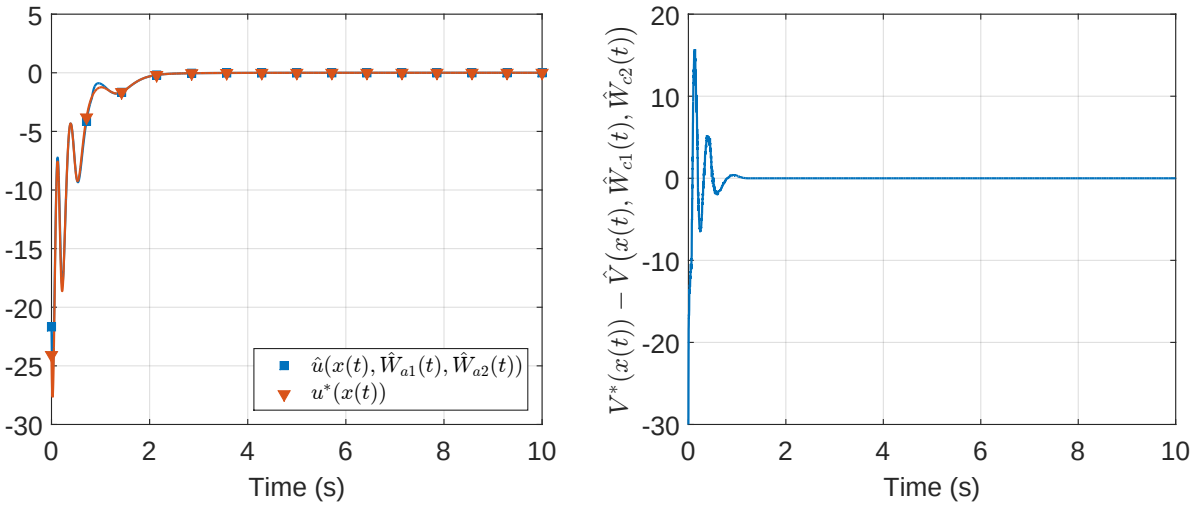
$$\eta_{c1} = 0.001, \ \eta_{c2} = 2, \ \eta_{a1} = 25, \ \eta_{a2} = 0.1, \ \beta_1 = 0.5, \ \gamma_1 = 2,$$

and the gains for the StaF update laws (3–17), (3–18), and (3–20) are selected as

$$k_{c1} = 0.001, \ k_{c2} = 0.09, \ k_{a1} = 1.5, \ k_{a2} = 0.01, \ \beta_1 = 0.003, \text{ and } \gamma_2 = 0.05.$$

Results

Figure 3-1a indicates that the control policy estimate converges to the optimal controller, while regulating the states to the origin, as seen in Figure 3-2. Figure 3-1b shows the value function approximation error, from which it is clear that the value



(a) The optimal control policy and estimate for the two-state system in (3–32).

(b) The value function estimation error for the two-state system in (3–32).

Figure 3-1. The control policies and value function estimation error for the two-state system in (3–32).

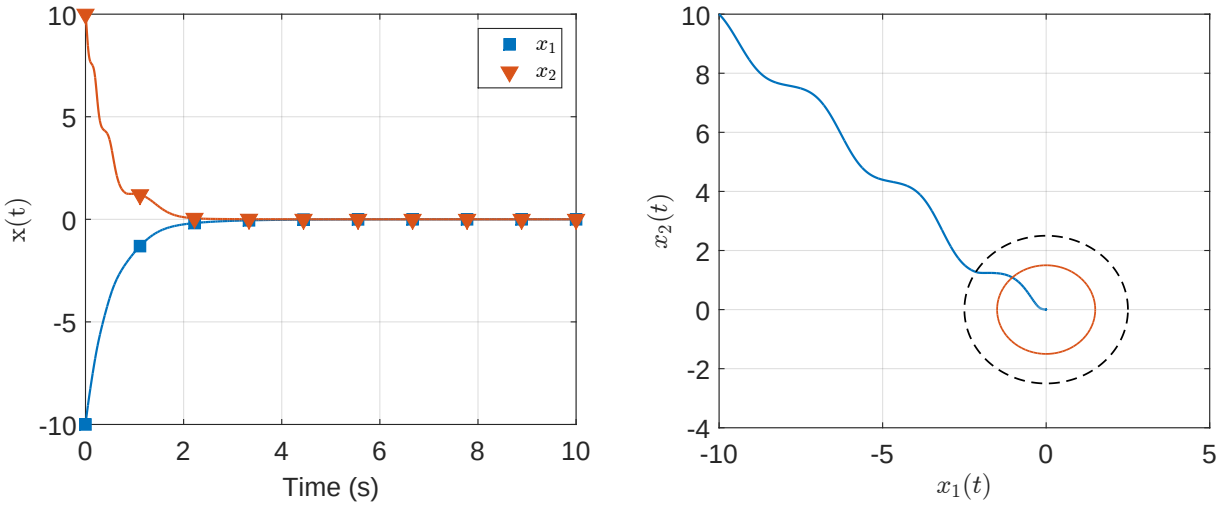
function estimate $\hat{V}$ converges to the optimal value function. Figure 3-3 shows that the estimated value function and policy weights for both the StaF (Figures 3-3b and 3-3d) and MBRL (Figures 3-3a and 3-3c) methods converge to steady-state values and remain bounded. The MBRL weights converge close to their optimal weights $W_1 = [0.5,\ 0,\ 1]^T$; however, the approximate StaF weights cannot be compared to their ideal weights because the optimal StaF weight are unknown.

3.6.2 Ten-State Dynamical System

To demonstrate the performance of the developed ADP method on a higher dimensional system, consider a centralized controller computing the control policies for a network of ten one-state dynamical systems where each system is in control affine form with dynamics represented as

$$f_i\left(x_i^o\right) = \left(\theta_{a,i} x_i^o + \theta_{b,i}\left(x_i^o\right)^2\right), \quad g_i\left(x_i^o\right) = \left(\cos\left(2x_i\right) + 2\right), \ \forall i = 1, \ldots, 10,$$

where $\theta_{a,i} = 2, 5, 0.1, 0.5, 2.5, 0.3, 0.5, 0.15, 3.5, 2$ and $\theta_{b,i} = 1, 0.5, , 1, 1, 1, 0.3, 1.1, 0.7, 0.9, 0.8$ for $i = 1, \ldots, 10$, respectively. The agent dynamics are combined to form one large



(a) State trajectory for the two-state system in (3–32). (b) Phase-space portrait for the two-state system in (3–32).

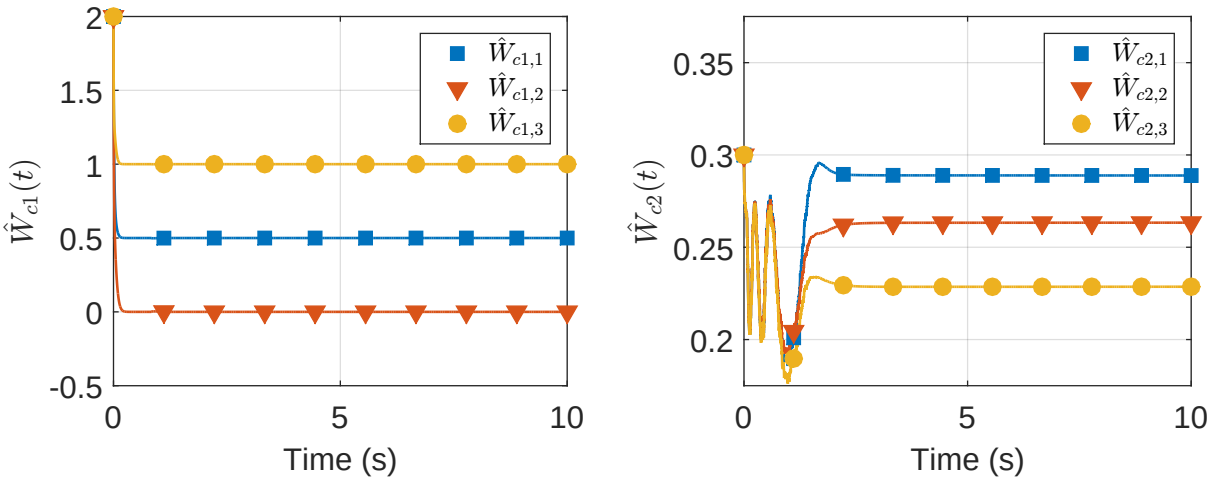
Figure 3-2. State regulation and state-space portrait for the two-state dynamical system. In Figure 3-2b, the region A' is the represented by the larger dashed circle while A is represented via the smaller circle.

dynamical system given by

$$f(x) = \begin{bmatrix} \theta_{a,1} x_1^o + \theta_{b,1} \left(x_1^o\right)^2 \\ \vdots \\ \theta_{a,10} x_6^o + \theta_{b,10} \left(x_6^o\right)^2 \end{bmatrix}, \; g(x) = \operatorname{diag}\left[\left(\cos\left(2x_1\right) + 2\right), \ldots, \left(\cos\left(2x_{10}\right) + 2\right)\right]. \tag{3–33}$$

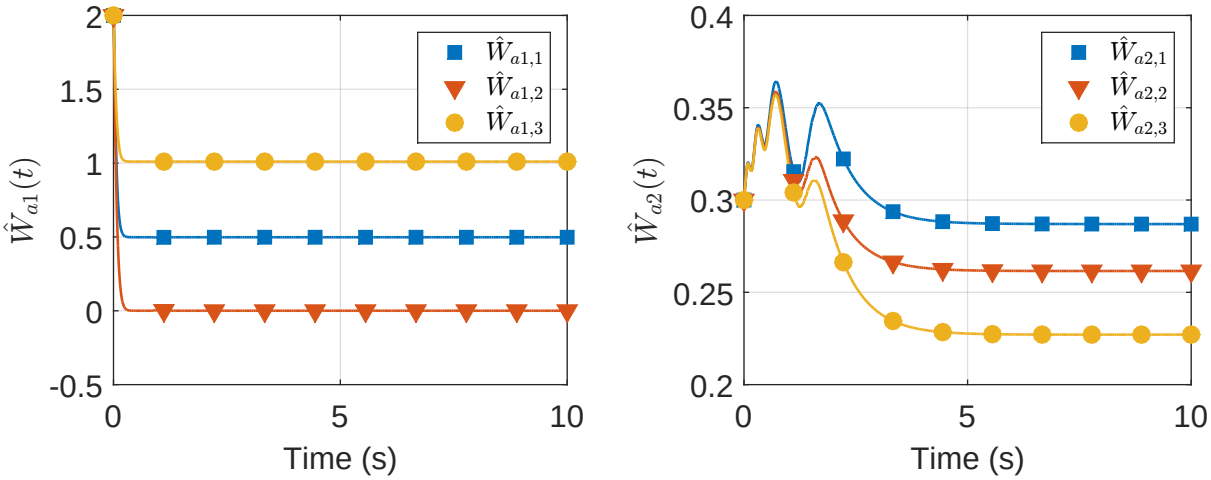
The transition function is selected to be the same as in (3–8) with $A = \overline{B_1(0)} \triangleq \{x^o : \|x^o\| \leq 1\}$ and $A' = \overline{B_2(0)} \triangleq \{x^o : \|x^o\| \leq 2\}$ and $\ell = 1.0$. The control objective is to minimize the cost functional in (3–2) with the instantaneous cost in (3–3) using the weighting matrices $Q = I_{10}$ and $R = I_{10}$.

The MBRL basis is selected to be a vector of twenty polynomials, and for BE extrapolation, twenty-one equally distributed points are selected in A'. The StaF basis is selected to be $\phi\left(x^o, c(x^o)\right) = \left[x^{oT} c_1(x^o), \ldots, x^{oT} c_{11}(x^o)\right]$, where $c_i\left(x^o\right) = x^o + d_i$ for $i = 1, \ldots, 11$. The centers d_i are selected to be the vertices of a 10-simplex. For BE extrapolation in B, a single point is selected at random from a uniform distribution over a



(a) R-MBRL critic approximations.

(b) StaF critic approximations.



(c) R-MBRL actor approximations.

(d) StaF actor approximations.

Figure 3-3. Value function and policy weight approximations for the two-state system in (3–32). The StaF actor and critic weights are updated using (3–17), (3–18), and (3–20). The R-MBRL actor and critic weights are updated using adaptation schemes which take a similar form to the StaF update laws, as discussed in Section 3.5.

$[2\nu\left(x^o\left(t\right)\right)]^{10}$ hypercube centered at the current state, where $\nu(x^o\left(t\right)) = \frac{0.0003 x^{oT} x^o}{1+0.5 x^{oT} x^o}$. When the states converge to A, the StaF update laws are turned off to reduce computational burden. The initial conditions for the system at $t_0 = 0$ are selected as

$$x\left(0\right) = \left[1.2, -0.3, 3, -2.4, -2.1, -2.7, -1.2, 1.2, 0.3, -1.8\right]^T,\ \hat{W}_{1c}\left(0\right) = \hat{W}_{1a}\left(0\right) = 5 \times 1_{20},$$
$$\hat{W}_{2c}\left(0\right) = \hat{W}_{2a}\left(0\right) = 0.25 \times 1_{11},\ \Gamma_1\left(0\right) = 350 \times I_{20},\ \Gamma_2\left(0\right) = 100 \times I_{11}.$$

The gains for the MBRL update laws are selected as

$$\eta_{c1} = 0.0005,\ \eta_{c2} = 30,\ \eta_{a1} = 25,\ \eta_{a2} = 0.01,\ \beta_1 = 0.06,\ \gamma_1 = 3,$$

and for the StaF update laws in (3–17), (3–18), and (3–20) the gains are selected as

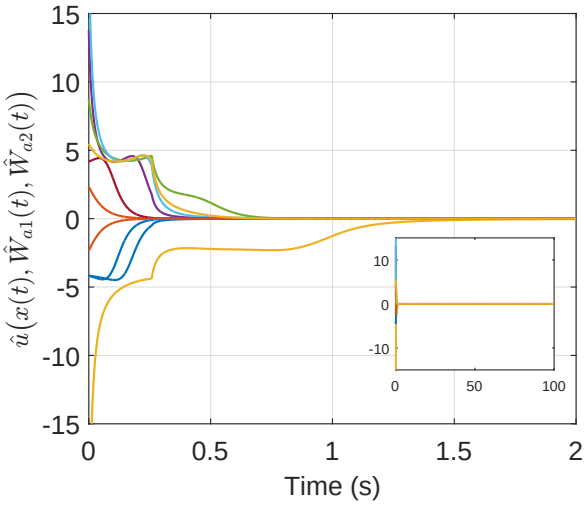
$$k_{c1} = 0.001,\ k_{c2} = 0.8,\ k_{a1} = 0.4,\ k_{a2} = 0.001,\ \beta_1 = 0.0001, \text{ and } \gamma_2 = 0.9.$$

Results

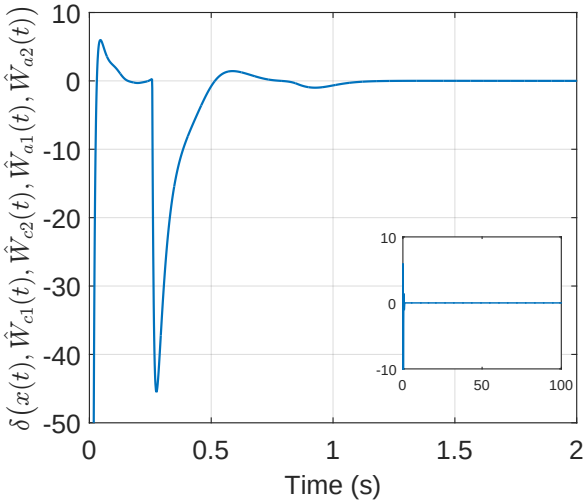
Figure 3-4a and Figure 3-4c show that the control policy and the system states converge to the origin. The oscillation-like effect between 0 and 1 seconds in Figure 3-4a comes from StaF approximation in B. Figure 3-4b indicates that the BE converges to zero. The transition of the BE between 0 and 1 second in Figure 3-4b is attributed to the transition of the value function approximation weight approximation as the state enters A'. Figure 3-5 shows that the approximate MBRL weights converge to steady-state values, and the StaF weights remain bounded.

3.6.3 Comparison

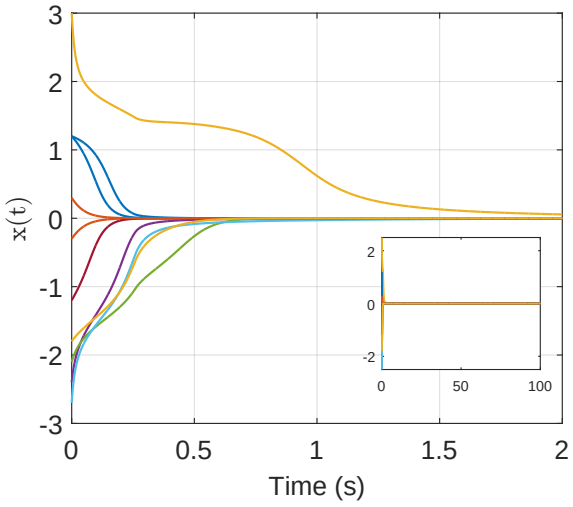
The developed technique is compared to the R-MBRL approximation technique in [18] and the StaF approximation technique in [1] via MATLAB® Simulink® running at 1000 Hz on an Intel® Core™ i5-2500K CPU at 3.30GHz. All systems are simulated for 100 seconds and the total cost, steady-state root-mean square (RMS) error, and running time are compared. The approximation method from [1] is implemented using polynomial StaF basis functions with centers at the vertices of an n-simplex for each



(a) Optimal control policy estimate for the ten-state dynamical system.

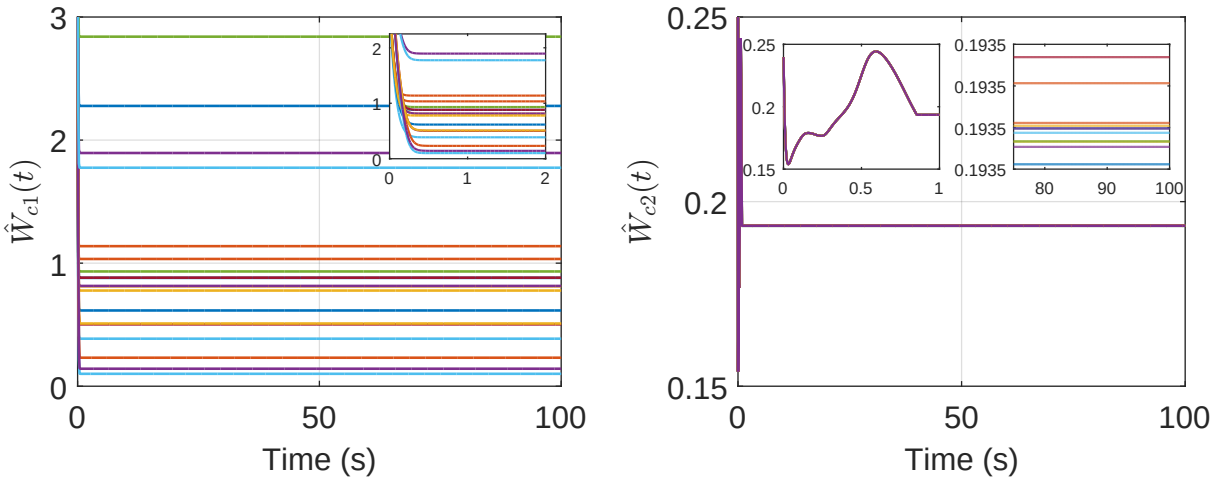


(b) Bellman Error using the developed method for a ten-state dynamical system.



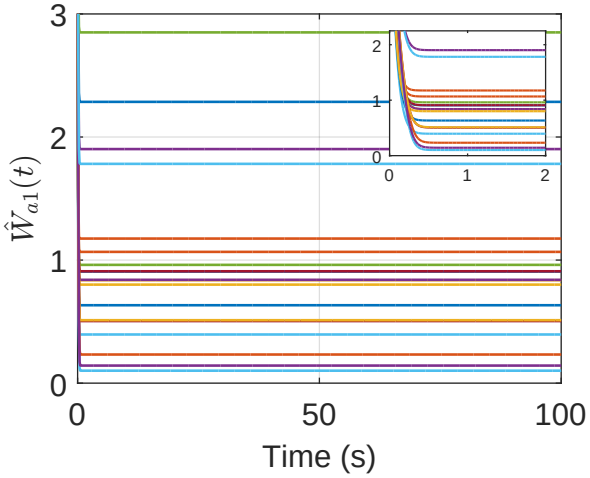
(c) The states for the ten-state dynamical system converge to the origin.

Figure 3-4. Control policy estimates, Bellman Error and states for the ten-state dynamical system in (3–33).

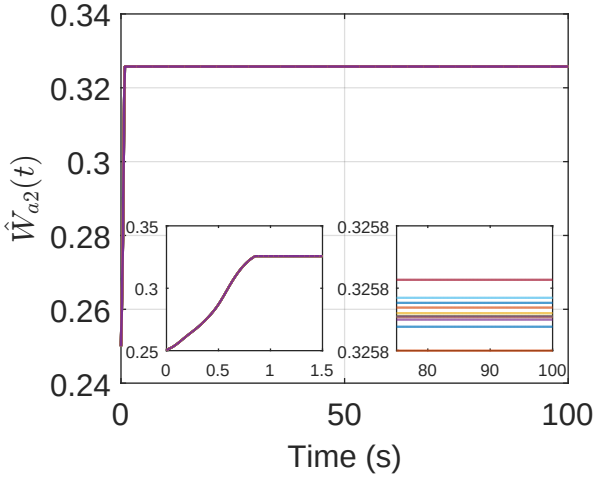


(a) R-MBRL critic approximations.

(b) StaF critic approximations.



(c) R-MBRL actor approximations.



(d) StaF actor approximations.

Figure 3-5. Value function and policy weight approximations using the R-MBRL and StaF critic and actor update laws for the ten-state dynamical system in (3–33).

Table 3-1. Simulation results. Steady-state RMS errors below 1×10^{-16} are considered to be zero.

(a) Two and three-state simulation results.

	Two-State Dynamical System			Three-State Dynamical System		
Controller	R-MBRL + StaF	StaF controller in [1]	R-MBRL controller in [18]	R-MBRL + StaF	StaF controller in [1]	R-MBRL controller in [18]
Total cost	150.55	150.62	150.50	24.85	25.57	24.55
RMS steady-state error	0	1.66×10^{-2}	0	0	1.19×10^{-4}	0
Running time (sec)	4.91	2.95	4.11	11.49	3.40	15.57

(b) Six and ten-state simulation results.

	Six-State Dynamical System			Ten-State Dynamical System		
Controller	R-MBRL + StaF	StaF controller in [1]	R-MBRL controller in [18]	R-MBRL + StaF	StaF controller in [1]	R-MBRL controller in [18]
Total cost	37.72	39.56	59.12	60.22	65.43	88.30
RMS steady-state error	0	1.81×10^{-8}	1.7×10^{-3}	0	2.76×10^{-10}	0
Running time (sec)	28.12	5.57	73.6	84.34	9.77	217.98

Table 3-2. Three-state simulation results with different sets A and A'. Steady-state RMS errors below 1×10^{-16} were considered to be zero.

	Three-State Dynamical System $x(0) = [-3, 3, -2.5]^T$				
Controller	R-MBRL + StaF $A = \{x^o : \|x^o\| \leq 0.25\}$ $A' = \{x^o : \|x^o\| \leq 1.25\}$	R-MBRL + StaF $A = \{x^o : \|x^o\| \leq 1.5\}$ $A' = \{x^o : \|x^o\| \leq 2.75\}$	R-MBRL + StaF $A = \{x^o : \|x^o\| \leq 3.0\}$ $A' = \{x^o : \|x^o\| \leq 4.5\}$	StaF controller in [1]	R-MBRL controller in [18]
Total cost	24.85	23.98	23.27	25.57	24.55
RMS steady-state error	0	0	0	1.19×10^{-4}	0
Running time (sec)	11.49	11.49	11.59	3.40	9.82

Table 3-3. Local cost when the system enters the set A for the developed method and the StaF-based method in [1].

	Two-State Dynamical System		Three-State Dynamical System		Six-State Dynamical System		Ten-State Dynamical System	
Controller	R-MBRL + StaF	StaF in [1]	R-MBRL + StaF	StaF in [1]	R-MBRL + StaF	StaF in [1]	R-MBRL + StaF	StaF in [1]
Total cost	150.55	150.62	24.85	25.57	37.72	39.56	60.22	65.43
Local cost	1.71	1.76	0.11	0.27	0.46	1.97	0.75	1.58

Table 3-4. Six-state simulation results with different sets A and A' under different initial conditions using the same gains for the update laws (3–17), (3–18), and (3–20). Steady-state RMS errors below 1×10^{-16} were considered to be zero.

	Six-State Dynamical System			
	$\|x(0)\| = 6.61$		$\|x(0)\| = 13.23$	
Controller	R-MBRL + StaF $A = \{x^o : \|x^o\| \leq 4\}$ $A' = \{x^o : \|x^o\| \leq 5\}$	StaF controller in [1]	R-MBRL + StaF $A = \{x^o : \|x^o\| \leq 10\}$ $A' = \{x^o : \|x^o\| \leq 10.5\}$	StaF controller in [1]
Total cost	21.59	27.78	83.55	111.57
Local cost	6.95	11.98	39.75	70.14
RMS steady-state error	0	1.61×10^{-9}	0	1.90×10^{-9}
Running time (sec)	26.46	5.35	26.86	5.16

n-dimensional problem. At a minimum $n + 1$ kernels need to be used with an n-dimensional system. The choice of kernel is only governed a few rules imposed by the StaF method, which can be found in [1, 32, 70]. Dot product kernels work well for the StaF application; examples include polynomial kernels and exponential kernels. The approximation method from [18] is implemented using polynomial basis functions selected via trial-and-error. Furthermore, the sets A and A' are selected via trial-and-error to demonstrate the effect of selecting different regions. The performance of the proposed method depends on the choice of A and A'. Hence, if the initial conditions are far from the origin then larger sets may be used, otherwise the sets A and A' should be smaller to provide enough time for the R-MBRL weights to be learned. It is seen that the developed technique converges similar to the R-MBRL technique in [18] but at a smaller cost and running time as the dimension of the system increases. In theory, the R-MBRL method should be closest to optimal because it provides an approximation over the entire operating domain. However, the choice of basis functions and the number of basis functions used for approximation has a major influence on the approximation. Hence, when the exact parameterization is known such as in the case of the two-state system, R-MBRL provides the smallest cost, but this is not necessarily true when the basis is not known a priori. The basis function used is directly correlated to the cost through the input; hence, basis functions with larger gradients will exhibit higher control efforts which can increase cost. An examination of the correlation between the type of basis function used and total cost for the R-MBRL method is out of the scope of this chapter. The increase in running time for the R-MBRL method in [18] for the six and ten-state systems occurs because the value function is approximated over the entire domain of operation instead of just a local region around the origin, requiring a large number of basis functions. The RMS error is practically zero since all of the methods provide a sufficiently accurate approximation of the value function, resulting in a stabilizing feedback.

The StaF-only approximation and the developed approximation technique results in a similar cost for the two, three, and six-state simulation when using difference gains. But for the ten-state simulation, the cost is smaller for the developed approximation technique. The StaF method in [1] also results in a slightly higher steady-state RMS error compared to the developed method. When increasing to a higher dimensional system such as the six and ten-state systems, the StaF method in [1] results in a much shorter running time when compared to the developed method because the developed method still requires stationary basis functions around the origin, which increases the running time.

In many applications such as station keeping of marine craft, the local cost or the cost which starts being calculated once the marine craft reaches a goal region is more important than the total cost for regulating to that region and staying there. Table 3-3 displays the local cost once the system enters the set A for the developed and StaF-based methods. The developed method results in a smaller local cost compared to the StaF method in [1]. Since the R-MBRL method contains a larger number of basis functions over A' compared to the StaF method, a better approximation over A is learned, resulting in a reduced local cost.

Table 3-4 provides a comparison of the developed method compared to the StaF method in [1] when the same gains are used and a large region A is selected with respect to the initial conditions. The results show that the StaF method has a smaller running time compared to the developed method; however, the developed method yields a lower cost compared to the StaF-only method. The developed method is capable of quickly learning the value function via BE extrapolation in the neighborhood A while the state still has not entered A.

Table 3-2 provides a comparison of the developed method with StaF and R-MBRL when the sets A and A' and the transition region $A' \setminus A$ are increased for the three-state dynamical system using different gains. When the sets get larger, a smaller total

cost results. The lower cost is because the R-MBRL method is approximating the value function over a larger area, and hence, provides a more accurate approximation compared to the local approximations of the StaF method. When the sets A and A' are increased, the developed method produces a smaller total cost compared to the R-MBRL method in [18], this is partially attributed to the fact that implementation of R-MBRL over a large region is challenging when an exact basis for value function approximation is not available. When the transition region $A' \setminus A$ increases, the gradient of λ decreases, possibly contributing to the smaller cost. Also in [18], the least-squares learning gain matrix $\Gamma_1(t)$ was updated without using recorded data, while the developed R-MBRL update law similar to (3–18) includes recorded data to improve the selection of $\Gamma_1(t)$.

The results in Tables 3-1-3-4 indicate that the optimal choice of the approximation method depends on the circumstance, and several advantages and disadvantages need to be taken into consideration when selecting which method to use. The StaF method is best suited for a high dimensional application requiring real-time performance where global optimality is not required. However, Table 3-4 shows that there are circumstances in which the developed method outperforms the StaF method in [1] in terms of total and local cost. Moreover, since the StaF method in [1] lacks memory, the weights need to be relearned every time the system passes through the predefined area of interest in the operating domain, whereas the developed uses the R-MBRL method to learn to static weights in that region and doesn't need to relearn the weights when the system leaves the neighborhood. The R-MBRL method in [18] is best suited for lower dimensional applications where global optimality is a premium. However, approximating the value function over the entire state-space requires a large number of basis functions, and hence, a large computational burden. Since the developed method reduces the area of interest, which reduces the number of basis functions required, it is computationally efficient when compared to R-MBRL in [18]. Applications with large operating domains

may benefit from the developed method since the value function can be learned in desired areas of the state-space, e.g., around the origin, independent of where the state is, using R-MBRL, while StaF keeps the system stable by approximating the value function around the state trajectory. Although the developed method shows a slight improvement over [1] in terms of cost and RMS error, more tuning parameters and an overall larger number of unknown parameters are required; hence, increasing the computational complexity of the tuning process and the computations.

3.7 Concluding Remarks

An infinite horizon optimal control problem was solved using a novel approximation methodology utilizing the StaF kernel method and a R-MBRL method. The operating domain, χ, of the system was segregated into two parts; a neighborhood, $A \subset \chi$, containing the origin where R-MBRL was employed, and the set $B = \chi \setminus A$ where the StaF method was employed. For a state initialized in B, the StaF method ensured stable and computationally efficient operation while a R-MBRL method achieved a sufficiently accurate estimate of the value function over the set A. When the state entered A, the R-MBRL technique was used to regulate the state to the origin.

Under specific conditions, Theorem 3.1 established that the developed control strategy results in uniform ultimate boundedness of the state trajectory. Simulation examples for two, three, six, and ten-state dynamical systems showed that the developed approximation method outperforms previous methods. When the dimension of the system increases, the developed method is able to estimate the value function sufficiently to reduce the local cost and the RMS error. Motivated by the computational efficiency of the StaF approximation method, the following chapter uses the local approximation of the StaF method to develop an online approximately optimal path planning strategy for an agent which encounters uncertain dynamic obstacles.

CHAPTER 4
APPROXIMATE OPTIMAL PATH-PLANNING TO AVOID UNKNOWN MOVING AVOIDANCE REGIONS

In this chapter, an infinite-horizon optimal regulation problem is considered for a control-affine nonlinear autonomous agent subject to input constraints in the presence of dynamic avoidance regions. StaF is implemented to approximate the value function in a local neighborhood of the agent. By performing local approximations, prior knowledge of the locations of avoidance regions is not required. To alleviate the *a priori* knowledge of the number of avoidance regions in the operating domain, an extension is provided that modifies the value function approximation. The developed feedback-based path-planning strategy guarantees uniformly ultimately bounded convergence of the approximated control policy to the optimal policy while also ensuring the agent remains outside avoidance regions. Results from three experiments are presented to illustrate the performance of the developed method, where a quadcopter achieves approximate optimal regulation while avoiding three mobile obstacles. To demonstrate the developed method, known avoidance regions are used in the first experiment, unknown avoidance regions are used in the second experiment, and an unknown time-varying obstacle directed by a remote pilot is included in the third experiment.

4.1 Problem Formulation

Consider an autonomous agent with control-affine nonlinear dynamics given by

$$\dot{x}(t) = f(x(t)) + g(x(t))\,u(t), \tag{4–1}$$

for all $t \in \mathbb{R}_{\geq t_0}$, where $x : \mathbb{R}_{\geq t_0} \to \mathbb{R}^n$ denotes the state, $f : \mathbb{R}^n \to \mathbb{R}^n$ denotes the drift dynamics, $g : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ denotes the control effectiveness, $u : \mathbb{R}_{t\geq t_0} \to \mathbb{R}^m$ denotes the control input, and $t_0 \in \mathbb{R}_{\geq 0}$ denotes the initial time. In addition, consider dynamic avoidance regions with nonlinear dynamics given by

$$\dot{z}_i(t) = h_i(z_i(t)), \tag{4–2}$$

for all $t \in \mathbb{R}_{\geq t_0}$, where $z_i : \mathbb{R}_{t \geq t_0} \to \mathbb{R}^n$ denotes the state of the center of the i^{th} avoidance region and $h_i : \mathbb{R}^n \to \mathbb{R}^n$ denotes the drift dynamics for the i^{th} zone in $\mathcal{M} \triangleq \{1, 2, \ldots, M\}$, where $\mathcal{M}$ is the set of avoidance regions in the state space $\mathbb{R}^n$.[1]

Remark 4.1. The dynamics in (4–2) are modeled as autonomous and isolated systems to facilitate the control problem formulation. Section 4.5 provides an extension to alleviate the need for Assumption 4.1.

The representation of the dynamics in (4–2) would require that the HJB in (4–10) have complete knowledge of the dynamics over the entire operating domain. However, motivated by real systems where agents may only have local sensing, it is desired to only consider the zone inside a detection radius. Therefore, to alleviate the need for the HJB to require knowledge of the avoidance region dynamics outside of the agents' ability to sense the obstacles, the avoidance regions are represented as

$$\dot{z}_i(t) = \mathscr{F}_i(x(t), z_i(t)) h_i(z_i(t)), \tag{4–3}$$

for all $t \in \mathbb{R}_{\geq t_0}$. Hence, all the terms in the HJB in (4–10), associated with the avoidance region, are zero when the avoidance regions are outside of the sensing abilities of the agent. In (4–3), $\mathscr{F}_i : \mathbb{R}^n \times \mathbb{R}^n \to [0, 1]$ is a smooth transition function that satisfies $\mathscr{F}_i(x, z_i) = 0$ for $\|x - z_i\| > r_d$ and $\mathscr{F}_i(x, z_i) = 1$ for $\|x - z_i\| \leq \bar{r}$, where $r_d \in \mathbb{R}_{>0}$ denotes the detection radius of the system in (4–1), and $\bar{r} \in (r_a, r_d)$ where $r_a \in \mathbb{R}_{>0}$ denotes radius of the avoidance region. Hence, from the agent's perspective, the dynamics of the obstacles do not affect the agent outside of the sensing radius.

Remark 4.2. In application, a standard practice is to enforce a minimum avoidance radius to ensure safety [28, 29]. In addition, the detection radius, r_d, and avoidance

[1] The terms avoidance regions and obstacles are used interchangeably.

radius, r_s, depends on the system parameters such as the maximum agent velocity limits.

Assumption 4.1. The number of dynamic avoidance regions M is known; however, the locations of the states of each region is unknown until it is within the sensing radius of the agent.[2]

Assumption 4.2. The drift dynamics f, h_i, and control effectiveness g are locally Lipschitz continuous, and g is bounded such that $0 < \|g(x(t))\| \leq \overline{g}$ for all $x \in \mathbb{R}^n$ and all $t \in \mathbb{R}_{\geq t_0}$ where $\overline{g} \in \mathbb{R}_{>0}$. Furthermore, $f(0) = 0$, and $\nabla f : \mathbb{R}^n \to \mathbb{R}^n \times \mathbb{R}^n$ is continuous.

Assumption 4.3. The equilibrium points z_i^e for the obstacles given by the dynamics in (4–3) lie outside of a ball of radius r_d centered at the origin. That is, the origin is sufficiently clear of obstacles. Furthermore, obstacles do not trap the agent, meaning the obstacles do not completely barricade the agent and there are no deadlocks. Moreover, the agent is assumed to be sufficiently agile to be able to outmaneuver the moving obstacles. Specifically, the obstacle velocities must be appropriately equal or less than the agent for the agent to have capability to avoid the obstacle in general.

Remark 4.3. To facilitate the development, the centers of the avoidance regions, shown in Figure 4-1, are augmented with the following:

- The total detection set is defined as $\mathcal{D} = \cup_{i \in \mathcal{M}} \mathcal{D}_i$, where
$$\mathcal{D}_i = \{x \in \mathbb{R}^n \mid \|x - z_i\| \leq r_d\}.$$
- The total conflict set is defined as $\mathcal{W} = \cup_{i \in \mathcal{M}} \mathcal{W}_i$, where
$$\mathcal{W}_i = \{x \in \mathbb{R}^n \mid r_a < \|x - z_i\| \leq \overline{r}\}.$$
- The total avoidance set is $\Omega = \cup_{i \in \mathcal{M}} \Omega_i$, where each local avoidance region is
$$\Omega_i = \{x \in \mathbb{R}^n \mid \|x - z_i\| \leq r_a\}.$$

[2] Section 4.5 presents an approach to alleviate Assumption 4.1.

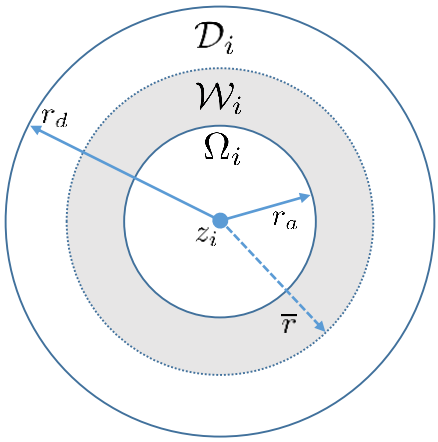


Figure 4-1. Augmented regions around each avoidance region.

Furthermore, the avoidance region and agent dynamics can be combined to form the following the system

$$\dot{\zeta}(t) = F(\zeta(t)) + G(\zeta(t))\, u(t), \tag{4–4}$$

for all $t \in \mathbb{R}_{\geq t_0}$, where $\zeta = \left[x^T, z_1^T, \ldots, z_M^T\right]^T \in \mathbb{R}^{\mathcal{N}}$, $\mathcal{N} = (M+1)\, n$ and

$$F(\zeta) = \begin{bmatrix} f(x) \\ \mathscr{F}_1(x, z_1)\, h_1(z_1) \\ \vdots \\ \mathscr{F}_M(x, z_M)\, h_M(z_M) \end{bmatrix}, G(\zeta) = \begin{bmatrix} g(x) \\ 0_{Mn \times m} \end{bmatrix}.$$

The goal is to simultaneously design and implement a controller u which minimizes the cost function

$$J(\zeta, u) \triangleq \int_{t_0}^{\infty} r(\zeta(\tau), u(\tau))\, d\tau, \tag{4–5}$$

subject to (4–4) while obeying $\sup_t(u_i) \leq \mu_{sat}\ \forall i = 1, \ldots, m$, where $\mu_{sat} \in \mathbb{R}_{>0}$ is the control effort saturation limit. In (4–5), $r : \mathbb{R}^{\mathcal{N}} \times \mathbb{R}^m \to [0, \infty]$ is the instantaneous cost defined as

$$r(\zeta, u) = Q_x(x) + \sum_{i=1}^{M} s_i(x, z_i)\, Q_z(z_i) + \Psi(u) + P(\zeta), \tag{4–6}$$

where $Q_x,\ Q_z : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ are user-defined PD functions that penalize the agent and obstacle states. The smooth scheduling function $s_i : \mathbb{R}^n \times \mathbb{R}^n \to [0,1]$, which allows the avoidance region states in the detection radius to be penalized, satisfies $s_i = 0$ for $\|x - z_i\| > r_d$ and $s_i = 1$ for $\|x - z_i\| \leq \bar{r}$. In (4–6), $\Psi : \mathbb{R}^m \to \mathbb{R}$ is a PD function penalizing the control input u, defined as

$$\Psi(u) \triangleq 2\sum_{i=1}^{m}\left[\int_0^{u_i}\left(\mu_{sat} r_i \tanh^{-1}\left(\frac{\xi_{u_i}}{\mu_{sat}}\right)\right) d\xi_{u_i}\right], \tag{4–7}$$

where u_i is the i^{th} element of the control u, ξ_{u_i} is an integration variable, and r_i are the diagonal elements which make up the symmetric PD weighting matrix $R \in \mathbb{R}^{m\times m}$ where $R \triangleq \operatorname{diag}\left\{\underline{R}\right\}$, and $\underline{R} \triangleq [r_1, \ldots, r_m] \in \mathbb{R}^{1\times m}$ [17]. The function $P : \mathbb{R}^{\mathcal{N}} \to \mathbb{R}$ in (4–6), called the avoidance penalty function, is a positive semi-definite compactly supported function defined as

$$P(\zeta) \triangleq \sum_{i=1}^{M}\left(\min\left\{0, \frac{\|x - z_i\|^2 - r_d^2}{\left(\|x - z_i\|^2 - r_a^2\right)^2}\right\}\right)^2. \tag{4–8}$$

Remark 4.4. The avoidance penalty function in (4–8) is zero outside of the compact set $\mathcal{D}$, and yields an infinite penalty when $\|x - z_i\| = r_a$ for any $i \in \mathcal{M}$. Other penalty/avoidance functions can be used; see [31] for a generalization of avoidance functions. The avoidance penalty function in (4–8) modifies that found in [31], which studies a generalization of avoidance penalty functions. Since the term in the denominator has quartic growth compared to only quadratic growth, the function in (4–8) is scaled differently compared to that found in [31]. Other growth factors can also be used which affects the rate at which the agent penalizes the avoidance regions once it detects them.

Assumption 4.4. There exist constants $\underline{q}_x,\ \overline{q}_x,\ \underline{q}_z,\ \overline{q}_z \in \mathbb{R}_{>0}$ such that $\underline{q}_x \|x\|^2 \leq Q_x(x) \leq \overline{q}_x \|x\|^2$ for all $x \in \mathbb{R}^n$, and $\underline{q}_z \|z_i\|^2 \leq Q_z(z_i) \leq \overline{q}_z \|z_i\|^2$ for all $z_i \in \mathbb{R}^n$ and $i \in \mathcal{M}$.

The infinite-horizon scalar value function for the optimal value function, denoted by $V^* : \mathbb{R}^{\mathcal{N}} \to \mathbb{R}_{\geq 0}$, is expressed as

$$V^*(\zeta) = \min_{u(\tau)\in U|\tau\in\mathbb{R}_{\geq t}} \int_t^\infty r(\zeta(\tau), u(\tau))\, d\tau, \tag{4–9}$$

where $U \subset \mathbb{R}^m$ denotes the set of admissible inputs. For the stationary solution, the HJB equation, which characterizes the optimal value function is given by

$$\begin{aligned} 0 &= \frac{\partial V^*(\zeta)}{\partial \zeta}\left(F(\zeta) + G(\zeta)\, u^*(\zeta)\right) + r(\zeta, u^*(\zeta)), \\ &= \frac{\partial V^*(\zeta)}{\partial x}\left(f(x) + g(x)\, u^*(\zeta)\right) + \sum_{i=1}^{M} \frac{\partial V^*(\zeta)}{\partial z_i}\left(\mathscr{F}_i(x, z_i)\, h_i(z_i)\right) + r(\zeta, u^*(\zeta)) \end{aligned} \tag{4–10}$$

with the condition $V^*(0) = 0$, where $u^* : \mathbb{R}^{\mathcal{N}} \to \mathbb{R}^m$ is the optimal control policy, which is determined from (4–10) as

$$u^*(\zeta) = -\mu_{sat} \operatorname{Tanh}\left(\frac{R^{-1}G(\zeta)^T}{2\mu_{sat}}\left(\nabla V^*(\zeta)\right)^T\right). \tag{4–11}$$

Remark 4.5. The following Lyapunov-based stability analysis indicates that the states $\zeta(t)$ remain outside of Ω, i.e. $\zeta(t) \notin \Omega$. Hence, the gradient is never taken over the discontinuity.

The HJB in (4–10) uses both the agent and avoidance region dynamics. However, because each avoidance region is modeled as (4–3), the terms that include them are zero when the regions are not detected. Furthermore, the analytical expression in (4–11) requires knowledge of the optimal value function. However, the analytical solution for the HJB, i.e. the value function, is not feasible to compute in general cases. Therefore, an approximation is sought using a NN approach.

4.2 Value Function Approximation

Recent developments in ADP have resulted in computationally efficient StaF kernels to approximate the value function [1]. To facilitate the development let $\chi \subset \mathbb{R}^{\mathcal{N}}$ be a compact set, with x and all z_i in the interior of χ. Based on the StaF method in [1]

and [95], after adding and subtracting a bounded avoidance function $P_a(\zeta)$, the optimal value function and controller can be approximated as

$$V^*(y) = P_a(y) + W(y)^T \sigma(y, c(\zeta)) + \epsilon(\zeta, y) \tag{4–12}$$

$$u^*(y) = -\mu_{sat} \operatorname{Tanh}\left(\frac{R^{-1} G(y)^T}{2\mu_{sat}}\left(\nabla P_a(y)^T + \nabla\sigma(y, c(\zeta))^T W(\zeta) + \nabla W(\zeta)^T \sigma(y, c(\zeta)) + \nabla\epsilon(y, \zeta)^T\right)\right), \tag{4–13}$$

where $c(\zeta) \in \left(\overline{B_r(\zeta)}\right)^L$ are centers around the current concatenated state ζ, $L \in Z_{>0}$ is the number of centers, and $y \in \overline{B_r(\zeta)}$ where $\overline{B_r(\zeta)}$ is a small compact set around the current state $\zeta \in \chi$. In (4–12), $W : \chi \to \mathbb{R}^L$ is the continuously differentiable ideal StaF weight function which changes with the state dependent centers, $\epsilon : \chi \to \mathbb{R}$ is the continuously differentiable bounded function reconstruction error, and $\sigma : \chi \to \mathbb{R}^L$ is a concatenated vector of StaF basis functions such that

$$\sigma(\zeta, c(\zeta)) = \begin{bmatrix} \sigma_0(x, c_0(x)) \\ s_1(x, z_1)\,\sigma_1(z_1, c_1(z_1)) \\ \vdots \\ s_M(x, z_M)\,\sigma_M(z_M, c_M(z_M)) \end{bmatrix}, \tag{4–14}$$

where $\sigma_0(x, c_0(x)) : \mathbb{R}^n \to \mathbb{R}^{P_x}$ and $\sigma_i(z_i, c_i(z_i)) : \mathbb{R}^n \to \mathbb{R}^{P_{z_i}}$ for $i \in \mathcal{M}$ are strictly positive definite, continuously differentiable StaF kernel function vectors, and $c_i : \mathbb{R}^n \to \mathbb{R}^n$ for $i \in \{0, 1, \ldots, M\}$ are state-dependent centers.[3] The formation of the vector of basis functions in (4–14) allows for certain weights of the approximation to be constant when the agent and no-entry zones are not in the detection regions. This formulation introduces a sparse-like approach because the basis functions which

[3] The dimension of the concatenated vector of StaF basis functions σ is $L = P_x + \sum_{i=1}^{M} P_{z_i}$.

correlate to the no-entry zones are off due to the scheduling function s_i, when they are outside of the detection regions. Hence, approximation of the value function is only influenced by the no-entry zones when they are in the detection regions $\mathcal{D}_i$.

Remark 4.6. Unlike the function P, which is not finite when $\|x - z_i\| = r_a$, for any $i \in \mathcal{M}$, the function P_a satisfies $P_a = 0$ when $x,\ z_i \notin \mathcal{D}_i$ for each $i \in \mathcal{M}$, and for all $0 \leq P(\zeta) \leq \overline{P}_a$, and $\|\nabla P_a(\zeta)\| \leq \overline{\|\nabla P_a\|}$ for all $\zeta \in \mathbb{R}^{\mathcal{N}}$. An example of $P_a(\zeta)$ includes $P_a(\zeta) \triangleq \sum_{i=1}^{M} P_{a,i}(x, z_i)$ where $P_{a,i} \triangleq \left(\min\left\{0, \frac{\|x-z_i\|^2 - r_d^2}{\left(\|x-z_i\|^2 - r_a^2\right)^2 + r_\varepsilon}\right\}\right)^2$ for $r_\varepsilon \in \mathbb{R}_{>0}$, or see [28–30] for other examples of bounded avoidance functions.

The optimal value function and controller are not known in general; therefore, approximations $\hat{V} : \mathbb{R}^{\mathcal{N}} \times \mathbb{R}^{\mathcal{N}} \times \mathbb{R}^L \to \mathbb{R}$ and $\hat{u} : \mathbb{R}^{\mathcal{N}} \times \mathbb{R}^{\mathcal{N}} \times \mathbb{R}^L \to \mathbb{R}^m$ are used where

$$\hat{V}\left(y, \zeta, \hat{W}_c\right) \triangleq P_a(y) + \hat{W}_c^T \sigma\left(y, c(\zeta)\right), \tag{4–15}$$

$$\hat{u}\left(y, \zeta, \hat{W}_a\right) \triangleq -\mu_{sat} \operatorname{Tanh}\left(\frac{R^{-1} G(y)^T}{2\mu_{sat}} \left(\nabla \sigma\left(y, c(\zeta)\right)^T \hat{W}_a + \nabla P_a^T(y)\right)\right). \tag{4–16}$$

In (4–15) and (4–16), $\hat{V}$ and $\hat{u}$ are evaluated at a point $y \in B_r(\zeta)$ using StaF kernels centered at ζ, while $\hat{W}_c,\ \hat{W}_a \in \mathbb{R}^L$ are the weight estimates for the ideal weight vector W. In actor-critic architectures, the estimates $\hat{V}$ and $\hat{u}$ replace the optimal value function V^* and optimal policy u^* in (4–10) to form the residual BE $\delta : \mathbb{R}^{\mathcal{N}} \times \mathbb{R}^{\mathcal{N}} \times \mathbb{R}^L \times \mathbb{R}^L \to \mathbb{R}$ defined as

$$\delta\left(y, \zeta, \hat{W}_c, \hat{W}_a\right) \triangleq \nabla \hat{V}\left(y, \zeta, \hat{W}_c\right)\left(F(y) + G(y)\,\hat{u}\left(y, \zeta, \hat{W}_a\right)\right) + r\left(y, \hat{u}\left(y, \zeta, \hat{W}_a\right)\right). \tag{4–17}$$

The aim of the actor and critic is to find a set of weights which minimize the BE for all $\zeta \in \mathbb{R}^{\mathcal{N}}$.

4.3 Online Learning

To implement the approximations online, at a given time instance t, the BE $\delta_t : \mathbb{R}_{\geq 0} \to \mathbb{R}$ is evaluated as

$$\delta_t(t) \triangleq \delta\big(\zeta(t), \zeta(t), \hat{W}_c(t), \hat{W}_a(t)\big), \tag{4–18}$$

where ζ denotes the state of the system in (4–4) starting at initial time t_0 with initial condition ζ_0, while $\hat{W}_c(t)$ and $\hat{W}_a(t)$ denote the critic weight and actor weight estimates at time t, respectively. The controller which influences the state $x(t) \subset \zeta(t)$ is

$$u(t) = \hat{u}\big(\zeta(t), \zeta(t), \hat{W}_a(t)\big). \tag{4–19}$$

Simulation of experience is used to learn online by extrapolating the BE to unexplored areas of the state space [1, 18]. Off-policy trajectories $\{x_k : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \to \mathbb{R}^n\}_{k=1}^N$ are selected by the critic such that each x_k maps the current state $x(t)$ to a point $x_k(x(t), t) \in B_r(x(t))$.

Remark 4.7. Rather than extrapolating the entire state vector of the system, as designed in [1, 18, 96], only the controlled states, i.e. the agent's states, are extrapolated to perform simulation of experience. Compared to experience replay results such as [17], which record a history stack of prior input-output pairs, the simulation of experience approach in this result only uses extrapolated states within a time-varying neighborhood of the current agent state. This is motivated by the StaF approximation method, which only provides a sufficient approximation of the value function a neighborhood of the current agent state.

The extrapolated BE $\delta_k : \mathbb{R}_{\geq 0} \to \mathbb{R}$ for each ζ_k takes the form

$$\delta_k(t) = \hat{W}_c^T(t)\,\omega_k(t) + \omega_{Pk}(t) + r\left(\zeta_k(t), \hat{u}_k(t)\right) \tag{4–20}$$

where $\zeta_k = \begin{bmatrix} x_k^T, & Z(t) \end{bmatrix}^T$,

$$\omega_{Pk}(t) \triangleq \nabla P_a(\zeta_k(t)) \left(F(\zeta_k(t)) + G(\zeta_k(t)) \hat{u}\left(\zeta_k(t), \zeta(t), \hat{W}_a(t)\right) \right),$$

$$\omega_k(t) \triangleq \nabla \sigma(\zeta_k(t), c(\zeta(t))) \left(F(\zeta_k(t)) + G(\zeta_k(t)) \hat{u}\left(\zeta_k(t), \zeta(t), \hat{W}_a(t)\right) \right),$$

and the extrapolated policies are

$$\hat{u}_k(t) \triangleq -\mu_{sat} \operatorname{Tanh}\left(\frac{R^{-1} G(\zeta_k(t))}{2\mu_{sat}} \left(\nabla \sigma\big(\zeta_k(t), c(\zeta(t))\big)^T \hat{W}_a(t) + \nabla P_a^T(\zeta_k(t)) \right) \right). \tag{4–21}$$

The concurrent learning-based least squares update laws are designed as

$$\dot{\hat{W}}_c(t) = -\Gamma(t) \left(\frac{k_{c1} \omega(t)}{\rho(t)} \delta(t) + \frac{k_{c2}}{N} \sum_{k=1}^{N} \frac{\omega_k(t)}{\rho_k(t)} \delta_k(t) \right) \tag{4–22}$$

$$\dot{\Gamma}(t) = \beta \Gamma(t) - k_{c1} \Gamma(t) \frac{\omega(t) \omega^T(t)}{\rho^2(t)} \Gamma(t) - \frac{k_{c2}}{N} \Gamma(t) \sum_{k=1}^{N} \frac{\omega_k(t) \omega_k^T(t)}{\rho_k^2(t)} \Gamma(t), \tag{4–23}$$

with $\Gamma(t_0) = \Gamma_0$. Furthermore, in (4–22) and (4–23) $\rho(t) \triangleq 1 + \gamma_1 \omega(t)^T \omega(t)$, $\rho_k(t) \triangleq 1 + \gamma_1 \omega_k(t)^T \omega_k(t)$ are normalizing factors, $k_{c1}, k_{c2}, \gamma_1 \in \mathbb{R}_{>0}$ are adaptation gains, $\beta \in \mathbb{R}_{>0}$ is a forgetting factor, and

$$\omega(t) \triangleq \nabla \sigma\big(\zeta(t), c(\zeta(t))\big) \Big(F(\zeta(t)) + G(\zeta(t)) \hat{u}\big(\zeta(t), \zeta(t), \hat{W}_a(t)\big) \Big).$$

The policy weights are updated to follow the critic weights using the actor update law designed as

$$\dot{\hat{W}}_a(t) = -\Gamma_a \Bigg(k_{a1} \left(\hat{W}_a(t) - \hat{W}_c(t) \right) + k_{a2} \hat{W}_a(t)$$

$$+ k_{c1} G_{a1}(t) \frac{\omega^T(t)}{\rho(t)} \hat{W}_c(t) + \frac{k_{c2}}{N} \sum_{k=1}^{N} G_{a1,k}(t) \frac{\omega_k^T(t)}{\rho_k(t)} \hat{W}_c(t) \Bigg) \tag{4–24}$$

where $k_{a1}, k_{a2} \in \mathbb{R}_{>0}$ are adaptation gains, $\Gamma_a \in \mathbb{R}^{L \times L}$ is a PD constant matrix, and

$$G_{a1}(t) \triangleq \mu_{sat} \nabla\sigma(\zeta(t), c(\zeta(t))) G(\zeta(t)) \times \left(\text{Tanh}\left(\frac{1}{k_u} \hat{\bar{D}}(t) \right) - \text{Tanh}\left(\frac{R^{-1}}{2\mu_{sat}} \hat{\bar{D}}(t) \right) \right),$$

$$G_{a1,k}(t) \triangleq \mu_{sat} \nabla\sigma(\zeta_k(t), c(\zeta(t))) G(\zeta_k(t)) \times \left(\text{Tanh}\left(\frac{1}{k_u} \hat{\bar{D}}_k(t) \right) - \text{Tanh}\left(\frac{R^{-1}}{2\mu_{sat}} \hat{\bar{D}}_k(t) \right) \right),$$

where $k_u \in \mathbb{R}_{>0}$ is a constant,

$$\hat{\bar{D}}(t) \triangleq G^T(\zeta(t)) \left(\nabla\sigma^T(\zeta(t), c(\zeta(t))) \hat{W}_a(t) + \nabla P_a^T(\zeta(t)) \right),$$

and

$$\hat{\bar{D}}_k(t) \triangleq G^T(\zeta_k(t)) \left(\nabla\sigma^T(\zeta_k(t), c(\zeta(t))) \hat{W}_a(t) + \nabla P_a^T(\zeta_k(t)) \right).$$

4.4 Stability Analysis

To facilitate the following stability analysis, let $B_\xi \subset \chi \times \mathbb{R}^L \times \mathbb{R}^L$ is a compact set containing the origin. The BEs in (4–18) and (4–20) can be expressed as

$$\delta_t = -\omega^T \tilde{W}_c + G_{a1}^T \tilde{W}_a + G_{a2}^T \tilde{W}_a + \Delta(\zeta),$$
$$\delta_k = -\omega_k^T \tilde{W}_c + G_{a1,k}^T \tilde{W}_a + G_{a2,k}^T \tilde{W}_a + \Delta_k(\zeta),$$

where $\tilde{W}_c$ and $\tilde{W}_a$ denote the critic and actor weight estimates as defined in Section 2.1. The terms G_{a2} and G_{a2k} are defined as $G_{a2} \triangleq \mu_{sat} \nabla\sigma G \left(\text{sgn}\left(\hat{\bar{D}} \right) - \text{Tanh}\left(\frac{1}{k_u} \hat{\bar{D}} \right) \right)$ and $G_{a2,k} \triangleq \mu_{sat} \nabla\sigma_k G_k \left(\text{sgn}\left(\hat{\bar{D}}_k \right) - \text{Tanh}\left(\frac{1}{k_u} \hat{\bar{D}}_k \right) \right)$. The functions $\Delta,\ \Delta_k : \mathbb{R}^{\mathcal{N}} \to \mathbb{R}$ are uniformly bounded over χ such that the residual bounds $\overline{\|\Delta\|},\ \overline{\|\Delta_k\|}$ decrease with decreasing $\overline{\|\nabla W\|}$ and $\overline{\|\nabla\epsilon\|}$.[4]

To facilitate the analysis, the system states x and selected states x_k are assumed to satisfy the following inequalities.

[4] For an arbitrary function ϕ, ϕ_k is defined as $\phi_k \triangleq \phi(\zeta_k(t))$.

Assumption 4.5. There exists constants $T \in \mathbb{R}_{>0}$ and $\underline{c}_1, \underline{c}_2, \underline{c}_3 \in \mathbb{R}_{\geq 0}$, such that

$$\underline{c}_1 I_L \leq \frac{1}{N} \sum_{k=1}^{N} \frac{\omega_k(t)\,\omega_k^T(t)}{\rho_k^2(t)},$$
$$\underline{c}_2 I_L \leq \int_t^{t+T} \left(\frac{1}{N} \sum_{k=1}^{N} \frac{\omega_k(\tau)\,\omega_k^T(\tau)}{\rho_k^2(\tau)} \right) d\tau, \ \forall t \in \mathbb{R}_{\geq t_0},$$
$$\underline{c}_3 I_L \leq \int_t^{t+T} \left(\frac{\omega(\tau)\,\omega^T(\tau)}{\rho^2(\tau)} \right) d\tau, \ \forall t \in \mathbb{R}_{\geq t_0},$$

where at least one of the constants $\underline{c}_1$, $\underline{c}_2$, or $\underline{c}_3$ is strictly positive [1].

Remark 4.8. In general, $\underline{c}_1$ can be made strictly positive by sampling redundant data, i.e, choosing $N \gg L$, and $\underline{c}_2$ can be made strictly positive by sampling extrapolated trajectories at a high frequency. Generally, $\underline{c}_3$ is strictly positive provided the system is PE, which is a strong assumption that cannot be verified online. Since only one constant has to be strictly positive, ω_k can be selected such that $\underline{c}_1 > 0$ or $\underline{c}_2 > 0$, since ω_k is a design variable. Unlike the strong PE given by the third inequality in Assumption 4.5, the first two inequalities can be verified online.[5]

Provided Assumption 4.5 is satisfied and $\lambda_{\min}\left\{\Gamma_0^{-1}\right\} > 0$, the update law in (4–23) ensures that the least squares gain matrix Γ satisfies

$$\underline{\Gamma} I_L \leq \Gamma(t) \leq \overline{\Gamma} I_L, \tag{4–25}$$

where the bounds $\underline{\Gamma}$ and $\overline{\Gamma}$ are defined as

$$\underline{\Gamma} = \frac{1}{\left(\lambda_{\max}\left\{\Gamma_0^{-1}\right\} + \frac{k_{c1}+k_{c2}}{4\gamma_1\beta}\right)},$$
$$\overline{\Gamma} = \frac{1}{\min\left\{\left(k_{c1}\underline{c}_3 + k_{c2}\max\left\{\underline{c}_1 T, \underline{c}_2\right\}\right), \lambda_{\min}\left\{\Gamma_0^{-1}\right\}\right\} e^{-\beta T}},$$

[5] See Footnote 1 in Chapter 3.

where $\lambda_{\min}\{\cdot\}$, $\lambda_{\max}\{\cdot\}$ denote the minimum and maximum eigenvalues, respectively (see [1]).

To facilitate the analysis, consider a candidate Lyapunov function $V_L : \mathbb{R}^{\mathcal{N}+2L} \times \mathbb{R}_{\geq t_0} \to \mathbb{R}$ given by

$$V_L(Y,t) = V^*(\zeta) + \frac{1}{2}\tilde{W}_c^T \Gamma^{-1}(t)\tilde{W}_c + \frac{1}{2}\tilde{W}_a^T \Gamma_a^{-1}\tilde{W}_a + \frac{1}{2}\sum_{i=1}^{M} z_i^T z_i, \tag{4–26}$$

where V^* is the optimal value function, and $Y = \left[\zeta^T, \tilde{W}_c^T, \tilde{W}_a^T\right]^T$. Since the optimal value function is positive definite, using (4–25) and [104, Lemma 4.3], (4–26) can be bounded as

$$\underline{\nu}_l(\|Y\|) \leq V(Y,t) \leq \overline{\nu}_l(\|Y\|), \tag{4–27}$$

for all $t \in \mathbb{R}_{\geq t_0}$ and for all $Y \in \mathbb{R}^{n+1+2L}$,where $\underline{\nu}_l, \overline{\nu}_l : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ are class $\mathcal{K}$ functions. To facilitate the following analysis, let $\nu_l : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ be a class $\mathcal{K}$ function such that

$$\nu_l(\|Y\|) \leq \frac{q}{2}\|x\|^2 + \frac{\underline{q}_z}{4}\sum_{i=1}^{M} s_i(x,z_i)\|z_i\|^2 + \left(\frac{k_{a1}+k_{a2}}{8}\right)\left\|\tilde{W}_a\right\|^2 + \frac{k_{c2}\underline{c}}{8}\left\|\tilde{W}_c\right\|^2, \tag{4–28}$$

and let $\underline{c} \in \mathbb{R}_{>0}$ be a constant defined as

$$\underline{c} \triangleq \frac{\beta}{2k_{c2}\overline{\Gamma}} + \frac{\underline{c}_1}{2}. \tag{4–29}$$

The sufficient conditions for the subsequent analysis are given by

$$\frac{k_{a1}+k_{a2}}{2} \geq \max\left\{\varphi_{ac}, \frac{\overline{\|\nabla W\|}G_R\overline{\|\nabla\sigma^T\|}}{\lambda_{\min}\{\Gamma_a\}}\right\}, \tag{4–30}$$

$$k_{c2}\underline{c} \geq \varphi_{ac}, \tag{4–31}$$

$$\frac{1}{2}\underline{q}_z \geq L_z, \tag{4–32}$$

$$\nu_\iota^{-1}(\iota) < \overline{\nu}_\iota^{-1}(\underline{\nu}_\iota(\xi)), \tag{4–33}$$

where L_z is the Lipschitz constant such that $\|h_i(z_i)\| \leq L_z\|z_i\|$ satisfying assumption (4.2) and $\varphi_{ac} \in \mathbb{R}_{>0}$ is defined in Appendix A.2.

Theorem 4.1. *Consider the augmented dynamic system (4–4) and the dynamic systems in (4–1) and (4–3). Provided Assumptions 4.1-4.5 are satisfied along with the sufficient conditions in (4–30)-(4–33), then system state $\zeta(t)$, input $u(t)$, and weight approximation errors $\tilde{W}_a$ and $\tilde{W}_c$ are UUB; furthermore, states $\zeta(t)$ starting outside of Ω remain outside of Ω.*

Proof. Consider the Lyapunov function candidate in (4–26). The time derivative is given by

$$\dot{V}_L = \dot{V}^* + \tilde{W}_c^T \Gamma^{-1} \left(\dot{W} - \dot{\hat{W}}_c \right) + \tilde{W}_a^T \Gamma_a^{-1} \left(\dot{W} - \dot{\hat{W}}_a \right) - \frac{1}{2} \tilde{W}_c^T \Gamma^{-1} \dot{\Gamma} \Gamma^{-1} \tilde{W}_c + \sum_{i=1}^{M} z_i^T \left(\mathscr{F}_i h_i \right).$$

Using the chain rule, the time derivative of the ideal weights $\dot{W}$ can be expressed as

$$\dot{W} = \nabla W \left(F + Gu \right). \tag{4–34}$$

Substituting in (4–22)-(4–24) with (4–34) yields

$$\begin{aligned}
\dot{V}_L = {} & \nabla V^* F + \nabla V^* G u + \sum_{i=1}^{M} z_i^T \left(\mathscr{F}_i h_i \right) + \tilde{W}_c^T \Gamma^{-1} \left(k_{c1} \Gamma \frac{\omega}{\rho} \delta_t + \frac{k_{c2}}{N} \Gamma \sum_{k=1}^{N} \frac{\omega_k}{\rho_k} \delta_k \right) \\
& + \tilde{W}_a^T k_{a1} \left(\hat{W}_a - \hat{W}_c \right) + \tilde{W}_a^T k_{a2} \hat{W}_a(t) + \tilde{W}_a^T \left(k_{c1} G_{a1} \frac{\omega^T}{\rho} - \frac{k_{c2}}{N} \sum_{k=1}^{N} G_{a1,k} \frac{\omega_k^T}{\rho_{ik}} \right) \hat{W}_c(t) \\
& - \frac{1}{2} \tilde{W}_c^T \Gamma^{-1} \left(\beta \Gamma - k_{c1} \Gamma \frac{\omega \omega^T}{\rho^2} \Gamma - \frac{k_{c2}}{N} \Gamma \sum_{k=1}^{N} \frac{\omega_k \omega_k^T}{\rho_k^2} \Gamma \right) \Gamma^{-1} \tilde{W}_c. \\
& + \left(\tilde{W}_c^T \Gamma^{-1} + \tilde{W}_a^T \right) \nabla W \left(F + Gu \right)
\end{aligned}$$

Using (4–6) with (4–10), (4–18)-(4–21), and Young's inequality, the Lyapunov derivative can be bounded as

$$\begin{aligned}
\dot{V}_L \leq {} & -\underline{q}_x \left\| x \right\|^2 - \frac{\underline{q}_z}{2} \sum_{i=1}^{M} s_i \left(x, z_i \right) \left\| z_i \right\|^2 - 2 \left(\frac{k_{a1} + k_{a2}}{8} \right) \left\| \tilde{W}_a \right\|^2 - 2 \left(\frac{k_{c2} \underline{c}}{8} \right) \left\| \tilde{W}_c \right\|^2 + \iota \\
& - \begin{bmatrix} \left\| \tilde{W}_c \right\| & \left\| \tilde{W}_a \right\| \end{bmatrix} \begin{bmatrix} \frac{k_{c2} \underline{c}}{2} & -\frac{\varphi_{ac}}{2} \\ -\frac{\varphi_{ac}}{2} & \frac{k_{a1} + k_{a2}}{4} \end{bmatrix} \begin{bmatrix} \left\| \tilde{W}_c \right\| \\ \left\| \tilde{W}_a \right\| \end{bmatrix} - \frac{\underline{q}_z}{2} \sum_{i=1}^{M} s_i \left(x, z_i \right) \left\| z_i \right\|^2 + \sum_{i=1}^{M} z_i^T \left(\mathscr{F}_i h_i \right)
\end{aligned}$$

where $\iota \in \mathbb{R}_{>0}$ is a positive constant defined in the appendix. Using (4–28), (4–30), and (4–31), the Lyapunov derivative reduces to

$$\dot{V}_L \leq -\nu_l\left(\|Y\|\right) - \left(\nu_l\left(\|Y\|\right) - \iota\right) - \frac{\underline{q_z}}{2}\sum_{i=1}^{M} s_i\left(x, z_i\right)\|z_i\|^2 + \sum_{i=1}^{M} z_i^T\left(\mathscr{F}_i h_i\right).$$

For the case when $x, z_i \notin \mathcal{D}\ \forall i \in \mathcal{M}$, the avoidance region dynamics in (4–3) can be used conclude that, $\frac{q_z}{2}\sum_{i=1}^{M} s_i\left(x, z_i\right)\|z_i\|^2 + \sum_{i=1}^{M} z_i^T\left(\mathscr{F}_i h_i\right) = 0$; therefore,

$$\dot{V}_L \leq -\nu_l\left(\|Y\|\right) - \left(\nu_l\left(\|Y\|\right) - \iota\right).$$

Provided the sufficient conditions in (4–30), (4–31), and (4–33) are met, then

$$\dot{V}_L \leq -\nu_l\left(\|Y\|\right), \quad \forall Y \in \chi,\ \forall \|Y\| \geq \nu_l^{-1}\left(\iota\right).$$

For the case when $\zeta \in \mathcal{W}$, Assumption 4.2 is used to conclude that

$$\dot{V}_L \leq -\nu_l\left(\|Y\|\right) - \left(\nu_l\left(\|Y\|\right) - \iota\right) - \frac{\underline{q_z}}{2}\sum_{i=1}^{M} s_i\left(x, z_i\right)\|z_i\|^2 + \sum_{i\in\mathcal{M}} L_z\|z_i\|^2.$$

Using the fact that $\inf_{x,z_i\in\mathcal{W}_i} s_i\left(x, z_i\right) = 1$ for any $i \in \mathcal{M}$, and provided the sufficient conditions in (4–30)-(4–33) hold,

$$\dot{V}_L \leq -\nu_l\left(\|Y\|\right), \quad \forall \|Y\| \geq \nu_l^{-1}\left(\iota\right). \tag{4–35}$$

Hence, (4–26) is non-increasing.

If $\|x - z_i\| \to r_a$ for some $i \in \mathcal{M}$, then $P\left(\zeta\right) \to \infty$, and $V^*\left(\zeta\right) \to \infty$. If $V^*\left(\zeta\right) \to \infty$ then $V_L\left(Y\right) \to \infty$. Since this is a contradiction to (4–26) being non-increasing, then $\forall \zeta\left(t_0\right) \notin \Omega\,, \zeta\left(t\right) \notin \Omega\ \forall t \geq t_0$. Hence, $V^*\left(\zeta\right)$ is finite and $\nabla V^*\left(\zeta\right)$ exists for all $\|x - z_i\| \neq r_a$.

After using (4–27), (4–33), and (4–35), [104, Theorem 4.18] can be invoked to conclude that Y is UUB such that $\limsup_{t\to\infty}\|Y\left(t\right)\| \leq \underline{\nu_\iota}^{-1}\left(\overline{\nu_\iota}\left(\nu_\iota^{-1}\left(\iota\right)\right)\right)$. Since $Y \in L_\infty$, it follows that $\zeta, \tilde{W}_c, \tilde{W}_a \in L_\infty$. Since W is a continuous function of ζ, $W \circ \zeta \in L_\infty$. Hence, $\hat{W}_a, \hat{W}_c \in L_\infty$ which implies $u \in L_\infty$. □

Remark 4.9. The sufficient condition in (4–30) can be satisfied by increasing the gain k_{a2} and selecting a gain Γ_a such that $\lambda_{\min}\{\Gamma_a\}$ is large. This will not affect the sufficient conditions in (4–31) and (4–32). Selecting extrapolated trajectories x_k such that $\underline{c}$ is sufficiently large will aid in satisfying (4–31) without affecting (4–30) or (4–32). In addition, selecting StaF basis such that $\overline{\|\nabla\sigma\|}$ is small will help satisfy the conditions in (4–30) and (4–31). To satisfy the sufficient condition in (4–32) without affecting (4–30) or (4–31), it suffices to select a function Q_z according to Assumption 4.4 such that $\underline{q}_x$ is larger than the Lipschitz constant L_z. Provided the StaF basis functions are selected such that $\overline{\|\epsilon\|}$, $\overline{\|\nabla\epsilon\|}$, and $\overline{\|\nabla W\|}$ are small, and k_{a2} and $\underline{c}$ are selected to be sufficiently large, then the sufficient condition in (4–33) can be satisfied.

4.5 Extension to Uncertain Number of Avoidance Regions and Uncertain Systems

The HJB in (4–10) requires the number of no-entry zones in the operating domain to be known, which may not always be available. However, adding and subtracting $P_a(x, Z)$, the following value function is introduced

$$V^*(x(t), Z(t)) = P_a(x(t), Z(t)) + V^{\#}(x(t), Z(t)), \tag{4–36}$$

where $V^{\#}(x(t), Z(t))$ is an approximation error of the optimal value function. Furthermore, the function $V^{\#}(x, Z)$ can be interpreted as time-varying map $V_t^{\#} : \mathbb{R}^n \times \mathbb{R}_{\geq t_0}$ such that $V_t^{\#}(x, t) = V^{\#}(x, Z)$ [106]. Therefore, (4–36) is rewritten as

$$V^*(x(t), Z(t)) = P_a(x(t), Z(t)) + V_t^{\#}(x(t), t). \tag{4–37}$$

The optimal controller u^* is admissible; hence, the value function $V^*(x, Z)$ is finite and $x, Z \notin \Omega$. Therefore, $P_a(x, Z)$ is continuous for $x, Z \notin \Omega$, hence (4–37) can be approximated via the StaF approximation method. However, because time does not lie on a compact domain, $V_t^{\#}$ can not be approximated directly using time as an input to the NN. To address this technical challenge, the mapping $\phi : \mathbb{R}_{\geq t_0} \to [0, \alpha]$, $\alpha \in \mathbb{R}_{>0}$ is

introduced such that $V_t^{\#}\left(x\left(t\right),t\right)=V_t^{\#}\left(x\left(t\right),\phi^{-1}\left(\kappa\right)\right)=V_\kappa^{\#}\left(x\left(t\right),\kappa\right)$ where $\kappa=\phi\left(t\right)$. Now, κ lies on a compact set and the function $V_\kappa^{\#}\left(x,\kappa\right)$ can be approximated using the StaF method as

$$V^{*}\left(x\left(t\right),Z\left(t\right)\right)=P_a\left(x\left(t\right),Z\left(t\right)\right)+W^T\left(\zeta^{\#}\left(t\right)\right)\sigma\left(y\left(t\right),c\left(\zeta^{\#}\left(t\right)\right)\right)+\varepsilon\left(y\left(t\right),\zeta^{\#}\left(t\right)\right),$$

with $\sigma\left(\zeta^{\#},c\left(\zeta^{\#}\right)\right)=\begin{bmatrix}\sigma_0\left(x,c_0\left(x\right)\right)\\ s_0\left(x\right)\sigma_1\left(\kappa,c_1\left(\kappa\right)\right)\end{bmatrix}$, where $\zeta^{\#}\triangleq\left[x^T,\kappa\right]^T$, $y\triangleq\left[y_x^T,y_\kappa\right]^T\in\overline{B_r\left(\zeta^{\#}\right)}$, and $s_0:\mathbb{R}^n\to[0,1]$ is a smooth function such that $s_0\left(0_{2\times1}\right)=0$.

Moreover, since $P_a\left(x,Z\right)=\sum_{i\in\mathcal{M}}P_{a,i}\left(x,z_i\right)$ is designed to be a bounded positive semi-definite (PSD) symmetric function, it follows that $\frac{\partial P_{a,i}\left(x,z_1,\ldots,z_m\right)}{\partial x}=-\frac{\partial P_{a,i}\left(x,z_1,\ldots,z_m\right)}{\partial z_i}$ for all $i\in\mathcal{M}$; hence, the HJB is represented as

$$\begin{aligned}0=r\left(x,Z,u\right)&+\frac{\partial V_\kappa^{\#}\left(\zeta^{\#}\right)}{\partial\zeta^{\#}}\left(F^{\#}\left(\zeta^{\#}\right)+G^{\#}\left(\zeta^{\#}\right)u\right)\\&+\sum_{i=1}^{M}\frac{\partial P_{a,i}}{\partial x}\left(f\left(x\right)+g\left(x\right)u-\mathscr{F}_i\left(x,z_i\right)h_i\left(z_i\right)\right),\end{aligned}\tag{4–38}$$

where $F^{\#}\left(\zeta^{\#}\right)\triangleq\begin{bmatrix}f\left(x\right)^T,&\frac{\partial\kappa}{\partial t}\end{bmatrix}^T$, and $G^{\#}\left(\zeta^{\#}\right)\triangleq\left[g\left(x\right)^T,0_{m\times1}\right]^T$. The HJB in (4–38) requires the knowledge of the uncertain dynamics $f\left(x\right)$ and $h_i\left(z_i\right)$. Using a NN approximator, the time-derivative of P_a is written as

$$\begin{aligned}\dot{P}_a&=\sum_{i=1}^{M}\frac{\partial P_{a,i}}{\partial x}\left(f\left(x\right)+g\left(x\right)u-\mathscr{F}_i\left(x,z_i\right)h_i\left(z_i\right)\right)\\&=Y_p\left(x,Z\right)\theta+\varepsilon_p\left(x,Z\right),\end{aligned}$$

where $Y_p:\mathbb{R}^n\times\mathbb{R}^{Mn}\to\mathbb{R}^{1\times l_p}$ is a selected basis such that $Y_p\left(x,Z\right)=0_{1\times l_p}$ when $\left\|x-z_i\right\|>r_d$, for all $i\in\mathcal{M}$, $\theta\in\mathbb{R}^{l_p}$ is an unknown weight, and $\varepsilon_p:\mathbb{R}^n\times\mathbb{R}^{Mn}\to\mathbb{R}$ is the unknown function approximation error. Likewise the agent drift dynamics can be

represented as $f(x(t)) = Y_f(x(t))\,\Xi + \varepsilon_f(x(t))$ with $Y_f : \mathbb{R}^n \to \mathbb{R}^{n\times l_f}$ being a known basis, $\Xi \in \mathbb{R}^{l_f}$ an unknown weight, and $\varepsilon_f : \mathbb{R}^n \to \mathbb{R}^n$ the function approximation error.[6]

Assumption 4.6. There exists constants $\overline{\varepsilon}_p, \overline{\varepsilon}_f, \overline{Y}_f, \overline{Y}_p, \overline{\theta}, \overline{\Xi} \in \mathbb{R}_{>0}$ such that $\sup_{\zeta\in\chi} \|Y_p(x,Z)\| \leq \overline{Y}_p$, $\sup_{\zeta\in\chi} \|\varepsilon_p(x,Z)\| \leq \overline{\varepsilon}_p$, $\sup_{x\in\chi} \|Y_f(x)\| \leq \overline{Y}_f$, $\sup_{x\in\chi} \|\varepsilon_f(x)\| \leq \overline{\varepsilon}_f$, $\|\theta\| \leq \overline{\theta}$, and $\|\Xi\| \leq \overline{\Xi}$ [18, 98].

Using the estimates $\hat{W}_c$,$\hat{W}_a$, $\hat{\theta}$, and $\hat{\Xi}$ in (4–38), the approximate BE $\hat{\delta} : \mathbb{R}^{n+1} \times \mathbb{R}^{n+1} \times \mathbb{R}^{\mathcal{N}} \times \mathbb{R}^L \times \mathbb{R}^L \times \mathbb{R}^{l_f+l_p} \to \mathbb{R}$ is defined as

$$\hat{\delta}\left(y, \zeta^{\#}, Z, \hat{W}_c, \hat{W}_a, \hat{\theta}, \hat{\Xi}\right) \triangleq Y_p(y_x, Z)\,\hat{\theta} + \omega^{\#}\left(y, \zeta^{\#}, Z, \hat{W}_a, \hat{\Xi}\right)^T \hat{W}_c + r\left(y_x, Z, \hat{u}\left(y, \zeta^{\#}, Z, \hat{W}_a\right)\right), \tag{4–39}$$

where $\omega^{\#}\left(y, \zeta^{\#}, Z, \hat{W}_a, \hat{\Xi}\right) \triangleq \nabla\sigma\left(y, c\left(\zeta^{\#}\right)\right)\left(Y^{\#}(y)\,\hat{\Xi}^{\#} + G^{\#}(y)\,\hat{u}\left(y, \zeta^{\#}, Z, \hat{W}_a\right)\right)$, $Y^{\#}(y) \triangleq \left[\, Y_f(y_x)^T, \ \frac{\partial y_\kappa}{\partial t} \,\right]^T$, $\hat{\Xi}^{\#} \triangleq \begin{bmatrix} \hat{\Xi} & 0_{l_f\times 1} \\ 0_{1\times 1} & 1 \end{bmatrix}$, and

$$\hat{u}\left(y, \zeta^{\#}, Z, \hat{W}_a\right) \triangleq -\mu_{sat}\,\mathrm{Tanh}\left(\frac{1}{2\mu_{sat}} R^{-1} G^{\#T}(y)\left(\nabla P_a^T(y_x, Z) + \nabla\sigma^T\left(y, c\left(\zeta^{\#}\right)\right)\hat{W}_a\right)\right), \tag{4–40}$$

where $\nabla P_a(y_x, Z) \triangleq \left[\frac{\partial P_a(y_x,Z)}{\partial x}, \ \frac{\partial P_a(y_x,Z)}{\partial \kappa}\right] = \left[\frac{\partial P_a(y_x,Z)}{\partial x}, \ 0\right]$. Using $\hat{\delta}$, the instantaneous BEs and approximate policies in (4–18)-(4–21) are re-defined as $\delta_t(t) \triangleq \hat{\delta}\left(\zeta^{\#}(t), \zeta^{\#}(t), Z(t), \hat{W}_c(t), \hat{W}_a(t), \hat{\theta}(t), \hat{\Xi}(t)\right)$, $\delta_k(t) \triangleq \hat{\delta}\left(\zeta_k^{\#}(t), \zeta^{\#}(t), Z(t), \hat{W}_c(t), \hat{W}_a(t), \hat{\theta}(t), \hat{\Xi}(t)\right)$, $u(t) \triangleq \hat{u}\left(\zeta^{\#}(t), \zeta^{\#}(t), Z(t), \hat{W}_a(t)\right)$, and $\hat{u}_k(t) \triangleq \hat{u}\left(\zeta_k^{\#}(t), \zeta^{\#}(t), Z(t), \hat{W}_a(t)\right)$, respectively.

Assumption 4.7. [1, 18]. There exists a compact set $\Theta \subset \mathbb{R}^{l_p+l_f}$, known *a priori*, which contains the unknown parameter vectors θ and Ξ. Let $\tilde{X} \triangleq \left[\tilde{\Xi}^T, \tilde{\theta}^T\right]^T =$

[6] If the agent dynamics $f(x(t))$ are assumed to be single integrator dynamics such that $f(x(t)) = 0_{n\times 1}$, system identification for the agent is not necessary.

$\left[\left(\Xi - \hat{\Xi}\right)^T, \left(\theta - \hat{\theta}\right)^T\right]^T$ and $\hat{X} = \left[\hat{\Xi}^T, \hat{\theta}^T\right]^T$ denote the total concatenated vector of parameter estimate errors and parameter estimates, respectively. The estimates $\hat{X} : \mathbb{R}_{\geq t_0} \to \mathbb{R}^{l_p + l_f}$ are updated based on switched update laws of the form

$$\dot{\hat{X}}(t) = f_{Xs}\left(\hat{X}(t), t\right), \quad \hat{X}(t_0) \in \Theta, \tag{4–41}$$

where $s \in \mathbb{N}$ is the switching index with $\left\{f_{Xs} : \mathbb{R}^{l_p+l_f} \times \mathbb{R}_{\geq t_0} \to \mathbb{R}^{l_p+l_f}\right\}_{s\in\mathbb{N}}$ being a family of continuously differentiable functions. There exist a continuously differentiable function $V_\theta : \mathbb{R}^{l_p+l_f} \times \mathbb{R}_{\geq t_0} \to \mathbb{R}_{\geq 0}$ that satisfies

$$\underline{\nu}_\theta\left(\left\|\tilde{X}\right\|\right) \leq V_\theta\left(\tilde{X}, t\right) \leq \overline{\nu}_\theta\left(\left\|\tilde{X}\right\|\right), \tag{4–42}$$

$$\frac{\partial V_\theta\left(\tilde{X}, t\right)}{\partial \tilde{X}}\left(-f_{Xs}\left(\tilde{X}(t), t\right)\right) + \frac{\partial V_\theta\left(\tilde{X}, t\right)}{\partial t} \leq -K_\theta\left\|\tilde{X}\right\|^2 + D\left\|\tilde{X}\right\|, \tag{4–43}$$

for all $t \in \mathbb{R}_{\geq t_0}$, $s \in \mathbb{N}$, and $\tilde{X} \in \mathbb{R}^{l_p+l_f}$. In (4–42), $\underline{\nu}_\theta, \overline{\nu}_\theta : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ are class $\mathcal{K}$ functions. In (4–43), $K_\theta \in \mathbb{R}_{>0}$ is an adjustable parameter, $D \in \mathbb{R}_{>0}$ is a positive constant, and the ratio $\frac{D}{K_\theta}$ is sufficiently small.[7]

Remark 4.10. If $f(x(t)) = 0_{n\times 1}$, then $Y^{\#}(y)$ and $\hat{\Xi}^{\#}$ simplify to $Y^{\#}(y) \triangleq \begin{bmatrix} 0_{l_f \times n}, & \frac{\partial y_\kappa}{\partial t} \end{bmatrix}^T$ and $\hat{\Xi}^{\#} \triangleq \begin{bmatrix} 0_{l_f\times 1} & 0_{l_f\times 1} \\ 0_{1\times 1} & 1 \end{bmatrix}$, respectively. Furthermore, Ξ does not need to be estimated for single integrator dynamics and the concatenated systems then reduce to $\tilde{X} \triangleq \tilde{\theta}$ and $f_{Xs}\left(\hat{X}(t), t\right) \triangleq f_{\theta s}\left(\hat{\theta}(t), t\right)$.

The conditions (4–42) and (4–43) in Assumption 4.7 imply that V_θ can be used as a candidate Lyapunov function to show the parameter estimates $\hat{\theta}$ and $\hat{\Xi}$ converge to a neighborhood of the true values. Update laws using CL-based methods can be designed to satisfy Assumption 4.7; examples of such update laws can be found in [55, 107–109]. The main result for the extension to systems with uncertainties and an

[7] The positive constant D can possibly depend on the parameter K_θ.

unknown number of avoidance regions uses $V_\theta + V_L$ as a candidate Lyapunov function and is summarized in the following theorem.

Theorem 4.2. *Provided Assumptions 4.2-4.7 along with the sufficient conditions in (4–30)-(4–33) are satisfied, and StaF kernels are selected such that* ∇W, ε, $\nabla\varepsilon$, *are sufficiently small, then the update laws in (4–22)-(4–24) with (4–40),* $\delta_t(t)$, *and* $\delta_k(t)$ *ensure that the state* x *and input* $u(t)$*, and weight approximation errors* $\tilde{W}_a$, $\tilde{W}_c$, $\tilde{\theta}$, $\tilde{\Xi}$ *are UUB; furthermore, states* $x(t)$, $z_i(t)$ *starting outside of* Ω *remain outside of* Ω*.*

Proof. The proof is a combination of Assumption 4.7 with Theorem 4.1 by using $V_L + V_\theta$ as a candidate Lyapunov function; hence, the proof is omitted to alleviate redundancy. □

4.6 Simulation-in-the-Loop Experiments

Three experiments are conducted to demonstrate the ability of an aerial vehicle to be autonomously regulated to the origin while avoiding dynamic avoidance regions. For each experiment, a Parrot Bebop 2.0 quadcopter is used as the aerial vehicle. The developed quadcopter controller requires feedback of its and the obstacles' position and orientation (pose). The pose of the quadcopter is obtained by a NaturalPoint, Inc. OptiTrack motion capture system at 120Hz. Using the Robotic Operating System (ROS) Kinetic framework and the *bebop_*autonomy *package* developed by [110] running on Ubuntu 16.04, the control policies are calculated for the quadcopter. The control policy is communicated from a ground station which broadcasts velocity commands at 120Hz over the 5GHz Wi-Fi channel.[8]

[8] The developed control policy is implemented as a velocity command to the quadcopter. While this allows an effective demonstration of the underlying strategy, improved performance could be obtained by implementing the policies through acceleration commands that do not rely on the on-board velocity tracking controller. Such an implementation could also have additional implications due to input constraints for acceleration commands.

All three experiments use simplified quadcopter dynamics represented by (4–1) with $f(x(t)) = 0_{2\times 1}$ and $g(x(t)) = I_2$ so that $\dot{x} = u$, where, without a loss of generality, $x(t) \in \mathbb{R}^2$ is the composite vector of the 2-D Euclidean coordinates, with respect to the inertial frame and $u \in \mathbb{R}^2$ are velocity commands broadcast to the quadcopter.[9] For the first two experiments, virtual spheres are used as the dynamic avoidance regions. The virtual spheres, which evolve according to linear oscillatory dynamics, are generated using ROS via Ubuntu on the ground station. The positions of the virtual spheres in the inertial frame are used in the designed method to interact with the vehicle, only when each position is within the detection radius of the quadcopter. For the third experiment, one of the virtual spheres is replaced by a remotely controlled (i.e. human piloted) quadcopter.

Experiment One

The first experiment is performed using the method developed in Sections 4.1-4.4. Three virtual avoidance regions are generated using heterogeneous oscillatory linear dynamics. The function $\mathscr{F}_i(x, z_i)$ is selected as

$$\mathscr{F}_i(x, z_i) = \begin{cases} 0, & \|x - z_i\| > r_d, \\ T_i^{(a,b]}(x, z_i), & r_d \geq \|x - z_i\| > \bar{r}, \\ 1, & \|x - z_i\| \leq \bar{r}, \end{cases} \tag{4–44}$$

where $T_i^{(a,b]}(x, z_i) \triangleq \frac{1}{2} + \frac{1}{2}\cos\left(\pi\left(\frac{\|x-z_i\|-a}{b-a}\right)\right)$ with the smooth scheduling function $s_i(x, z_i) = \mathscr{F}_i(x, z_i)$, and P_a is selected as $P_a = \sum_{i=1}^{M}\left(\min\left\{0, \frac{\|x-z_i\|^2 - r_d^2}{\left(\|x-z_i\|^2 - r_a^2\right)^2 + r_\varepsilon}\right\}\right)^2$. For value function approximation, the agent is selected to have the StaF basis $\sigma_0(x, c(x)) =$

[9] The experiments are performed using 2-D Euclidean coordinates (without the inclusion of altitude) for the state $x(t)$ for ease of experimental execution and implementation, and result exposition. However, since the development does not restrict the state dimension, experiments can also be extended to use 3-D Euclidean coordinates as $x(t)$.

$\left[x^T c_1(x), x^T c_2(x), x^T c_3(x)\right]^T$, where $c_i(x) = x + \nu(x) d_i$, $i = 1, 2, 3$, where $\nu(x) \triangleq \frac{0.5 x^T x}{1 + x^T x}$ and the offsets are selected as $d_1 = \begin{bmatrix} 0, & -1 \end{bmatrix}^T$, $d_2 = \begin{bmatrix} 0.866, & -0.5 \end{bmatrix}^T$, and $d_3 = \begin{bmatrix} -0.866, & -0.5 \end{bmatrix}^T$. The StaF basis σ_i for each obstacle is selected to be the same as the agent, except that the state changes from x to z_i. Assumption 4.5 discussed how the extrapolated regressors ω_k are design variables. Thus, instead of using input-output data from a persistently exciting system, the dynamic model can be used and evaluated at a single time-varying extrapolated state to achieve sufficient excitation. It was shown in [1, Section 6.3] that the use of a single time-varying extrapolated point results in improved computational efficiency when compared using a large number of stationary extrapolated states. Motivated by this insight, at each time a single point is selected at random from a $0.2\nu(x(t)) \times 0.2\nu(x(t))$ uniform distribution centered at the current state. The initial critic and actor weights and gains are selected as $W_c(0) = U[0, 4] 1_{12\times 1}$, $W_a(0) = 1_{12\times 1}$, and $\Gamma_a = I_{12}$, and the selected parameters are shown in Table 5-1.

Experiment Two

The second experiment is performed using the extension in Section 4.5 and similar to Experiment One, three virtual avoidance regions are generated with heterogeneous oscillatory linear dynamics. The agent has the same basis $\sigma_0(x)$ as the first experiment, while the basis $\sigma_1(\kappa, c(\kappa))$ is selected as $\sigma_1(\kappa, c(\kappa)) = \left[\kappa^T c_1(\kappa), \kappa^T c_2(\kappa), \right]^T$, where $\kappa = \phi(t) \triangleq \frac{0.25}{0.01t+1}$ and $c_i(\kappa) = \kappa + \nu(\kappa) d_i$, $i = 1, 2$ where $\nu(\kappa)$ is the same function as in the first experiment except evaluated at κ and the offsets are selected as $d_1 = 0.25$, and $d_2 = 0.05$. For the total basis $\sigma\left(\zeta^{\#}, c\left(\zeta^{\#}\right)\right)$, the function $s_0(x)$ is selected as $s_0(x) = \frac{\nu(x)}{0.5}$. The initial critic and actor weights and adaptive gains are selected as $W_c(0) = U[0, 4] 1_{5\times 1}$, $W_a(0) = 1_{5\times 1}$, and $\Gamma_a = I_5$. The rest of the parameters are selected to remain the same as in the first experiment and are shown in Table 5-1. Since the agent dynamics are modeled as single integrator dynamics with $f(x(t)) = 0_{2\times 1}$,

Table 4-1. Initial conditions and parameters selected for the simulation.

Agent Initial conditions at $t_0 = 0$ $x(0) = [-6.3, 1.5]^T$,
Penalizing parameters and input saturations $Q_x(x) = x^T q_x x$, $Q_z(z_i) = z_i^T q_z z_i$, $R = 10 I_2$, $q_x = \text{diag}\{2.0, 1.0\}$, $q_z = \text{diag}\{2.0, 2.0\}$, $\mu_{sat} = 0.5$,
Gains for ADP update laws $k_{c1} = 0.05$, $k_{c2} = 0.75$, $k_{a1} = 0.75$, $k_{a2} = 0.01$, $\gamma_1 = 1$, $\beta = 0.001$, $k_u = 1$,
Radii $r_d = 0.7$, $\bar{r} = 0.45$, $r_a = 0.2$, $r_\varepsilon = 0.15$.

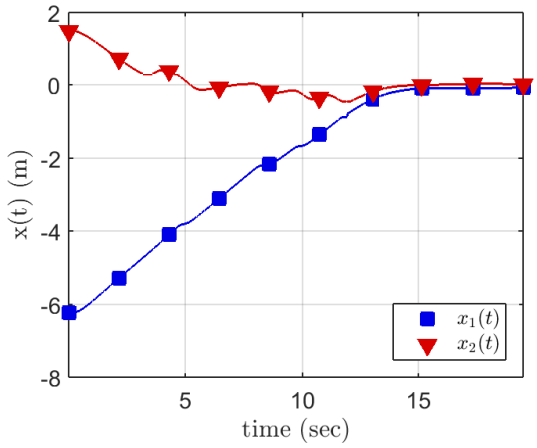
system identification was not performed on the agent.[10] However, to approximate θ in Section 4.5, the ICL method in [109, Section IV.B] was utilized with the basis $Y_p(x, Z) = \text{Tanh}\left(V_p^T \nabla P_a(y_x, Z)^T\right)$, where $V_p = U[-5, 5]\, 1_{3\times 10}$ is a constant weight matrix.[11]

Experiment Three

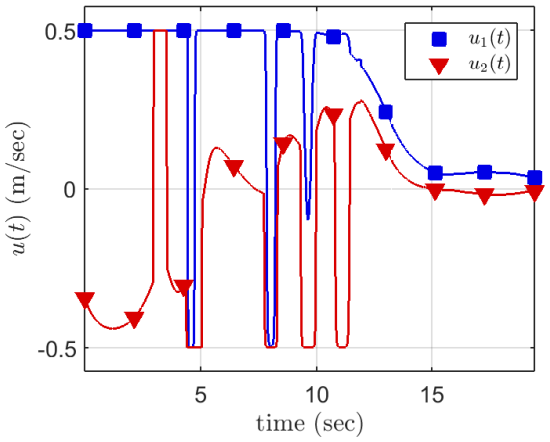
The third experiment is performed using the extension in Section 4.5 where the first avoidance region, denoted by the state z_1 and represented by another Parrot Bebop quadcopter, is flown/controlled manually by hand. The virtual avoidance regions with states z_2 and z_3 are simulated as in the previous experiments. The radii were changed to $r_d = 1.0$, $\bar{r} = 0.7$, and $r_a = 0.45$ to reduce the chance of the quadcopters colliding, the gains q_x, q_z where changed to $q_x = \text{diag}\{0.5, 0.5\}$ and $q_z = \text{diag}\{3.0, 3.0\}$, and the rest of the parameters remained the same as in the second experiment.

[10] Not performing system identification on the agent reduces redundancy in parameter identification because the unknown weight θ in the function in the time derivative of P_a is already being approximated. Furthermore, as stated in Footnote 6, if the agent is implemented using single integrator dynamics, then system identification can be ignored on the agent drift dynamics $f(x(t))$.

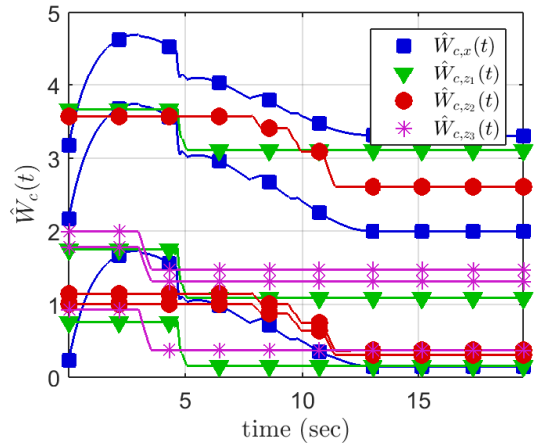
[11] To keep the weight estimates bounded, a projection algorithm was used similar to [109, Section IV.B] and the update laws where turned off when no avoidance regions were sensed.



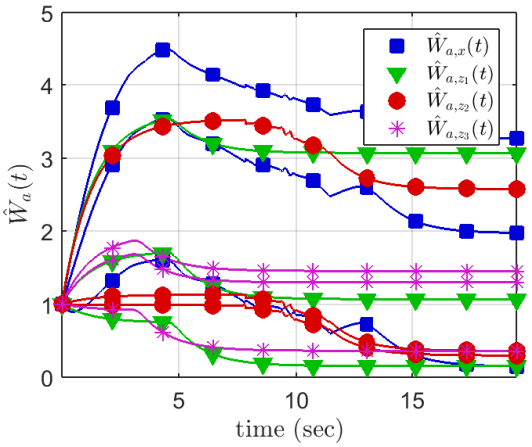
(a) The agent states.



(b) The agent approximate optimal input.



(c) The critic weight estimates.



(d) The actor weight estimates.

Figure 4-2. The states, control policy, and weight estimates are shown in addition to the distances between the agent and each avoidance region center for the first experiment. Figure 4-2a shows that the agents states converge to a close neighborhood of the origin. When the agent detects the avoidance regions, the commanded input, shown in Figure 4-2b, causes the agent to steer off-course as shown by the change in the trajectory of x_2 in Figure 4-2a.

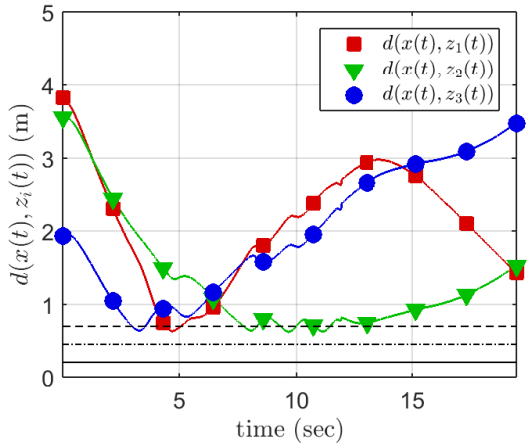
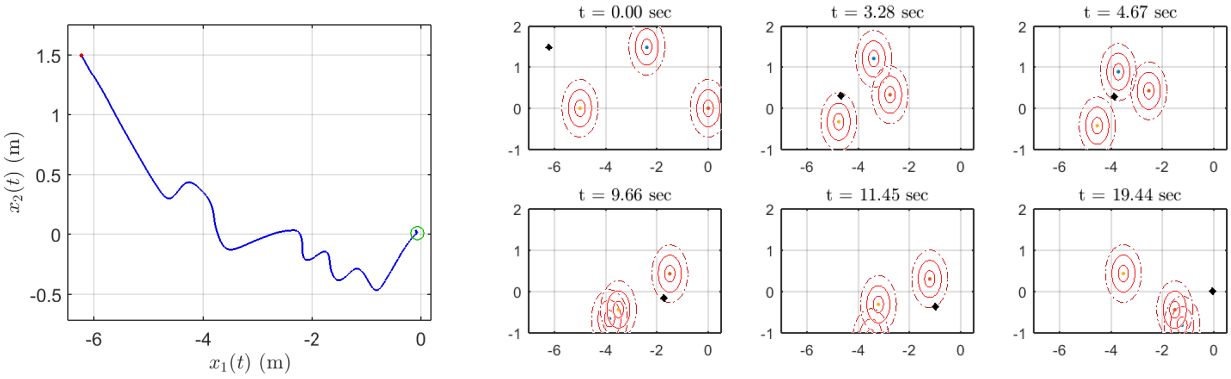


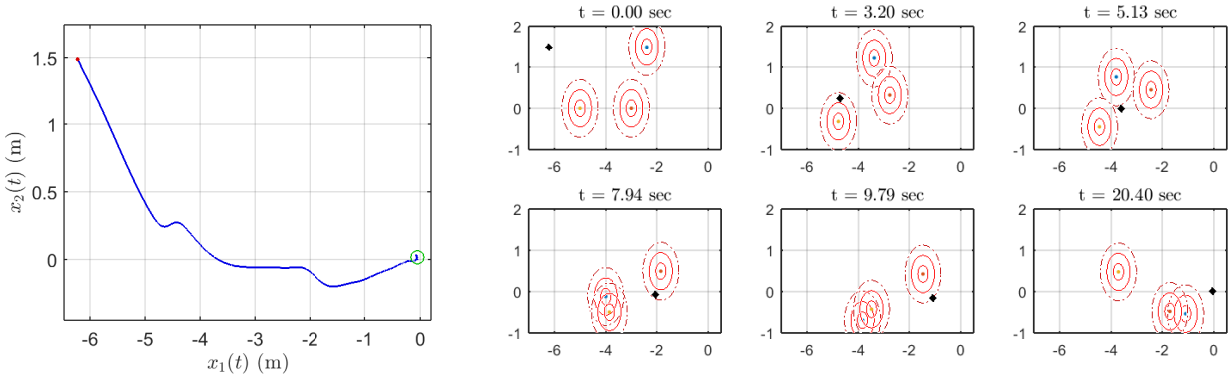
Figure 4-3. The distance between the agent and avoidance regions.The two dashed horizontal lines represent the detection radius and conflict radius denoted by $r_d = 0.7$ and $\overline{r} = 0.45$, respectively, while the solid horizontal line represents the radius of the avoidance region denoted by $r_a = 0.2$.

Results

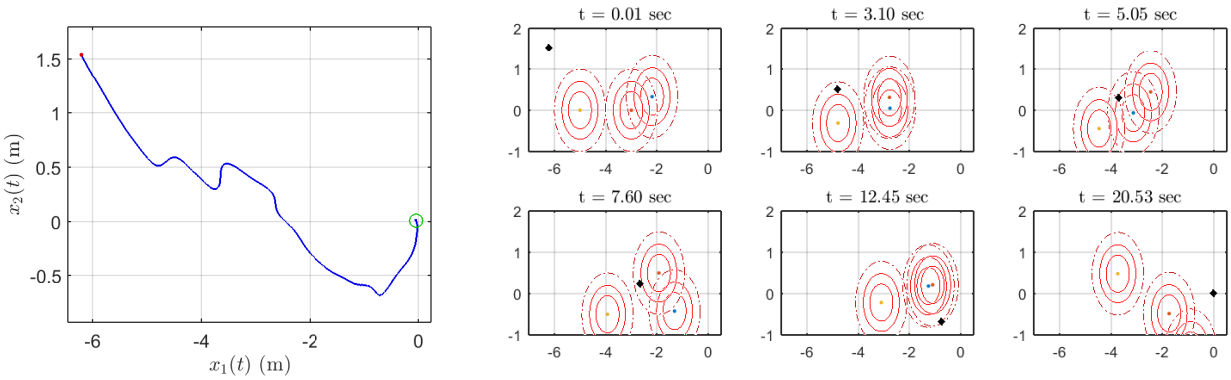
The first experimental validation for the development in Sections 4.1-4.4 are shown in Figures 4-2 and 4-4. Figures 4-2a and 4-2b illustrate that the agent, as well as the agent's control policy, remains bounded around the origin. Figure 4-2b shows that the control of the agent is bounded by $0.5\frac{m}{sec}$ even in the presence of the mobile avoidance regions. The input does not converge to zero because of aerodynamic disturbances, when the quadcopter reaches the origin. The critic and actor weight estimates remain bounded and converge to steady-state values, as presented in Figure 4-2c and Figure 4-2d. However, because of the state-following nature of the StaF approximation method, the ideal weights are unknown; hence the estimate cannot be compared to their ideal values. Even though the agent enters the detection region as shown by Figure 4-3 and Figure 4-4a, the developed method drives the agent away from the avoidance regions and towards the origin. When encountering avoidance region z_2 between the 8^{th} and 12^{th} seconds, the agent was able to maneuver around the avoidance region without collision despite multiple encounters with it because the avoidance region was moving close to the origin and obstructing the path.



(a) The agent phase-space portrait (left) and the positions of the agent and avoidance regions (right) for the first experiment.

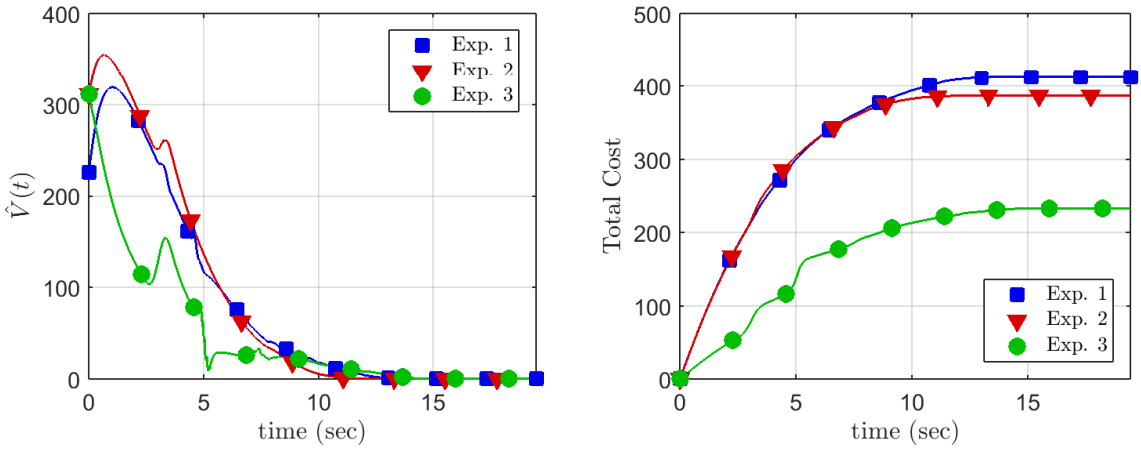


(b) The agent phase-space portrait (left) and positions of the agent and avoidance region (right) for the second experiment.



(c) The agent phase-space portrait (left) and positions of the agent and avoidance region (right) for the third experiment.

Figure 4-4. The phase-space portrait for the agent and the positions of the agent and avoidance regions for each experiment. In each figure, the left plot shows the agent's phase-space portrait where the green circle is the agent's final position. The plots on the right of each figure show the agent's and avoidance regions positions at certain time instances where the diamond represents the agent state and the circles represent the avoidance regions.



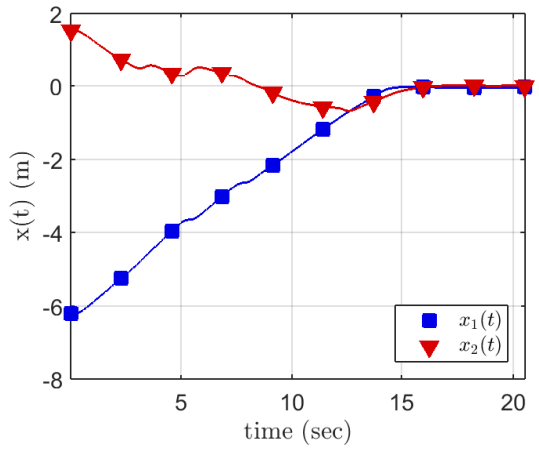
(a) Approximate value function.

(b) Total cost.

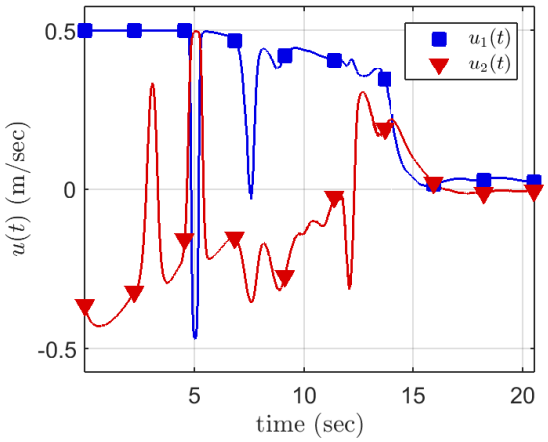
Figure 4-5. The approximate value functions and total costs for the three experiments.

The second and third experiments were performed to validate the development in Section 4.5 with the results shown in Figures 4-4-4-6. Specifically, the second experiment was performed using similar conditions and parameters as in the first experiment. Figure 4-4b indicates that the agent is capable of adjusting its path when it encounters the avoidance regions and the agent is regulated to the origin without colliding with the avoidance regions. The approximate value function and total cost for the first two experiments are shown in Figure 4-5. Both experiments resulted in similar costs and approximate value functions. Specifically, Figure 4-5a shows that the approximate value function remains positive and converges to zero when the agent reaches the origin.

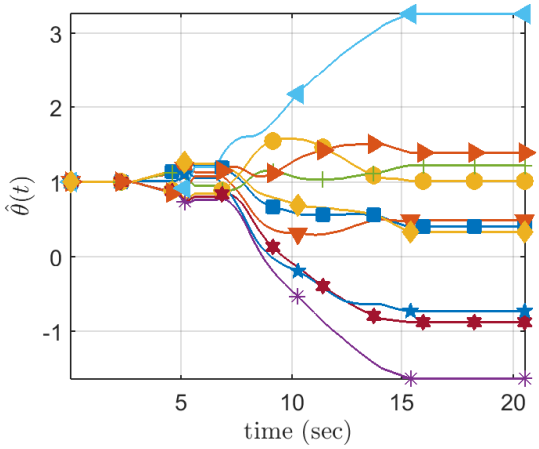
Furthermore, the third experiment extends the second experiment further by substituting one of the autonomous avoidance regions for a non-autonomous one. Specifically, a manually controlled avoidance region is used, which is controlled to approach the agent throughout the experiment. Figures 4-4c-4-6 show the results of the experiment. In Figure 4-4c, the agent is forced away from the direction of the origin, but still manages to redirect itself without colliding with the avoidance regions. The approximate value function and total cost for the third experiment are also shown in



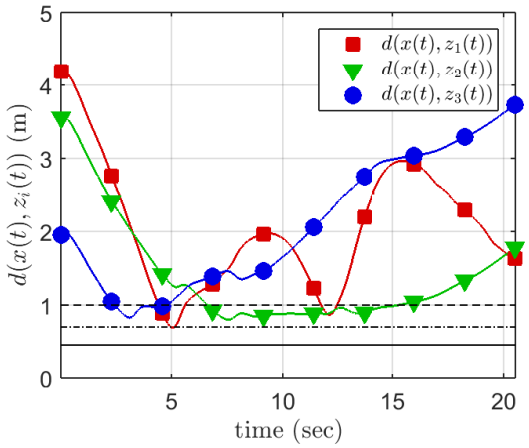
(a) The agent states.



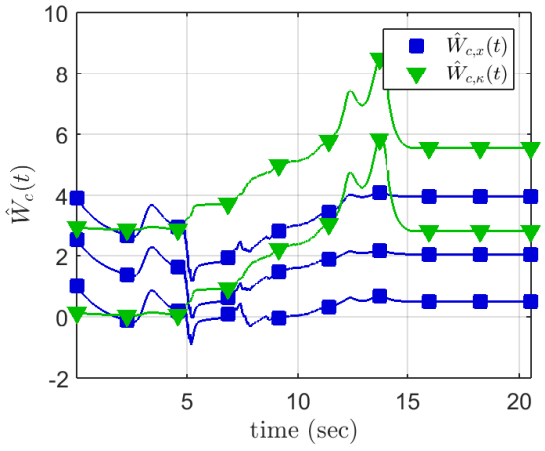
(b) The agent approximate optimal input.



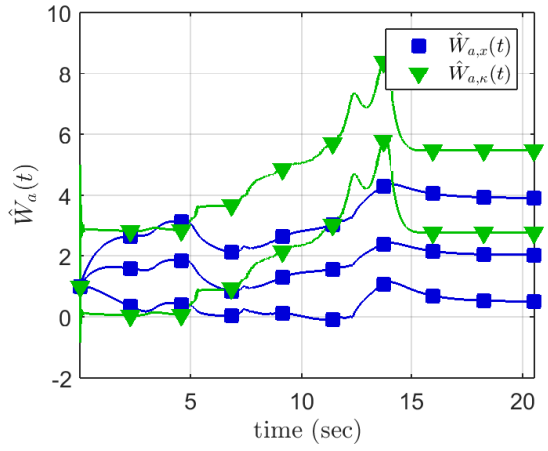
(c) The estimates of θ.



(d) The distance between the agent and avoidance regions.



(e) The critic weight estimates.



(f) The actor weight estimates.

Figure 4-6. The states, control policy, and weight estimates of the agent are shown in addition to the distances between the agent and each avoidance region center for the third experiment.

Figure 4-5. Since one of the avoidance regions was remotely controlled, it's trajectory was nonautonomous; hence, the agent's trajectory differed when interacting with it and the applied control policy did not saturate as much compared to the first experiment, resulting in a smaller total cost. Figures 4-6a shows that the agent is regulated to the origin and that its state is adjusted online in real-time by the input, as shown in Figure 4-6b, when it encounters the avoidance regions. The input remains bounded by the controller saturation of $0.5\frac{m}{sec}$, and converges to a small bounded residual of the origin. The estimates of the unknown weights θ are shown in Figure 4-6c, which remain bounded, but since the ideal basis is unknown and the ideal weights are unknown, the estimates cannot be compared to the actual weights. Figure 4-6d displays the distance between the agent and each avoidance region center, and shows that the agent does not get within r_a of the avoidance regions. Moreover, as soon as the agent gets within $\bar{r}$ of the avoidance region, it moves away from z_i. Additionally, when the agent detects the avoidance region, i.e. $\|x - z_i\| \leq r_d$, the control policy is adjusted, which can be seen from Figure 4-6b and Figure 4-6d. Moreover, the critic and actor weights estimates using the transformation in Section 4.5 are shown in Figure 4-6e and Figure 4-6f, respectively. The figures show that the estimates remain bounded and converge to steady-state values. Similar to the first experiment, the ideal weights are unknown, thus the weight estimates cannot be compared to the ideal weights.

The results in Figures 4-2-4-6 show that the developed method is capable of handling uncertain dynamic avoidance regions while regulating an autonomous agent. The agent locally detects the avoidance regions and then adjusts its path online.

4.7 Concluding Remarks

An online approximate path-planning strategy in the presence of mobile avoidance regions is developed. Because the avoidance regions need to only be known inside a detection radius, they are modeled using local dynamics. Since the avoidance regions are coupled with the agent in the HJB, the basis of the approximation also uses the

avoidance region state when approximating the value function. Because the states are not always known, a scheduling function is used to turn off the basis, which then stops updating the weight approximations for the avoidance regions when they are not detected. Theorem 4.1 showed the UUB of the states and that the states of the coupled system remain outside of the avoidance set. An extension to systems with uncertain dynamics and an unknown number of avoidance regions was presented, and Theorem 4.2 summarized the overall stability for the system with uncertainties. Three experiments were performed which demonstrated successful implementation of the developed path-planning and avoidance region evasion strategy.

CHAPTER 5
APPROXIMATE OPTIMAL INFLUENCE OVER AN AGENT THROUGH AN UNCERTAIN INTERACTION DYNAMIC

An approximate optimal indirect regulation problem is considered for two nonlinear uncertain agents. An pursuing agent is tasked with optimally intercepting and directing a roaming agent to a goal location. The roaming agent is not directly controlled by the pursuer agent, but instead moves based on some uncertain interaction dynamic. A virtual controller designed to yield a desired influence on the roaming agent, and ADP is used to develop an approximate optimal solution. Because system uncertainties are considered in both agents, ICL is used to identify uncertain dynamics. A Lyapunov-based stability analysis is performed which proves the closed-loop pursuing and roaming agent systems are UUB. Simulation and experimental results are provided to demonstrate the performance of the developed method.

5.1 Problem Formulation

In the subsequent development, the goal is to regulate a roaming agent to a desired user-defined goal location.[1] However, the roaming agent may not know where the goal location is or may not be cooperating to go there. The influencing agent knows the goal location, and simultaneously is tasked to optimally intercept and escort the roaming agent through an interaction dynamic [54, 55, 93].

Furthermore, consider a roaming agent governed by the drift dynamics

$$\dot{z}(t) = f(z(t), \eta(t)), \tag{5–1}$$

where $z : \mathbb{R}_{\geq t_0} \to \mathbb{R}^n$ is the roaming agent's state, $\eta : \mathbb{R}_{\geq t_0} \to \mathbb{R}^n$ denotes the influencing agent's state, $t_0 \in \mathbb{R}_{\geq 0}$ is the initial time, and $f : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ is an uncertain locally

[1] The influencing agent in this work is synonymous with predator, pursuing, or herding agents, while the roaming agent is synonymous with prey, evading, or target agents in works such as [43–46, 50, 54, 55].

Lipschitz function. The dynamics in (5–1) are not directly controllable; however, (5–1) can be influenced through interaction with the controlled pursuing agent governed by the uncertain dynamics

$$\dot{\eta}(t) = h(z(t), \eta(t)) + g(\eta(t)) u(t), \tag{5–2}$$

where $h : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ is an unknown locally Lipschitz function representing the influencing agent drift dynamics, $g : \mathbb{R}^n \to \mathbb{R}^{n \times m_\eta}$ is the known control effectiveness matrix, and $u : \mathbb{R}_{\geq t_0} \to \mathbb{R}^{m_\eta}$ is the influencing agent's control input.

Assumption 5.1. The control effectiveness matrix $g(\eta)$ is bounded and full column rank for all $\eta \in \mathbb{R}^n$, and $g^+ : \mathbb{R}^n \to \mathbb{R}^{m_\eta \times n}$ is a bounded and locally Lipschitz pseudo inverse defined as $g^+ \triangleq \left(g^T g\right)^{-1} g^T$ [106].

Assumption 5.2. There exists class $\mathcal{K}$ functions $\overline{\alpha}_1, \overline{\alpha}_2 : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ such that the uncertain dynamics in (5–1) can be bounded as $\|f(z(t), \eta(t))\| \leq \overline{\alpha}_1(\|z(t) - \eta(t)\|) + \overline{\alpha}_2(\|z(t) - z_g\|)$, where $z_g \in \mathbb{R}^n$ is a fixed goal location.[2]

To quantify the objective, a regulation error $e_z : \mathbb{R}_{\geq t_0} \to \mathbb{R}^n$ is defined as

$$e_z(t) \triangleq z(t) - z_g. \tag{5–3}$$

Additional error system development is motivated by backstepping approaches, where the agent control input is designed based on a unique error system development that requires both the influencing and roaming agent errors to converge to the goal. Specifically, an auxiliary error, denoted by $e_\eta : \mathbb{R}_{\geq t_0} \to \mathbb{R}^n$, is defined as

$$e_\eta(t) \triangleq \eta(t) - \eta_d(t), \tag{5–4}$$

[2] Assumption 5.2 indicates that the dynamics of the roaming agent in (5–1) depend on the distance between the influencing and roaming agents and the distance between the roaming agent and the goal location. The roaming agent dynamics in results such as [54, 55], can be shown to satisfy Assumption 5.2.

where $\eta_d : \mathbb{R}_{\geq t_0} \to \mathbb{R}^n$ is a desired virtual state. Because the influencing agent's state $\eta(t)$ may be non-affine in the roaming agent dynamics in (5–1), the aim of the virtual state $\eta_d(t)$ is to minimize the regulation error in (5–3). To quantify this aspect, another auxiliary error, denoted by $e_d : \mathbb{R}_{\geq t_0} \to \mathbb{R}^n$, is defined as

$$e_d(t) \triangleq \eta_d(t) - z_g - k_d e_z(t), \tag{5–5}$$

where $k_d \in \mathbb{R}$ is positive constant control gain. The control gain k_d is generally selected to be greater such that $k_d \geq 1$. The virtual state $\eta_d(t)$ is injected into (5–5) with the goal of regulating $e_d(t)$. Based on (5–5) and the subsequent analysis, the time-derivative of $\eta_d(t)$ is designed as

$$\dot{\eta}_d(t) \triangleq \mu_d(t), \tag{5–6}$$

where $\mu_d : \mathbb{R}_{\geq t_0} \to \mathbb{R}^n$ is a subsequently designed virtual input.

Remark 5.1. In this chapter, single integrator dynamics are used for the virtual state for simplicity. However, the virtual state can also evolve according dynamics such as $\dot{\eta}_d(t) \triangleq -A_d \eta_d(t) + B_d \mu_d(t)$, where $A_d, B_d \in \mathbb{R}^{n \times n}$ are PD matrices.

Remark 5.2. It will be shown in Theorem 5.2 that the states $e_z(t)$, $e_d(t)$, and $e_\eta(t)$, and virtual controller $\mu_d(t)$ will converge to a neighborhood containing the origin. Hence, the virtual state $\eta_d(t)$ will converge to a region of the origin, implying that the roaming agent will be regulated to a neighborhood of the desired location.

Remark 5.3. The error signals in (5–4) and (5–5) have been modified from the preliminary result in [93], and resemble those of backstepping approaches such as [55]; however, compared to [55] optimality is considered for the overall system in this result.

After taking the time-derivative of (5–5) and using (5–1), and (5–3)-(5–6), the error dynamics for $e_d(t)$ are

$$\dot{e}_d(t) = -k_d f(z(t), \eta(t)) + \mu_d(t).$$

To determine the error dynamics for $e_\eta(t)$, (5–2) and (5–6) are substituted into the time-derivative of (5–4) to obtain

$$\dot{e}_\eta(t) = h(z(t), \eta(t)) + g(\eta(t))\mu_\eta(t) + g(\eta(t))u_d(t) - \mu_d(t), \tag{5–7}$$

where $\mu_\eta(t) : \mathbb{R}_{\geq t_0} \to \mathbb{R}^{m_\eta}$ is defined as $\mu_\eta(t) \triangleq u(t) - u_d(t)$, and $u_d(t) : \mathbb{R}_{\geq t_0} \to \mathbb{R}^{m_\eta}$ denotes a desired input. Based on (5–7) and the subsequent stability analysis, the desired input is designed as

$$u_d(t) = g(\eta_d(t))^+ (\mu_d(t) - h(z(t), \eta_d(t))). \tag{5–8}$$

Substituting (5–8) into (5–7) yields the following closed-loop system

$$\begin{aligned}\dot{e}_\eta(t) = {} & h(z(t), \eta(t)) + g(\eta(t))\left(\mu_\eta(t) - g^+(\eta_d(t))h(z(t), \eta_d(t))\right) \\ & + \left(g(\eta(t))g^+(\eta_d(t)) - I_n\right)\mu_d(t).\end{aligned}$$

To formulate the optimal control problem such that the errors in (5–3)-(5–5) are minimized, the influencing and roaming agent states are transformed. To facilitate this transformation, let $x(t) \triangleq \left[e_z^T(t), e_d^T(t), e_\eta^T(t)\right]^T$ and $x_d(t) \triangleq \left[e_z^T(t), e_d^T(t), 0_{1\times n}\right]^T$ denote the concatenated state and desired concatenated state, respectively. In addition, define the mappings $s_1, s_2 : \mathbb{R}^{3n} \to \mathbb{R}^n$ as $s_1(x(t)) \triangleq e_z(t) + z_g$, and $s_2(x(t)) \triangleq e_\eta(t) + e_d(t) + k_d e_z(t) + z_g$. Using (5–3)-(5–5), the roaming and influencing agent states are represented as $z(t) = s_1(x(t))$ and $\eta(t) = s_2(x(t))$ respectively.[3]

[3] Using the mappings s_1 and s_2, the bounds in Assumption 5.2 can be represented such as $\|f(s_1(x(t)), s_2(x(t)))\| \leq \overline{\alpha}_1(\|(k_d - 1)e_z(t) + e_\eta(t) + e_d(t)\|) + \overline{\alpha}_2(\|e_z(t)\|)$.

Using these relationships, a composite autonomous error system can be written as

$$\dot{x}(t) = F(x(t)) + G(x(t))\mu(t), \tag{5–9}$$

where $\mu(t) \triangleq \begin{bmatrix} \mu_\eta^T(t) & \mu_d^T(t) \end{bmatrix}^T \in \mathbb{R}^m$ is the total vector of policies with $m = m_\eta + n$, while $F : \mathbb{R}^{3n} \to \mathbb{R}^{3n}$ and $G : \mathbb{R}^{3n} \to \mathbb{R}^{3n \times m}$ are defined as

$$F(x(t)) \triangleq \begin{bmatrix} f(s_1(x(t)), s_2(x(t))), \\ -k_d f(s_1(x(t)), s_2(x(t))), \\ h(s_1(x(t)), s_2(x(t))) - F_{sd}(x(t)), \end{bmatrix},$$

and

$$G(x(t)) \triangleq \begin{bmatrix} 0_{n \times m_\eta}, & 0_{n \times n}, \\ 0_{n \times m_\eta}, & I_n, \\ g(s_2(x(t))), & G_{sd}(x(t)), \end{bmatrix},$$

where

$$F_{sd}(x(t)) \triangleq g(s_2(x(t)))\, g^+(s_2(x_d(t)))\, h(s_1(x(t)), s_2(x_d(t))),$$

and $G_{sd}(x(t)) \triangleq g(s_2(x(t)))\, g(s_2(x_d(t)))^+ - I_n$.

5.1.1 Optimal Control Development

Given (5–9), the goal is to design controllers $\mu_d(t)$ and $\mu_\eta(t)$ to minimize the cost function

$$J(x, \mu) \triangleq \int_{t_0}^{\infty} r(x(\tau), \mu(\tau))\, d\tau, \tag{5–10}$$

subject to (5–9), where $r : \mathbb{R}^{3n} \times \mathbb{R}^m \to \mathbb{R}_{\geq 0}$ is the instantaneous cost defined as

$$r(x, \mu) \triangleq Q(x) + P(x) + \Psi(\mu). \tag{5–11}$$

In (5–11), $Q : \mathbb{R}^{3n} \to \mathbb{R}_{\geq 0}$ is a user-defined continuous PD function (e.g., $x^T Q_x x$ where $Q_x \in \mathbb{R}^{3n \times 3n}$ is a PD matrix) which can be bounded such as $\underline{q}\|x\|^2 \leq Q(x) \leq \overline{q}\|x\|^2$ for all $x \in \mathbb{R}^{3n}$, where $\underline{q}, \overline{q} \in \mathbb{R}_{>0}$; $\Psi(\mu) \triangleq \mu^T R \mu$ where $R = \text{diag}\{R_\eta, R_d\}$, where $R_\eta \in \mathbb{R}^{m_\eta \times m_\eta}$ and $R_d \in \mathbb{R}^{n \times n}$ are user-defined PD symmetric weighting matrices; and

$P : \mathbb{R}^{3n} \to \mathbb{R}$ is a user-defined continuous PSD penalty function such that $P(x) = 0$ when $\|s_1(x) - s_2(x)\| \leq r_a(x)$ and $P(x) > 0$ when $\|s_1(x) - s_2(x)\| > r_a(x)$, where $r_a : \mathbb{R}^{3n} \to \mathbb{R}_{\geq 0}$ is a design parameter.

Remark 5.4. Examples of functions that satisfy the conditions for $P(x)$ include $P(x) = e^{\frac{1}{2\alpha}(p_x(x))^2} - 1$, $P(x) = \alpha (p_x(x))^2$, $P(x) = \alpha \ln\left(\cosh(p_x(x))^2\right)$, where $p_x(x) \triangleq \max\left\{0, \|(1-k_d)e_z - e_\eta - e_d\|^2 - r_a(x)^2\right\}$, and $\alpha \in \mathbb{R}_{>0}$, or even piecewise continuous smooth functions that saturate at a constant.

The optimal value function, denoted by $V^* : \mathbb{R}^{3n} \to \mathbb{R}_{\geq 0}$, is expressed as

$$V^*(x(t)) = \inf_{\mu(\tau)\in U \left| \tau \in \mathbb{R}_{\geq t}\right.} \int_t^\infty r(x(\tau), \mu(\tau))\, d\tau. \tag{5–12}$$

The HJB equation which characterizes the optimal value function is given by

$$0 = \nabla V^*(x(t))\left(F(x(t)) + G(x(t))\mu^*(x(t))\right) + r(x(t), \mu^*(x(t))), \tag{5–13}$$

with $V^*(0) = 0$, where $\mu^* : \mathbb{R}^{3n} \to \mathbb{R}^m$ denotes the admissible (see Definition 2.1) optimal input policy, which is determined from (5–13) as

$$\mu^*(x) = -\frac{1}{2} R^{-1} G(x)^T (\nabla V^*(x))^T. \tag{5–14}$$

5.2 Approximate Optimal Control

5.2.1 System Identification

Similar to previous chapters, the HJB in (5–13) and optimal controller in (5–14) require knowledge of both the drift dynamics $k_d f(z(t), \eta(t))$ and $h(z(t), \eta(t))$. Since these function are unknown, we approximately minimize the cost function in (5–10) while simultaneously learning these functions. Various methods could be employed to learn the functions (cf., [58, 81, 84, 99, 100]). The following is based on the ICL strategy in [58]. Using the universal function approximation property [97], (5–1) and (5–2) can be represented as

$$\dot{\breve{x}}(t) = S(x(t))\theta + \varepsilon(x(t)) + \breve{G}(x(t), u(t)), \tag{5–15}$$

where $\breve{x}(t) \triangleq [k_d z(t),\ \eta(t)]^T \in \mathbb{R}^{2\times n}$, $\breve{G}(x(t), u(t)) \triangleq \begin{bmatrix} 0_{n\times 1}, & g(x(t))u(t) \end{bmatrix}^T \in \mathbb{R}^{2\times n}$, $\theta \triangleq \begin{bmatrix} \theta_z^T & \theta_\eta^T \end{bmatrix}^T \in \mathbb{R}^{p\times n}$, $S(x) \triangleq \begin{bmatrix} S_z^T(x(t)) & 0_{1\times p_\eta} \\ 0_{1\times p_z} & S_\eta^T(x(t)) \end{bmatrix} \in \mathbb{R}^{2\times p}$, and $\varepsilon(x(t)) \triangleq \begin{bmatrix} \varepsilon_z(x(t)) & \varepsilon_\eta(x(t)) \end{bmatrix}^T \in \mathbb{R}^{2\times n}$. In (5–15), the weights $\theta_j \in \mathbb{R}^{p_j\times n}$ are unknown, the basis functions $S_j : \mathbb{R}^{3n} \to \mathbb{R}^{p_j}$ are user-defined, and $\varepsilon_j : \mathbb{R}^{3n} \to \mathbb{R}^n$ is the function approximation error for $j = \{z, \eta\}$, and $p = p_z + p_\eta$ denotes the total number of rows of θ.[4] [5]

Remark 5.5. If $h(z(t), \eta(t)) = 0_{n\times 1}$ in (5–2), then the terms in (5–15) can be reduced to $\breve{x}(t) \triangleq [k_d z(t)]^T \in \mathbb{R}^{1\times n}$, $\breve{G}(x(t), u(t)) \triangleq \begin{bmatrix} 0_{n\times 1} \end{bmatrix}^T \in \mathbb{R}^{1\times n}$, $\theta \triangleq \begin{bmatrix} \theta_z^T \end{bmatrix}^T \in \mathbb{R}^{p_z\times n}$, $S(x) \triangleq \begin{bmatrix} S_z^T(x(t)) \end{bmatrix} \in \mathbb{R}^{1\times p_z}$, and $\varepsilon(x(t)) \triangleq \begin{bmatrix} \varepsilon_z(x(t)) \end{bmatrix}^T \in \mathbb{R}^{1\times n}$, respectively.

Assumption 5.3. There exist $\overline{\theta},\ \overline{S},\ \overline{\varepsilon} \in \mathbb{R}_{>0}$ such that $\|\theta\| \le \overline{\theta}$, $\sup_{x\in\chi} \|S(x)\| \le \overline{S}$, and $\sup_{x\in\chi} \|\varepsilon(x)\| \le \overline{\varepsilon}$ [18, 98].

Based on the ICL strategy in [58], let $\Delta t_\theta \in \mathbb{R}_{>0}$ denote an integration time-window, where the integral of (5–15) at time $t_i \in [\Delta t_\theta, t]$ can be represented as $\breve{x}(t_i) - \breve{x}(t_i - \Delta t_\theta) = \mathcal{S}_i\theta + \mathcal{E}_i + \mathcal{G}_i$ where $\mathcal{S}_i = \mathcal{S}(t_i) \triangleq \int_{t_i-\Delta t_\theta}^{t_i} S(x(\tau))\, d\tau$, $\mathcal{E}_i = \mathcal{E}(t_i) \triangleq \int_{t_i-\Delta t_\theta}^{t_i} \varepsilon(x(\tau))\, d\tau$, and $\mathcal{G}_i = \mathcal{G}(t_i) \triangleq \int_{t_i-\Delta t_\theta}^{t_i} \breve{G}(x(\tau), u(\tau))\, d\tau$. A least-squares based

[4] The unknown weights θ_z and θ_η can be estimated independently using separate update laws. To alleviate redundancy, a combined approximation method is presented in this chapter.

[5] If an exact basis is known for both agent dynamics, then $\varepsilon(x(t)) = 0_{2\times n}$.

parameter estimate update law is designed as

$$\dot{\hat{\theta}}(t) = k_\theta \Gamma_\theta(t) \sum_{i=1}^{M} \mathcal{S}^T(t_i) \left(\breve{x}(t_i) - \breve{x}(t_i - \Delta t_\theta) - \mathcal{G}(t_i) - \mathcal{S}(t_i) \hat{\theta} \right), \tag{5–16}$$

$$\dot{\Gamma}_\theta(t) = \beta_\theta \Gamma_\theta(t) - k_\theta \Gamma_\theta(t) \sum_{i=1}^{M} \mathcal{S}^T(t_i) \mathcal{S}(t_i) \Gamma_\theta(t), \tag{5–17}$$

where $k_\theta, \beta_\theta \in \mathbb{R}_{>0}$ is an update gain and forgetting factor, respectively, and $M \in \mathbb{Z}_{>0}$ is the number of data points collected in the history stack.

Remark 5.6. Generally, the number of data points (i.e., the size of the history stack) needs to be at least $\frac{p}{2}$ to have enough information for $\sum_{i=1}^{M} \mathcal{S}_i^T \mathcal{S}_i$ to become full rank as stated in Assumption 5.4, i.e., $M \geq \frac{p}{2}$. While M may be unknown a priori, it can be determined by checking Assumption 5.4 online.

Assumption 5.4. There exists $T_1 \in \mathbb{R}_{>0}$ such that $T_1 > \Delta t_\theta$ and a strictly positive constant $\lambda_1 \in \mathbb{R}_{>0}$ where $\lambda_1 I_p \leq \sum_{i=1}^{M} \mathcal{S}_i^T \mathcal{S}_i$, $\forall t \geq T_1$ [58].

Provided $\lambda_{\min}\left\{\Gamma_\theta^{-1}(t_0)\right\} > 0$, and Assumption 5.4 is satisfied, Γ_θ satisfies $\underline{\Gamma}_\theta I_p \leq \Gamma_\theta(t) \leq \overline{\Gamma}_\theta I_p$, using similar arguments to [101, Corollary 4.3.2], where $\underline{\Gamma}_\theta, \overline{\Gamma}_\theta \in \mathbb{R}_{>0}$. Let $Z_\theta(t) = \text{vec}\left(\tilde{\theta}(t)\right)$ denote a vector of parameter estimate errors defined in Section 2.1. Also let $V_\theta : \mathbb{R}^{np} \times \mathbb{R}_{\geq t_0} \to \mathbb{R}$ be defined as the candidate Lyapunov function

$$V_\theta(Z_\theta, t) \triangleq \frac{1}{2} \text{tr}\left(\tilde{\theta}^T \Gamma_\theta^{-1}(t) \tilde{\theta}\right), \tag{5–18}$$

which can be bounded as $\frac{1}{2\overline{\Gamma}_\theta} \|Z_\theta\|^2 \leq V_\theta(Z_\theta, t) \leq \frac{1}{2\underline{\Gamma}_\theta} \|Z_\theta\|^2$, for all $t \in \mathbb{R}_{\geq t_0}$ and $Z_\theta \in \mathbb{R}^{np}$.

Theorem 5.1. *Provided Assumptions 5.3 and 5.4 are satisfied, the adaptive update laws in (5–16) and (5–17) ensure that the estimation error $\tilde{\theta}$ remains bounded for all $t \geq T_1$ such that*

$$\|Z_\theta(t)\| \leq c_\Gamma \sqrt{c_M e^{-\lambda_\theta (t - T_1)} + \left(1 - e^{-\lambda_\theta (t - T_1)}\right) c_B}, \tag{5–19}$$

where $c_\Gamma \triangleq \sqrt{\frac{\overline{\Gamma}_\theta}{\underline{\Gamma}_\theta}}$, $\lambda_\theta \triangleq \frac{k_\theta c_{\theta 2} \underline{\Gamma}_\theta}{2}$, $c_{\theta 1} \triangleq \frac{\beta_\theta}{k_\theta \overline{\Gamma}_\theta}$, $c_{\theta 2} \triangleq c_{\theta 1} + \lambda_1$, $c_M \triangleq \|Z_\theta(t_0)\|^2 + \frac{4\upsilon_1^2}{c_{\theta 1}^2}$, $c_B \triangleq \frac{4\upsilon_1^2}{c_{\theta 2}^2}$, *and* $\upsilon_1 \triangleq \sup_{t \in \mathbb{R}_{\geq 0}} \left\| \sum_{i=1}^{M} \mathcal{S}_i^T \mathcal{E}_i \right\|$.

Proof. Taking the time-derivative of (5–18), substituting for (5–16) and (5–17), using the fact that for $t < T_1$, $\sum_{i=1}^{M} \mathcal{S}_i^T \mathcal{S}_i \geq 0$,

$$\dot{V}_\theta (Z_\theta, t) \leq -\frac{1}{2} k_\theta c_{\theta 1} \|Z_\theta\|^2 + k_\theta \|Z_\theta\| \upsilon_1. \tag{5–20}$$

Completing the squares, using the bounds on (5–18), and invoking the Comparison Lemma [104, Lemma 3.4] yields

$$V_\theta (Z_\theta (t), t) \leq V_\theta (Z_\theta (t_0), t_0) e^{-\lambda_\theta \frac{k_\theta c_{\theta 1} \underline{\Gamma}_\theta}{2} (t - t_0)} + \left(1 - e^{-\frac{k_\theta c_{\theta 1} \underline{\Gamma}_\theta}{2} (t - t_0)}\right) \frac{2\upsilon_1^2}{\underline{\Gamma}_\theta c_{\theta 1}^2}, \tag{5–21}$$

for all $t \in [t_0, T_1)$. Then, $\|Z_\theta (t)\|^2 \leq \frac{\overline{\Gamma}_\theta}{\underline{\Gamma}_\theta} \left(\|Z_\theta (t_0)\|^2 + \frac{4\upsilon_1^2}{c_{\theta 1}^2}\right)$ follows for all $t \in \mathbb{R}_{\geq t_0}$.

After $\sum_{i=1}^{M} \mathcal{S}_i^T \mathcal{S}_i$ becomes full rank, (5–16), (5–17), and Assumption 5.4 are used in the time-derivative of (5–18) to yield

$$\dot{V}_\theta (Z_\theta, t) \leq -\frac{1}{4} k_\theta c_{\theta 2} \|Z_\theta\|^2 + \frac{k_\theta \upsilon_1^2}{c_{\theta 2}}, \tag{5–22}$$

for all $t \geq T_1$. Using the Comparison Lemma [104, Lemma 3.4], $\forall t \geq T_1$

$$V_\theta (Z_\theta (t), t) \leq V_\theta (Z_\theta (T_1), T_1) e^{-\lambda_\theta (t - T_1)} + \left(1 - e^{-\lambda_\theta (t - T_1)}\right) \frac{c_B}{2\underline{\Gamma}_\theta}. \tag{5–23}$$

From (5–21), $V_\theta (Z_\theta (T_1), T_1) \leq V_\theta (Z_\theta (t_0), t_0) e^{-\lambda_\theta \frac{k_\theta c_{\theta 1} \underline{\Gamma}_\theta}{2} (t - t_0)} + \frac{2\upsilon_1^2}{\underline{\Gamma}_\theta c_{\theta 1}^2}$ follows, and using (5–23) along with the bounds $\frac{1}{2\overline{\Gamma}_\theta} \|Z_\theta\|^2 \leq V_\theta (Z_\theta, t) \leq \frac{1}{2\underline{\Gamma}_\theta} \|Z_\theta\|^2$ results in (5–19). □

Remark 5.7. After $\sum_{i=1}^{M} \mathcal{S}_i^T \mathcal{S}_i$ becomes full rank, as $t \to \infty$, the residual bound in (5–23) (i.e., $\frac{c_B}{2\underline{\Gamma}_\theta}$) is smaller compared to the residual bound in (5–21) (i.e., $\frac{2\upsilon_1^2}{\underline{\Gamma}_\theta c_{\theta 1}^2}$). This is because the term c_B depends on $c_{\theta 2}$ and $c_{\theta 2} > c_{\theta 1}$. In addition, data selection and purging techniques such as [81] and [77] can be used to select data which reduces the residuals even further.

5.2.2 Value Function Approximation

The value function $V^* (x)$, which is unknown, can be approximated via computationally efficient StaF kernels such as in Chapter 4 [1, 95, 96]. To facilitate the following development, let $\overline{B_r (x)}$ be defined as the closure of an open ball centered at $x \in \mathbb{R}^{3n}$

with radius $r \in \mathbb{R}_{>0}$, and let $\chi \subseteq \mathbb{R}^{3n}$ be a compact set. Using state-following centers, $c : \chi \to \chi^L$, centered around $x \in \chi$ such that $c(x) \in \overline{B_r(x)}$, the value function in (5–12) can be represented as

$$V^*(y) = W(x)^T \sigma(y, c(x)) + \epsilon_v(x, y), \tag{5–24}$$

where $y = \left[y_{e_z}^T, y_{e_d}^T, y_{e_\eta}^T\right] \in \overline{B_r(x)}$ represents a composite state vector in the neighborhood of x [1, 95, 96], and the states y_{e_z}, y_{e_d}, and y_{e_η} represent states in the neighborhoods of e_z, e_d, and e_η, respectively (i.e., $y_{e_z} \in \overline{B_r(e_z)}$, $y_{e_d} \in \overline{B_r(e_d)}$, and $y_{e_z} \in \overline{B_r(e_\eta)}$). In (5–24), $W : \chi \to \mathbb{R}^L$ is a vector of continuously differentiable ideal StaF weight functions, $\sigma : \chi \times \chi \to \mathbb{R}^L$ is a bounded vector of continuously differentiable nonlinear kernels, and $\epsilon_v : \chi \times \chi \to \mathbb{R}$ is a continuously differentiable function approximation error.

Since the ideal StaF weight $W(x)$ and function approximation error are unknown in (5–24), an approximate value function $\hat{V} : \mathbb{R}^{3n} \times \mathbb{R}^{3n} \times \mathbb{R}^L \to \mathbb{R}$ is expressed as

$$\hat{V}\left(y, x, \hat{W}_c\right) = \hat{W}_c^T \sigma(y, c(x)), \tag{5–25}$$

and an approximate policy $\hat{\mu} : \mathbb{R}^{3n} \times \mathbb{R}^{3n} \times \mathbb{R}^L \to \mathbb{R}^m$ is expressed as

$$\hat{\mu}\left(y, x, \hat{W}_a\right) = -\frac{1}{2} R^{-1} G(x)^T \nabla\sigma(y, c(x))^T \hat{W}_a, \tag{5–26}$$

where $\hat{W}_c, \hat{W}_a \in \mathbb{R}^L$ denote the critic and actor weight estimates, respectively. Substituting (5–25) and (5–26) along with the estimate $\hat{\theta}$ into (5–13) results in the BE $\delta : \mathbb{R}^{3n} \times \mathbb{R}^{3n} \times \mathbb{R}^L \times \mathbb{R}^L \times \mathbb{R}^{p \times n} \to \mathbb{R}$ given by

$$\begin{aligned}\delta\left(y, x, \hat{\theta}, \hat{W}_c, \hat{W}_a\right) = \nabla\hat{V}\left(y, x, \hat{W}_c\right)\left(F_1\left(y, \hat{\theta}\right) - F_2\left(y, \hat{\theta}\right) + G(y)\,\hat{\mu}\left(y, x, \hat{W}_a\right)\right) \\ + Q(y) + P(y) + \Psi\left(\hat{\mu}\left(y, x, \hat{W}_a\right)\right),\end{aligned} \tag{5–27}$$

where

$$F_1\left(y,\hat{\theta}\right) \triangleq \begin{bmatrix} \frac{1}{k_d}\hat{\theta}_z^T S_z\left(y\right) \\ -\hat{\theta}_z^T S_z\left(y\right) \\ \hat{\theta}_\eta^T S_\eta\left(y\right) \end{bmatrix},$$

and

$$F_2\left(y,\hat{\theta}\right) \triangleq \begin{bmatrix} 0_{2n\times 1} \\ g\left(y_\eta\right) g^+\left(y_{\eta_d}\right) \hat{\theta}_\eta^T S_\eta\left(\left[y_{e_z}^T, y_{e_d}^T 0_{n\times 1}^T\right]^T\right) \end{bmatrix}.$$

The controller for the influencing agent is $\hat{u}\left(y,x,\hat{\theta},\hat{W}_a\right) = \hat{\mu}_\eta\left(y,x,\hat{W}_a\right) + \hat{u}_d\left(y,x,\hat{\theta},\hat{W}_a\right)$, where $\hat{u}_d\left(y,x,\hat{\theta},\hat{W}_a\right) \triangleq g^+\left(y_{n_d}\right)\left(\hat{\mu}_d\left(y,x,\hat{W}_a\right) - S_\eta\left(\left[y_{e_z}^T,\ y_{e_d}^T,\ 0_{n\times 1}^T\right]^T\right)\hat{\theta}_\eta\right)$, and the approximate optimal terms $\hat{\mu}_\eta\left(y,x,\hat{W}_a\right)$ and $\hat{\mu}_d\left(y,x,\hat{W}_a\right)$ come from $\hat{\mu}\left(y,x,\hat{W}_a\right) = \left[\hat{\mu}_\eta^T\left(y,x,\hat{W}_a\right), \hat{\mu}_d^T\left(y,x,\hat{W}_a\right)\right]^T$ given in (5–26).

5.2.3 Online Learning

At each time instance $t \in \mathbb{R}_{\geq t_0}$, the BE in (5–27) is evaluated at the current state (i.e., $y = x\left(t\right)$), and the current parameter estimate $\hat{\theta}\left(t\right), \hat{W}_c\left(t\right)$, and $\hat{W}_a\left(t\right)$, resulting in the instantaneous BE and influencing agent control policy given as

$$\delta_t\left(t\right) \triangleq \delta\left(x\left(t\right), x\left(t\right), \hat{\theta}\left(t\right), \hat{W}_c\left(t\right), \hat{W}_a\left(t\right)\right), \tag{5–28}$$

and $u\left(t\right) \triangleq \hat{u}\left(x\left(t\right), x\left(t\right), \hat{\theta}\left(t\right), \hat{W}_a\left(t\right)\right)$ respectively. However, if only the BE, given by $\delta_t\left(t\right)$, is used to update the estimate $\hat{W}_c$, then an exciting probing signal would need to be injected into the input $\hat{\mu}\left(t\right)$ (cf. [21, 87, 91, 106]). In contrast to injecting a probing signal, learning via simulation of experience is performed by extrapolating the BE to unexplored states in $\overline{B_r\left(x\left(t\right)\right)}$. Moreover, sets of functions $\left\{x_i : \mathbb{R}^{3n} \times \mathbb{R}_{\geq t_0} \to \mathbb{R}^{3n}\right\}_{i=1}^N$ are selected by the critic such that $x_i\left(x\left(t\right), t\right) \in \overline{B_r\left(x\left(t\right)\right)}$. Then, extrapolated versions of the BE and total input are evaluated at $y = x_i\left(x\left(t\right), t\right)$ as $\delta_{ti}\left(t\right) \triangleq$

$\delta\left(x_i\left(x\left(t\right),t\right),x\left(t\right),\hat{\theta}\left(t\right),\hat{W}_c\left(t\right),\hat{W}_a\left(t\right)\right)$ and $u_i\left(t\right)=\hat{u}\left(x_i\left(x\left(t\right),t\right),x\left(t\right),\hat{\theta}\left(t\right),\hat{W}_a\left(t\right)\right)$, respectively.[6]

The critic aims to find a set of weights that minimize the BE; hence, the critic is updated according to

$$\dot{\hat{W}}_c\left(t\right)=-\Gamma_c\left(t\right)\left(k_{c1}\frac{\omega\left(t\right)}{\rho^2\left(t\right)}\delta_t\left(t\right)+\frac{k_{c2}}{N}\sum_{i=1}^{N}\frac{\omega_i\left(t\right)}{\rho_i^2\left(t\right)}\delta_{ti}\left(t\right)\right), \tag{5–29}$$

$$\dot{\Gamma}_c\left(t\right)=\beta_c\Gamma_c\left(t\right)-\Gamma_c\left(t\right)k_{c1}\frac{\omega\left(t\right)\omega^T\left(t\right)}{\rho^2\left(t\right)}\Gamma_c\left(t\right)-\Gamma_c\left(t\right)\frac{k_{c2}}{N}\sum_{i=1}^{N}\frac{\omega_i\left(t\right)\omega_i^T\left(t\right)}{\rho_i^2\left(t\right)}\Gamma_c\left(t\right), \tag{5–30}$$

where $\rho\left(t\right)\triangleq 1+\gamma_1\omega^T\left(t\right)\omega\left(t\right)$, $\rho_i\left(t\right)\triangleq 1+\gamma_1\omega_i^T\left(t\right)\omega_i\left(t\right)$, and $k_{c1},k_{c2},\gamma_1,\beta_c\in\mathbb{R}_{>0}$ are learning gains,

$$\omega\left(t\right)=\nabla\sigma\left(x\left(t\right),c\left(x\left(t\right)\right)\right)\Big(F_1\big(x\left(t\right),\hat{\theta}\left(t\right)\big)-F_2\big(x\left(t\right),\hat{\theta}\left(t\right)\big)+G\left(x\left(t\right)\right)\hat{\mu}\big(x\left(t\right),x\left(t\right),\hat{W}_a\left(t\right)\big)\Big),$$

and $\omega_i\left(t\right)=\nabla\sigma_i\left(F_{1i}-F_{2i}+G_i\hat{\mu}_i\right)$, with $\nabla\sigma_i\triangleq\nabla\sigma\left(x_i\left(x\left(t\right),t\right),c\left(x\left(t\right)\right)\right)$, $F_{1i}\triangleq F_1\left(x_i\left(x\left(t\right),t\right),\hat{\theta}\left(t\right)\right)$, $F_{2i}\triangleq F_2\left(x_i\left(x\left(t\right),t\right),\hat{\theta}\left(t\right)\right)$, $G_i\triangleq G\left(x_i\left(x\left(t\right),t\right)\right)$, and $\hat{\mu}_i\triangleq\hat{\mu}\left(x_i\left(x\left(t\right),t\right),x\left(t\right),\hat{W}_a\left(t\right)\right)$. Similar to previous chapters, to facilitate learning, off-policy trajectories are selected, which can contain excitation signals to achieve a virtual excitation. Hence, the states x and x_i in this chapter are assumed to satisfy Assumption 4.5 where ω_k is substituted with ω_i.

Using Assumption 4.5 along with $\lambda_{\min}\left\{\Gamma_c^{-1}\left(t_0\right)\right\}>0$, a similar argument to [101, Corollary 4.3.2] can be used to show that $\underline{\Gamma}_c I_L\leq\Gamma_c\left(t\right)\leq\overline{\Gamma}_c I_L$ where $\underline{\Gamma}_c$ and $\overline{\Gamma}_c$ are positive bounds [1].

[6] Compared to Chapter 4 where part of the state vector is extrapolated, in this chapter the entire state is extrapolated.

The actor weight estimate is updated to follow the critic weight estimate as

$$\dot{\hat{W}}_a(t) = -K_a k_{a1}\left(\hat{W}_a(t) - \hat{W}_c(t)\right) - K_a k_{a2}\hat{W}_a(t) + K_a \frac{k_{c1}}{4} G_\sigma^T(t)\hat{W}_a(t)\frac{\omega^T(t)}{\rho^2(t)}\hat{W}_c(t) + K_a \frac{k_{c2}}{4N}\sum_{i=1}^{N} G_{\sigma i}^T(t)\hat{W}_a(t)\frac{\omega_i^T(t)}{\rho_i^2(t)}\hat{W}_c(t), \tag{5–31}$$

where $G_\sigma(t) \triangleq \nabla\sigma(x(t), c(x(t)))\, G_R(x(t)) \cdot \nabla\sigma^T(x(t), c(x(t)))$, $G_{\sigma i}(t) \triangleq \nabla\sigma_i G_i R^{-1} G_i^T \nabla\sigma_i^T$, $G_R(x(t)) \triangleq G(x(t))\, R^{-1} G^T(x(t))$, $k_{a1}, k_{a2} \in \mathbb{R}_{\geq 0}$ are learning gains, and $K_a \in \mathbb{R}^{L\times L}$ is a positive-definite symmetric matrix.

5.3 Stability Analysis

To facilitate the following stability analysis, let $B_\zeta \subset \mathbb{R}^{3n+np+2L}$ denote a closed ball of radius $\zeta \in \mathbb{R}_{>0}$ centered at the origin. By defining the critic and actor weight estimate errors according to Section 2.1 as $\tilde{W}_c$ and $\tilde{W}_a$, respectively, the BEs, $\delta_t(t)$ and $\delta_{ti}(t)$, are

$$\begin{aligned}\delta_t &= -\omega^T\tilde{W}_c + \frac{1}{4}\tilde{W}_a^T G_\sigma \tilde{W}_a - W^T\nabla\sigma\left(\tilde{F}_1 - \tilde{F}_2\right) + \Delta(x),\\ \delta_{ti} &= -\omega_i^T\tilde{W}_c + \frac{1}{4}\tilde{W}_a^T G_{\sigma i}\tilde{W}_a - W^T\nabla\sigma_i\left(\tilde{F}_{1i} - \tilde{F}_{2i}\right) + \Delta(x_i),\end{aligned} \tag{5–32}$$

where $\tilde{F}_1 \triangleq F_1\left(x, \tilde{\theta}\right)$, $\tilde{F}_2 \triangleq F_2\left(x, \tilde{\theta}\right)$, and $\tilde{F}_{1i} \triangleq F_1\left(x_i, \tilde{\theta}\right)$, $\tilde{F}_{2i} \triangleq F_2\left(x_i, \tilde{\theta}\right)$. In (5–32), the functions $\Delta, \Delta_i : \mathbb{R}^n \to \mathbb{R}$ are uniformly bounded over a compact set χ such that $\overline{\|\Delta\|}$ and $\overline{\|\Delta_i\|}$ decrease with decreasing $\|\epsilon_v\|$, $\|\varepsilon\|$, and $\|W\|$.

Let $Z_L \triangleq \left[\begin{array}{cccc} x^T, & \tilde{W}_c^T, & \tilde{W}_a^T, & Z_\theta^T \end{array}\right]^T$ denote the concatenated state vector, and let $V_L : \mathbb{R}^{3n+2L+np} \times \mathbb{R}_{\geq t_0} \to \mathbb{R}$ denote a candidate Lyapunov function defined as

$$V_L(Z_L, t) \triangleq V^*(x) + \frac{1}{2}\tilde{W}_c^T\Gamma_c^{-1}(t)\tilde{W}_c + \frac{1}{2}\tilde{W}_a^T K_a^{-1}\tilde{W}_a + V_\theta(Z_\theta, t), \tag{5–33}$$

which, for class $\mathcal{K}$ functions $\underline{v}_l, \overline{v}_l : \mathbb{R} \to \mathbb{R}_{\geq 0}$, can be bounded as

$$\underline{v}_l(\|Z_L\|) \leq V_L(Z_L, t) \leq \overline{v}_l(\|Z_L\|) \tag{5–34}$$

for all $t \in \mathbb{R}_{\geq t_0}$ and $Z_L \in \mathbb{R}^{3n+2L+np}$.

Theorem 5.2. *Provided Assumptions 4.5 and 5.1-5.4 are satisfied, and*

$$\lambda_{\min}\{H\} > 0, \tag{5–35}$$

$$\sqrt{\frac{\iota}{\kappa}} \leq \underline{v}_l^{-1}\left(\overline{v}_l\left(\zeta\right)\right), \tag{5–36}$$

where

$$H \triangleq \begin{bmatrix} \left(\frac{k_{a1}+k_{a2}}{4} - \varphi_a\right) & -\frac{\varphi_{ac}}{2} & 0 \\ -\frac{\varphi_{ac}}{2} & \frac{\left(\frac{\beta_c}{\Gamma_c}+k_{c2}\varsigma_1\right)}{8} & -\frac{\varphi_{c\theta}}{2} \\ 0 & -\frac{\varphi_{c\theta}}{2} & \frac{k_\theta c_{\theta 1}}{8} \end{bmatrix},$$

and $\kappa, \varphi_a, \varphi_{ac}, \varphi_{c\theta}, \iota \in \mathbb{R}_{>0}$ are defined in Appendix A.3, then the system errors defined in Z_L and approximate policy $\mu\left(t\right)$ are

$$\limsup_{t\to\infty} \left\|Z_L\left(t\right)\right\| \leq \underline{v}_l^{-1}\left(\overline{v}_l\left(\sqrt{\frac{\iota}{\kappa}}\right)\right). \tag{5–37}$$

Proof. Taking the time derivative of (5–33) along the system trajectory, and using the fact that $\dot{V}^*\left(x,t\right) = \nabla V^*\left(F\left(x\right) + G\left(x\right)\mu\right)$, results in

$$\dot{V}_L = \nabla V^*\left(F + G\mu\right) + \dot{V}_\theta\left(Z_\theta, t\right) + \tilde{W}_c^T\Gamma_c^{-1}\left(\dot{W} - \dot{\hat{W}}_c\right) - \frac{1}{2}\tilde{W}_c^T\left(\Gamma_c^{-1}\dot{\Gamma}_c\Gamma_c^{-1}\right)\tilde{W}_c$$
$$+ \tilde{W}_a^T K_a^{-1}\left(\dot{W} - \dot{\hat{W}}_a\right).$$

Substituting (5–13) and using $\dot{W} \triangleq \nabla W\left(x\right)\left(F\left(x\right) + G\left(x\right)\mu\right)$ yields

$$\dot{V}_L = -r\left(x, \mu^*\left(x\right)\right) - \nabla V^* G\mu^* - \frac{1}{2}\tilde{W}_c^T\Gamma_c^{-1}\dot{\Gamma}_c\Gamma_c^{-1}\tilde{W}_c + \tilde{W}_c^T\Gamma_c^{-1}\left(\nabla W\left(F + G\mu\right) - \dot{\hat{W}}_c\right)$$
$$+ \nabla V^* G\mu + \tilde{W}_a^T K_a^{-1}\left(\nabla W\left(F + G\mu\right) - \dot{\hat{W}}_a\right) + \dot{V}_\theta\left(Z_\theta, t\right).$$

Using (5–29) and (5–30), then substituting in (5–11), (5–20), (5–31), and (5–32), and using $\hat{W}_a = W - \tilde{W}_a$, $\hat{W}_c = W - \tilde{W}_c$, bounding, and completing the squares yields

$$\dot{V}_L \leq -\kappa\left\|Z_L\right\|^2 - \kappa\left\|Z_L\right\|^2 + \iota - Z_v^T H Z_v,$$

where $Z_v \triangleq \left[\left\| \tilde{W}_a \right\|, \left\| \tilde{W}_c \right\|, \|Z_\theta\| \right]^T$. Provided the sufficient condition in (5–35) is met, then for all $Z \in B_\zeta$

$$\dot{V}_L \leq -\kappa \|Z_L\|^2, \ \forall \|Z_L\| \geq \sqrt{\frac{\iota}{\kappa}} > 0. \tag{5–38}$$

Using (5–34), (5–36), and (5–38), [104, Theorem 4.18] is invoked to conclude that all trajectories $Z_L(t)$ that satisfy $\|Z_L(t_0)\| \leq \overline{v}_l^{-1}\left(\underline{v}_l^-(\zeta)\right)$, remain bounded for all $t \in \mathbb{R}_{\geq t_0}$ and satisfy (5–37). Since $Z_L \in \mathcal{L}_\infty$, it follows that $x,\ \tilde{W}_c,\ \tilde{W}_a,\ \tilde{\theta} \in \mathcal{L}_\infty$ and therefore $\mu \in \mathcal{L}_\infty$. Furthermore, since $x \in \mathcal{L}_\infty$ and W is a continuous function of x, then $W(x) \in \mathcal{L}_\infty$. Moreover, since $x \in \mathcal{L}_\infty$, it follows that $e_d,\ e_\eta,\ e_z \in \mathcal{L}_\infty$. Using (5–3)-(5–5), $z \in \mathcal{L}_\infty$, and $\eta_d \in \mathcal{L}_\infty$; hence, $\eta,\ (z-\eta) \in \mathcal{L}_\infty$ follows. Finally, since $\mu, \tilde{\theta}, g^+, \eta_d \in \mathcal{L}_\infty$, then $u_d, \hat{\theta} \in \mathcal{L}_\infty$ and $u \in \mathcal{L}_\infty$. □

Remark 5.8. The sufficient condition in (5–35) can be satisfied by increasing the gains k_{a2} and γ_1, and selecting K_a and R with large minimum eigenvalues. In addition, increasing the number of neurons and number of sample points for the system identification, i.e., $p_z \gg n$, $p_\eta \gg n$, and $M \gg p$, and also selecting extrapolation points $x_i(x(t), t)$ so that $\underline{c}_1$ is large will also help ensure the sufficient condition in (5–35) is satisfied.

5.4 Simulation

To demonstrate the performance of the developed method, a two-dimensional simulation is performed for the roaming agent in (5–1) and the influencing agent in (5–2) with $f(z(t), \eta(t)) = (Ae(t) + Be_z(t)) \exp\left(-\frac{1}{2} e(t)^T e(t)\right)$, where $e(t) = z(t) - \eta(t)$, and without a loss of generality $h(z(t), \eta(t)) = 0_{2\times 1}$, $g(\eta(t)) = I_2$, respectively. The unknown parameters to be identified by the ICL update law in (5–16) and (5–17) are $\theta_\eta \triangleq 0_{2\times 2}$, $\theta_z = [k_d A, B]^T$, where $A = \begin{bmatrix} 1 & -0.6 \\ 0.5 & 1.5 \end{bmatrix}$ and $B = \begin{bmatrix} 0.05 & 0 \\ 0.25 & 0.1 \end{bmatrix}$. For parameter identification the basis functions are selected as $S_\eta(x(t)) = e(t)$ and $S_z(x(t)) = \exp\left(-\frac{1}{2} e(t)^T e(t)\right) \times \left[e^T(t),\ e_z^T(t)\right]^T$. The StaF basis is selected as $\sigma(x, c(x)) = [\sigma_1(x, c_1(x)), \ldots, \sigma_7(x, c_7(x))]^T$, where $\sigma_i(x(t), c_i(x(t))) = x^T(t)(x(t) + 0.7\nu(x(t)) d_i)$,

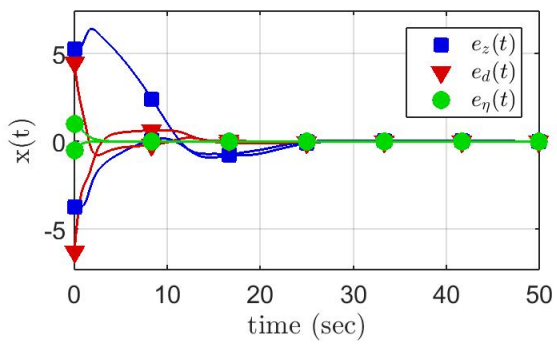
Table 5-1. Initial conditions and parameters selected for the simulation.

Initial conditions at $t_0 = 0$
$z(0) = [-3.75,\ 4.75]^T$, $\eta(0) = [-0.5, 0.5]^T$, $\eta_d(0) = [0, -0.5]^T$, $z_g = [0, -0.5]^T$, $\hat{W}_c(0) = 1 \times 1_{7\times1}$, $\hat{W}_a(0) = 0.75 \times 1_{7\times1}$, $\Gamma_c(0) = 2I_7$, $\hat{\theta}(0) = U[-1, 1] \times 1_{6\times2}$, $\Gamma_\theta(0) = 100I_6$.
Penalizing parameters
$Q(x) = x^T Q_x x$, $P(x) = \left(\max\left\{0, \|(1-k_d)e_z(t) - e_\eta(t) - e_d(t)\|^2 - {r_a}^2\right\}\right)^2$, $Q_x = \text{diag}\{20, 20, 20, 20, 5, 5\}$, $R = 5I_4$, $k_d = 1.2$, $r_a = 0.15$.
Gains and parameters for ADP update laws
$k_{c1} = 0.001$, $k_{c2} = 0.25$, $k_{a1} = 0.25$, $k_{a2} = 0.005$, $\gamma_1 = 0.75$, $\beta_c = 0.001$, $K_a = I_7$, $k_\theta = 0.5$, $\beta_\theta = 2$, $M = 100$, $N = 1$.

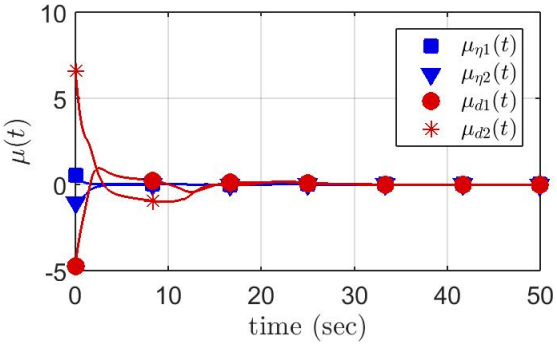
$\nu(x(t)) = \frac{0.7x^T(t)x(t)}{(1+x^T(t)x(t))}$, and d_i are the vertices of a 6-simplex [1, 32, 95]. To perform BE extrapolation, a single trajectory $x_i(x(t), t)$ is selected at random from a uniform distribution over a $\frac{5}{7}\nu(x(t)) \times \frac{5}{7}\nu(x(t))$ square centered at the current state $x(t)$. The selected initial conditions and parameters are provided in Table 5-1.

5.4.1 Discussion

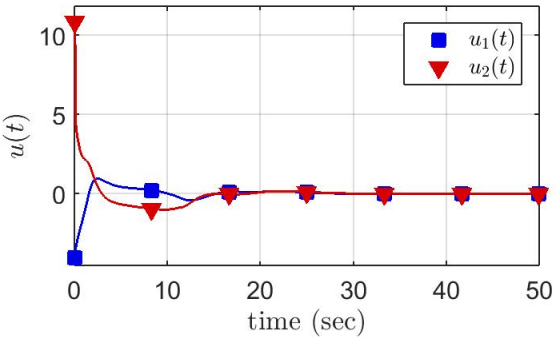
Figures 5-1-5-2 demonstrate that the influencing agent regulates the roaming agent to the goal location z_g. Figure 5-1a shows that the concatenated state $x(t)$ converges to the origin. Figure 5-1b shows that the input mismatch $\mu_\eta(t) = u(t) - u_d(t)$ converges faster than $\mu_d(t)$; hence, the influencing agent is using the desired input which is based on regulating the roaming agent to the desired location. The influencing agent's applied input $u(t) = \mu_\eta(t) + u_d(t)$, shown in Figure 5-1c, remains bounded and converges once the agent's reach the goal location. Since the roaming and influencing agents are modeled using linearly-parameterizable dynamics with an exactly known basis, the parameter estimates can be compared to the true values. Figure 5-1d shows that the system parameter estimates converge to the true values. Figures 5-1e-5-1f show that the critic and actor weight estimates remain bounded; however, because the optimal StaF weights are unknown, the estimates cannot be compared to their ideal values. The positions of the roaming and influencing agents are shown in Figure 5-2. The roaming



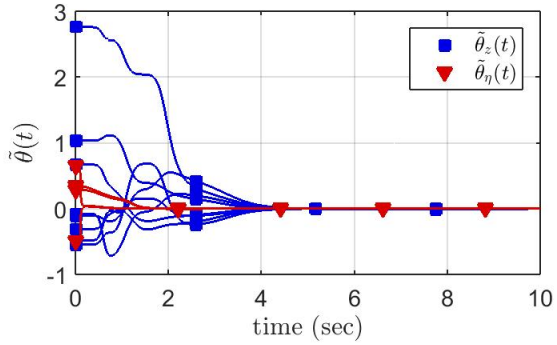
(a) Concatenated state.



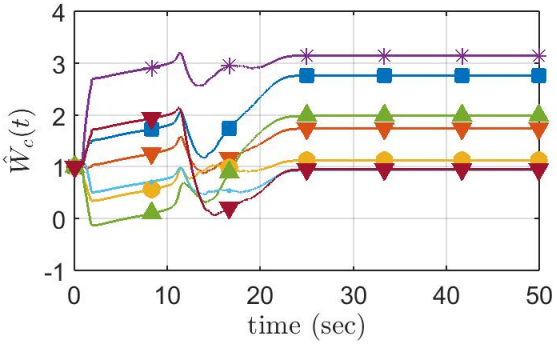
(b) Approximate optimal policy.



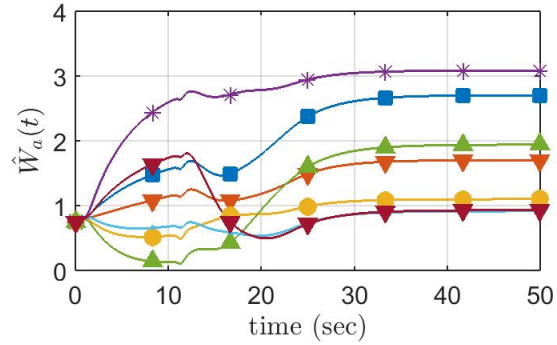
(c) Influencing agent's input.



(d) System ID weight estimation errors.



(e) Critic weight estimates.



(f) Actor weight estimates.

Figure 5-1. The (a) concatenated state $x\left(t\right)$, (b) approximate optimal input $\mu\left(t\right)$, (c) applied influencing agent input $u\left(t\right)$, and (d) system identification errors $\tilde{\theta}\left(t\right)$ all converge to the origin. The (e) critic and actor (f) StaF weight estimates remain bounded.

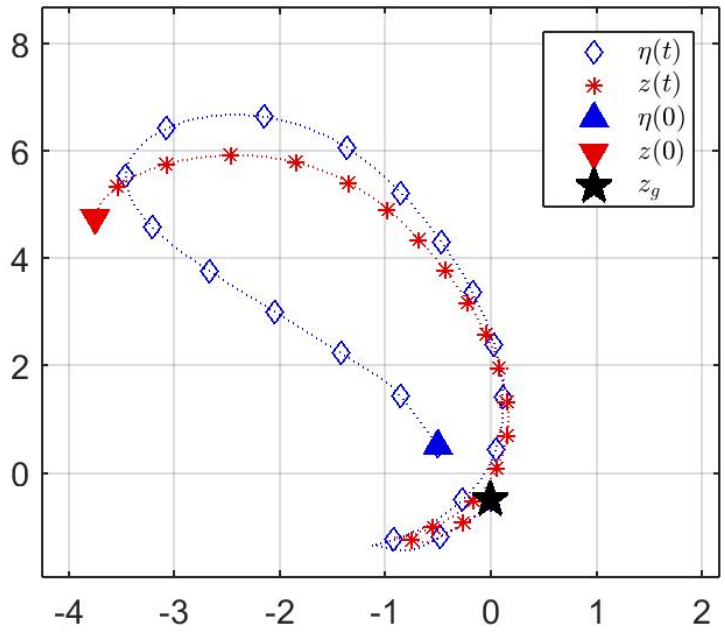


Figure 5-2. Positions of the influencing and roaming agents. The influencing agent (blue diamonds) intercepts and regulates the roaming agent (red stars) to the goal location (black star). The initial influencing agent condition is given by the blue triangle and the initial roaming agent condition is given by the red triangle.

agent is not independently motivated to go to the desired location; hence, in Figure 5-2 the roaming agent initially diverges from the goal location. Once the influencing agent approaches the roaming agent, the roaming agent starts moving away. Motivated to regulate the roaming to the goal location, the influencing agent begins to regulate the roaming agent toward z_g. When the roaming agent first passes by z_g, it tries to escape; however, the influencing agent pushes the roaming agent back in the opposite direction and finally to the goal location of $[0, -0.5]^T$.

5.5 Experiment

Experimental results are also provided to illustrate the performance of the developed approach. A series of ten experiments were conducted, where a different combination of state penalty, Q, and input penalty, R, weights were used to produce different performance characteristics. A Parrot Bebop 2 quadcopter was used as the

Figure 5-3. The unactuated paper platform (left) representing the roaming agent, and the Parrot Bebop 2.0 quadcopter (right) representing the influencing agent.

influencing agent and an unactuated paper platform, shown in Figure 5-3, was used as the roaming agent. The unactuated paper platform was constructed from a paper plate top and bottom fastened to a colored poster board. The turbulent air caused by the quadcopter propellers produce a repulsing force, which causes the nearby roaming agent to slide away. The same experimental platform used in Section 4.6 is leveraged to implement the controller on the quadcopter. A video of a typical run of this experiment is available at [111].

The influencing agent was implemented using dynamics such that $h\left(\eta\left(t\right),z\left(t\right)\right)=0_{2\times1}$; hence the dynamics did not need to be estimated according to Remark 5.5. To identify the interaction dynamics in (5–1) , $p_z = 4$ Gaussian radial basis functions were selected. Each center of the basis was located in a quadrant around the influencing agent, and the standard deviation is selected as $\sqrt{0.5}$ m. Using this representation, the influencing agent estimated the repulsion effects it had on the roaming agent. To approximate the value function, the StaF basis is selected as $\sigma\left(x,c\left(x\right)\right)=\left[\sigma_1\left(x,c_1\left(x\right)\right),\ldots,\sigma_7\left(x,c_7\left(x\right)\right)\right]^T$, where $\sigma_i\left(x\left(t\right),c_i\left(x\left(t\right)\right)\right)=\exp\left(\frac{x^T(t)c(x(t))}{\|x(0)\|^2}\right)$, $c\left(x\left(t\right)\right)=\left(x\left(t\right)+\|x\left(0\right)\|\nu\left(x\left(t\right)\right)d_i\right)$, $\nu\left(x\left(t\right)\right)=\frac{0.05x^T(t)x(t)}{\left(\|x(0)\|^2+x^T(t)x(t)\right)}$, and d_i are the vertices of a 6-simplex. To perform BE extrapolation, ten trajectories $x_i\left(x\left(t\right),t\right)$ are selected at random from a uniform distribution over a $\nu\left(x\left(t\right)\right)\times\nu\left(x\left(t\right)\right)$ square centered

Table 5-2. Initial conditions and parameters selected for the experiments.

Conditions at $t_0 = 0$
$\hat{W}_c(0) = 0.2 \times 1_{5\times1}$, $\hat{W}_a(0) = 0.1 \times 1_{5\times1}$, $\Gamma_c(0) = 0.01 I_5$, $\hat{\theta}(0) = U[-0.1, 0.1] \times 1_{4\times2}$, $\Gamma_\theta(0) = 0.1 I_4$.
Penalizing parameters
$Q = x^T Q_x x$, $P(x) = 0$, $k_d = 1.15$.
Gains and parameters for ADP update laws
$k_{c1} = 0.1$, $k_{c2} = 0.9$, $k_{a1} = 0.9$, $k_{a2} = 0.1$, $\gamma_1 = 0.5$, $\beta_c = 0.001$, $K_a = I_5$, $N = 10$, $\beta_\theta = 0.1$, $M = 50$.

Table 5-3. State and input penalty weights for each experiment.

Experiment	R	Q_x $(\times10^2)$	$\lambda_{avg}\{R\}$	$\lambda_{avg}\{Q\}$ $(\times10^2)$
1	diag $\{20, 10, 50, 25\}$	diag $\{1, 10, 1, 10, 1, 1\}$	26.25	4.0
2	diag $\{20, 5, 50, 25\}$	diag $\{1, 20, 1, 20, 1, 1\}$	25.0	7.33
3	diag $\{20, 5, 50, 25\}$	diag $\{1, 30, 1, 30, 1, 1\}$	25.0	10.67
4	diag $\{25, 10, 55, 30\}$	diag $\{1, 30, 1, 30, 1, 1\}$	30.0	10.67
5	diag $\{30, 20, 65, 40\}$	diag $\{1, 30, 1, 30, 1, 1\}$	38.75	10.67
6	diag $\{40, 30, 75, 50\}$	diag $\{1, 30, 1, 30, 1, 1\}$	48.75	10.67
7	diag $\{40, 30, 80, 55\}$	diag $\{1.5, 30, 1.5, 30, 1, 1\}$	51.25	10.83
8	diag $\{40, 30, 100, 75\}$	diag $\{1.5, 30, 1.5, 30, 1, 1\}$	61.25	10.83
9	diag $\{40, 30, 120, 100\}$	diag $\{1.5, 30, 1.5, 30, 1, 1\}$	72.5	10.83
10	diag $\{60, 40, 150, 125\}$	diag $\{1.5, 30, 1.5, 30, 1, 1\}$	93.75	10.83

at the current state $x(t)$. The goal of the experiment is to indirectly regulate the roaming to a neighborhood of radius $r_{goal} = 0.5$ m of the desired location $z_g = [-2, 0]^T$ m. The selected initial conditions and parameters are provided in Table 5-2.

A survey of ten experiments was performed where different combinations of penalty weights for the state and policy, Q_x and R (shown in Table 5-3) are used, while other parameters remained constant between experiments. Norms of the initial concatenated state and regulation error; the total root-mean square (RMS) values of the norms of the concatenated state x, regulation error e_z, and applied input u; the total cost; and time-to-completion (TTC) are calculated and tabulated in Table 5-4.

Table 5-4. The results for the survey of ten experiments with varying state and input penalty weights.

Experiment	Concatenated State Initial Norm $\|x(0)\|$ (m)	Regulation Error Initial Norm $\|e_z(0)\|$ (m)	Concatenated State Total RMS $\|x\|_{RMS}$ (m)	Regulation Error Total RMS $\|e_z\|_{RMS}$ (m)	Applied Input Total RMS $\|u\|_{RMS}$ $(\frac{m}{sec})$	Total Cost $(\times 10^3)$	TTC (sec)
1	3.973	3.861	3.084	3.043	0.284	20.51	20.60
2	4.026	3.801	3.178	3.102	0.332	26.34	19.07
3	4.403	4.252	3.529	3.476	0.336	26.28	19.31
4	4.363	4.293	3.584	3.561	0.309	30.12	21.35
5	4.455	4.343	3.540	3.499	0.314	28.97	19.19
6	4.367	4.307	3.551	3.531	0.255	39.39	24.87
7	4.285	4.223	3.398	3.368	0.284	35.23	18.34
8	4.372	4.197	3.442	3.379	0.296	49.22	21.39
9	3.973	3.798	3.100	3.027	0.273	37.73	17.95
10	3.930	3.544	2.980	2.844	0.302	66.57	19.49

5.5.1 Discussion

Experiment results are provided in Table 5-4. To display part of the experimental trials, two runs (experiments two and nine), containing vastly different penalty weights and different trajectories, were selected to show the performance of the developed strategy. The concatenated state norm, $\|x(t)\|$, the regulation error norm, $\|e_z(t)\|$, and the phase-space portrait for experiments two and nine are shown in Figures 5-4-5-5, respectively. Figures 5-4a and 5-4b show that the norms of the concatenated state and regulation error for experiment two decrease until the roaming agent is regulated to a neighborhood of the goal location (i.e, $\|e_z(t)\| \leq r_{goal}$). The trajectories of the influencing and roaming agents are shown in Figure 5-4c. Specifically, the influencing agent moves toward the roaming agent to guide it toward the goal location z_g. When the influencing agent approached the roaming agent, the roaming agent moves in the direction of the goal. However, as the roaming agent begins to drift in a wrong direction, the influencing agent adjusts its trajectory to regulate the roaming back in the direction of the goal location.

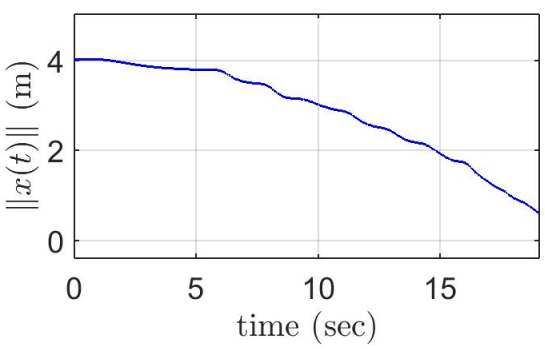
To show the performance of the agents under different state and input penalty weights, Figure 5-5, shows similar metrics as in Figure 5-4. Specifically, the norms of

the concatenated state $x(t)$ and regulation error $e_z(t)$ for experiment nine are displayed in Figures 5-5a and 5-5b; showing that the total state and regulation error decrease for experiment nine as the roaming agent is regulated to a neighborhood of the goal. Figure 5-5c shows the phase-space portrait for both agents. Compared to Figure 5-4c, Figure 5-5c shows that different state and input penalty weights have an effect of how the agents will interact. Moreover, in experiment nine, the influencing agent still achieves the objective of regulating the roaming agent to a neighborhood of the z_g.

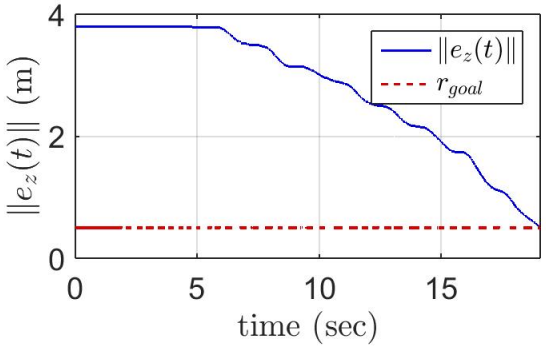
Table 5-3 shows the selected state and input penalty weights for each experiment, while the results are shown in Table 5-4. Specifically, Table 5-4 shows the effect of the system penalty weights and initial setup on the system performance, including: the concatenated state, regulation error, and applied input total RMS values; total cost; and TTC. Moreover, Table 5-4 shows that when the state penalty weights are kept constant, but the input penalty weights are increased, then the total RMS values for the applied input decrease. But, when the state penalty weight is increased, the total RMS values of the concatenated state and regulation error decrease. Moreover, as the penalty weights are increased, the total cost is increased because the influencing agent's actions and the states of both agents are being penalized more. Finally, due to the complex environment, the roaming agent was affected by factors such as varying friction; hence, the TTC was greatly affected between experiments.

5.6 Concluding Remarks

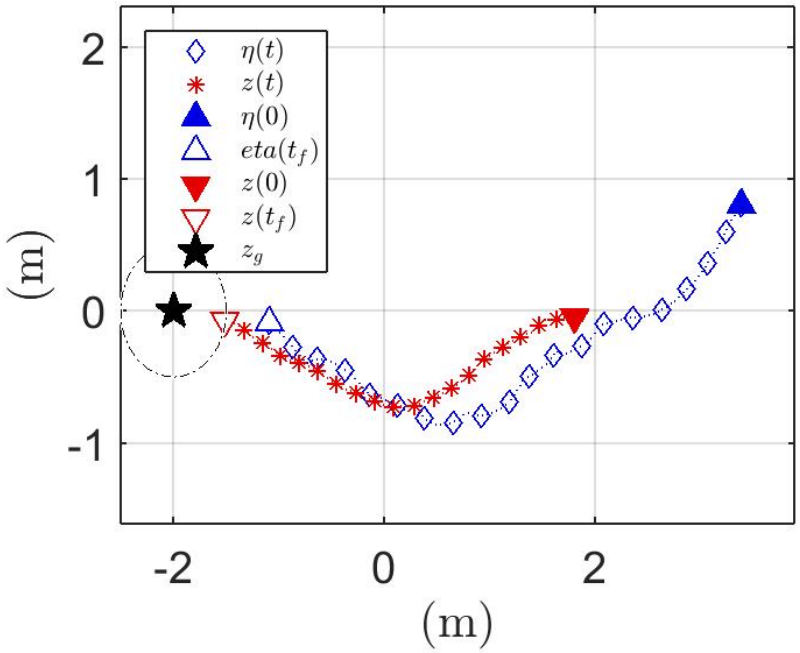
An indirect regulation problem is investigated for a roaming agent being directed by an influencing agent via interaction dynamics. To estimate the uncertainties in the roaming and influencing agent dynamics, a data-based estimator, which relaxes the PE condition, is used. The problem is posed as an infinite-horizon optimal control problem and a local StaF-based ADP method is used to approximate the optimal value function and controller. UUB convergence is shown via a Lyapunov stability analysis for the



(a) Concatenated state norm.

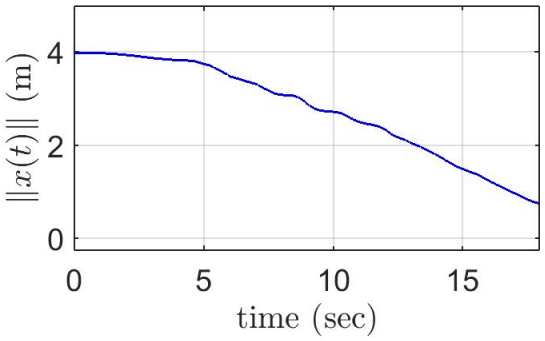


(b) Regulation error norm.

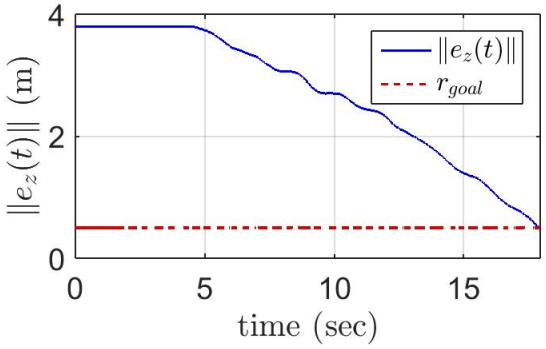


(c) Phase-space portrait.

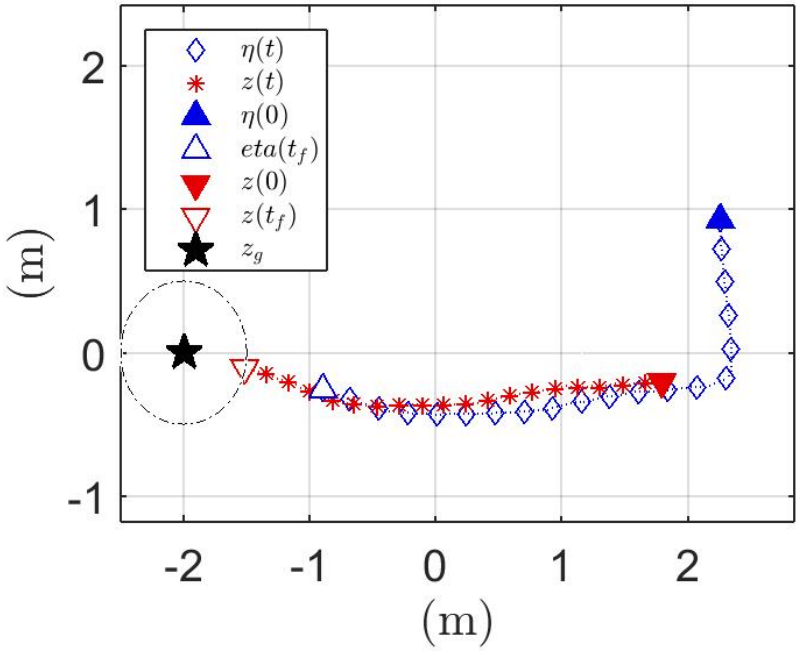
Figure 5-4. The (a) concatenated state norm, $\|x\,(t)\|$, and (b) regulation error norm, $\|e_z\,(t)\|$, converge toward zero. The (c) phase-space portrait shows the trajectories of the roaming and influencing agents, where the influencing agent (blue diamonds) regulates the roaming agent (red stars) to a neighborhood ($r_{goal} = 0.5$ m) of goal location (black star). The initial influencing agent condition is given by the blue triangle and the initial roaming agent condition is given by the red triangle.



(a) Concatenated state norm.



(b) Regulation error norm.



(c) Phase-space portrait.

Figure 5-5. The (a) concatenated state norm, $\|x\,(t)\|$, (b) regulation error norm, $\|e_z\,(t)\|$, and (c) phase-space portrait for experiment nine. In Figure 5-5c, the influencing agent (blue diamonds) regulates the roaming agent (red stars) to a neighborhood ($r_{goal} = 0.5$ m) of goal location (black star). The initial influencing agent condition is given by the blue triangle and the initial roaming agent condition is given by the red triangle.

closed-loop error system. Simulation results in addition to experimental results for two-state influencing and roaming agents are included, which illustrate the performance of the developed method. Motivated by simplifying the strategy and possibly consider worst case client agent dynamics, the next chapter formulates the problem as a differential game.

CHAPTER 6
APPROXIMATE OPTIMAL INFLUENCE OVER AN AGENT: A GAME-BASED APPROACH

In this chapter, an approximately optimal regulation problem consisting of a single influencing agent and client agent is considered. The goal of the influencing agent is to intercept and then regulate the roaming agent to a desired location unknown to the roaming agent. Two error systems are designed such that the influencing agent first tracks and then steers the client agent to a desired location. The problem is formulated as a differential game where the client agent's worst case disturbing policy and the influencing agents optimal policy are estimated via ADP. The computationally efficient StaF kernel method is used to generate forward-in-time approximations of the policies, and a Lyapunov-based stability is included to show stability of the overall system. Simulation results demonstrate the performance of the developed method.

6.1 Problem Formulation

Consider a roaming agent governed by the drift dynamics

$$\dot{z}(t) = f(z(t), \eta(t), t), \tag{6–1}$$

where $z : \mathbb{R}_{t\geq t_0} \to \mathbb{R}^n$ is the roaming agent state, $\eta : \mathbb{R}_{t\geq t_0} \to \mathbb{R}^n$ denotes the influencing agent state, $t_0 \in \mathbb{R}_{\geq 0}$ is the initial time, and $f : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}_{t\geq t_0} \to \mathbb{R}^n$ denotes the unknown drift dynamics which are locally Lipschitz in the states and continuous in t. The roaming agent dynamics in (6–1) are assumed to satisfy Assumption 5.2. The objective is for the influencing agent to regulate the roaming agent to some goal location, given by $z_g \in \mathbb{R}^n$, which is unknown to the roaming agent. Since (6–1) is not directly controlled, a controlled pursuing agent governed by the dynamics in (5–2) with control input of dimension $m_\eta = m$ is used to influence the roaming agent through an indirect interaction.

To facilitate the objective, two errors, $e_1, e_2 : \mathbb{R}_{\geq t_0} \to \mathbb{R}^n$, are defined as

$$e_1(t) \triangleq z(t) - \eta(t), \tag{6–2}$$

$$e_2(t) \triangleq (\eta(t) - z_g) - k_1 (z(t) - z_g), \tag{6–3}$$

where $k_1 \in \mathbb{R}_{>1}$ is a gain. The error in (6–2) is used to quantify the influencing agent's ability to intercept the roaming agent, while (6–3) quantifies its ability to steer the target to the goal location. Let $x(t) \triangleq \left[e_1^T(t), e_2^T(t)\right]^T \in \mathbb{R}^{2n}$ denote the concatenated state vector, and define the mappings $s_1, s_2 : \mathbb{R}^{2n} \to \mathbb{R}^n$ as $s_1(x(t)) \triangleq \frac{1}{(1-k_1)}(e_1(t) + e_2(t)) + z_g$, and $s_2(x(t)) \triangleq \frac{1}{(1-k_1)}(k_1 e_1(t) + e_2(t)) + z_g$. Using (6–2)-(6–3), and the mappings s_1 and s_2, the roaming and influencing agent states are represented in terms of the errors as $z(t) = s_1(x(t))$ and $\eta(t) = s_2(x(t))$, respectively. The roaming agent dynamics in (6–1) are not directly controlled and are non-autonomous. Thus, to facilitate the following development, (6–1) can be rewritten as

$$\dot{z}(t) = \Delta f(x(t)) + d(t), \tag{6–4}$$

where $d : \mathbb{R}_{\geq t_0} \to \mathbb{R}^n$ represents the roaming agent's unknown disturbing input, and $\Delta f : \mathbb{R}^{2n} \to \mathbb{R}^n$ denotes the unknown roaming agent's residual interaction drift dynamics.

Remark 6.1. For simplicity and to keep in line with previous literature, $\Delta f(x(t))$ can be absorbed into $d(t)$ where (6–4) can be represented $\dot{z}(t) = \bar{d}(t)$. Hence, these can be defined dynamics as the worst-case dynamics for the roaming agent.

Taking the time-derivative of (6–2) and (6–3), substituting in (5–2) and (6–4), and using the definitions for s_1, and s_2 yields

$$\dot{x}(t) = F(x(t)) + G(x(t))u(t) + Kd(t), \tag{6–5}$$

where $F : \mathbb{R}^{2n} \to \mathbb{R}^{2n}$, $G : \mathbb{R}^{2n} \to \mathbb{R}^{2n\times m}$, $K \in \mathbb{R}^{2n\times n}$ are defined as

$$F\left(x\left(t\right)\right) \triangleq \begin{bmatrix} \Delta f\left(x\left(t\right)\right) - h\left(s_1\left(x\left(t\right)\right), s_2\left(x\left(t\right)\right)\right) \\ h\left(s_1\left(x\left(t\right)\right), s_2\left(x\left(t\right)\right)\right) - k_1 \Delta f\left(x\left(t\right)\right) \end{bmatrix},$$

$G\left(x\left(t\right)\right) \triangleq \left[-g\left(s_2\left(x\left(t\right)\right)\right)^T, g\left(s_2\left(x\left(t\right)\right)\right)^T\right]^T$, and $K \triangleq \left[I_n,\ -k_1 I_n\right]^T$, respectively.

6.1.1 Optimal Control Development

Given the error system in (6–5), the goal is to find $u\left(t\right)$ and $d\left(t\right)$ that optimize the cost functional

$$J\left(x, u, d\right) \triangleq \int_{t_0}^{\infty} r\left(x\left(\tau\right), u\left(\tau\right), d\left(t\right)\right) d\tau, \tag{6–6}$$

subject to the dynamics in (6–5). In (6–6), $r : \mathbb{R}^{2n} \times \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ is the instantaneous cost defined as $r\left(x, u, d\right) \triangleq Q\left(x\right) + u^T R u - \gamma^2 d^T d$, where $Q : \mathbb{R}^{2n} \to \mathbb{R}_{\geq 0}$ is a user-defined PD function which satisfies $\underline{q}\left\|x\right\|^2 \leq Q\left(x\right) \leq \overline{q}\left\|x\right\|^2$ for all $x \in \mathbb{R}^{2n}$, where $\underline{q}, \overline{q} \in \mathbb{R}_{>0}$, $R \in \mathbb{R}^{m\times m}$ is a PD symmetric weight matrix, and $\gamma \in \mathbb{R}_{>0}$ is a constant gain.

The optimal value function, denoted by $V^* : \mathbb{R}^{2n} \to \mathbb{R}_{\geq 0}$, is expressed as

$$V^*\left(x\left(t\right)\right) \triangleq \min_{u(\tau)} \max_{d(\tau)} \int_t^{\infty} r\left(x\left(\tau\right), u\left(\tau\right), d\left(\tau\right)\right) d\tau,$$

for $\tau \in \mathbb{R}_{\geq t}$, and is characterized by the HJI equation given by

$$\begin{aligned} 0 = {} & r\left(x\left(t\right), u^*\left(x\left(t\right)\right), d^*\left(x\left(t\right)\right)\right) + \nabla V^*\left(x\left(t\right)\right) F\left(x\left(t\right)\right) \\ & + \nabla V^*\left(x\left(t\right)\right)\left(G\left(x\left(t\right)\right) u^*\left(x\left(t\right)\right) + K d^*\left(x\left(t\right)\right)\right), \end{aligned} \tag{6–7}$$

for all $t \in \mathbb{R}_{\geq t_0}$ with $V^*\left(0\right) = 0$ [112–115]. Given a solution $V^*\left(x\right) \geq 0$ to (6–7), the associated optimal influencing agent input policy $u^* : \mathbb{R}^{2n} \to \mathbb{R}^m$ and worst-case roaming agent disturbing policy $d^* : \mathbb{R}^{2n} \to \mathbb{R}^n$ are determined from (6–7) as

$$u^*\left(x\right) = -\frac{1}{2} R^{-1} G\left(x\right)^T \left(\nabla V^*\left(x\right)\right)^T, \tag{6–8}$$

and

$$d^{*}\left(x\right)=\frac{1}{2\gamma^{2}}K^{T}\left(\nabla V^{*}\left(x\right)\right)^{T}, \tag{6–9}$$

respectively.

6.2 Approximate Optimal Control

6.2.1 System Identification

The HJI in (6–7) requires knowledge of the roaming and influencing agent drift dynamics. Using the universal function approximation property [97], on any compact set $\chi \subset \mathbb{R}^{2n}$, the functions $k_1\Delta f\left(x\left(t\right)\right)$ and $h\left(s_1\left(x\left(t\right)\right), s_2\left(x\left(t\right)\right)\right)$ in (6–5) can be represented as $k_1\Delta f\left(x\left(t\right)\right)=\theta_z^T S_z\left(x\left(t\right)\right)+\varepsilon_z\left(x\left(t\right)\right)$, and $h\left(s_1\left(x\left(t\right)\right), s_2\left(x\left(t\right)\right)\right)=\theta_\eta^T S_\eta\left(x\left(t\right)\right)+\varepsilon_\eta\left(x\left(t\right)\right)$, respectively, where $\theta_j \in \mathbb{R}^{p_j\times n}$ is the unknown weight matrix, $S_j : \mathbb{R}^{2n} \to \mathbb{R}^{p_j}$ is the known basis function, and $\varepsilon_j : \mathbb{R}^{2n} \to \mathbb{R}^{n}$ is the function approximation error where $j=\{z,\eta\}$ is the index referring to which dynamics each quantity belongs to.[1]

To further facilitate the development, the function $F\left(x\left(t\right)\right)$ in (6–5) can be represented as

$$F\left(x\left(t\right)\right)=L\Lambda\left(\theta^T\right)S\left(x\left(t\right)\right)+L\varepsilon\left(x\left(t\right)\right),$$

where $L \triangleq \begin{bmatrix} \frac{1}{k_1}I_n, & -I_n \\ -I_n & I_n \end{bmatrix}$, $\theta \triangleq \left[\theta_z^T, \theta_\eta^T\right]^T \in \mathbb{R}^{(p_z+p_\eta)\times n}$ denotes the combined unknown weights, $S\left(x\left(t\right)\right) \triangleq \left[S_z^T\left(x\left(t\right)\right), S_\eta^T\left(x\left(t\right)\right)\right]^T \in \mathbb{R}^{p_z+p_\eta}$ the combined known basis, $\varepsilon\left(x\left(t\right)\right) \triangleq \left[\varepsilon_z^T\left(x\left(t\right)\right), \varepsilon_\eta^T\left(x\left(t\right)\right)\right]^T \in \mathbb{R}^{2n}$ the combined function approximation errors, and the mapping $\Lambda : \mathbb{R}^{(p_z+p_z)\times n} \to \mathbb{R}^{2n\times(p_z+p_\eta)}$ is defined as $\Lambda\left(\theta^T\right) \triangleq \begin{bmatrix} \theta_z^T & 0_{n\times p_\eta} \\ 0_{n\times p_z} & \theta_\eta^T \end{bmatrix}$. The NN weights θ, basis $S\left(x\right)$, and function approximation error $\varepsilon\left(x\right)$ are assumed to satisfy Assumption 5.3.

[1] If $\Delta f\left(x\left(t\right)\right)$ is absorbed into $d\left(t\right)$ according to Remark 6.1, then $\theta_z^T S_z\left(x\left(t\right)\right)+\varepsilon_z\left(x\left(t\right)\right)=0_{n\times 1}$ and there is no need to estimate θ_z.

Using an estimate $\hat{\theta} \in \mathbb{R}^{(p_\eta+p_z)\times n}$, the drift dynamics are approximated as $\hat{F}_\theta\left(x(t), \hat{\theta}\right) = L\Lambda\left(\hat{\theta}^T(t)\right) S(x(t))$. To facilitate the use of adaptive methods in this section to identify the uncertainties in $F(x(t))$, the following assumptions are required.

Assumption 6.1. There is exist a constant $\overline{d} \in \mathbb{R}_{>0}$ such that the roaming agent's unknown disturbing input $d(t)$ in (6–5) is bound as $\|d(t)\| \leq \overline{d}$.[2]

Assumption 6.2. [1, 18]. A known a priori compact set $\Theta \subset \mathbb{R}^{p_\eta+p_z}$ contains the unknown weight matrix θ. The weight estimates $\hat{\theta} : \mathbb{R}_{\geq t_0} \to \mathbb{R}^{(p_\eta+p_z)\times n}$ are updated based on a switched update law

$$\dot{\hat{\theta}}(t) = \mathcal{F}_s\left(\hat{\theta}(t), t\right), \tag{6–10}$$

with $\hat{\theta}(t_0) \subset \Theta$. In (6–10), $\left\{\mathcal{F}_s : \mathbb{R}^{(p_z+p_\eta)\times n} \times \mathbb{R}_{\geq t_0} \to \mathbb{R}^{(p_z+p_\eta)\times n}\right\}_{s\in\mathbb{Z}_{>0}}$ denotes a family of continuously differentiable functions and $s \in \mathbb{Z}_{>0}$ denotes the switching index, which ensure that $\left\|\theta - \hat{\theta}(t)\right\| \leq B_\theta \in \mathbb{R}_{\geq 0}$.[3] There exists a function $V_\theta : \mathbb{R}^{(p_\eta+p_z)n} \times \mathbb{R}_{\geq t_0} \to \mathbb{R}_{\geq 0}$ such that

$$\underline{v}_\theta\left(\left\|\tilde{\theta}\right\|^2\right) \leq V_\theta\left(\tilde{\theta}, t\right) \leq \overline{v}_\theta\left(\left\|\tilde{\theta}\right\|^2\right), \tag{6–11}$$

$$\frac{\partial V_\theta\left(\tilde{\theta}, t\right)}{\partial \tilde{\theta}}\left(-\mathcal{F}_s\left(\tilde{\theta}, t\right)\right) + \frac{\partial V_\theta\left(\tilde{\theta}, t\right)}{\partial t} \leq -K_\theta\left\|\tilde{\theta}\right\|^2 + D_\theta\left\|\tilde{\theta}\right\|, \tag{6–12}$$

for all $t \in \mathbb{R}_{\geq t_0}$, $s \in \mathbb{N}$, and $\tilde{\theta}(t) \in \mathbb{R}^{(p_z+p_\eta)\times n}$, where $\tilde{\theta}(t) \triangleq \theta - \hat{\theta}(t)$, $\underline{v}_\theta, \overline{v}_\theta : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ are class $\mathcal{K}$ functions, $K_\theta \in \mathbb{R}_{>0}$ is an adjustable parameter, and $D_\theta \in \mathbb{R}_{\geq 0}$ is a constant, and the ratio $\frac{D_\theta}{K_\theta}$ is sufficiently small.

Remark 6.2. Various methods can be used to satisfy Assumption 6.2 (cf., [58, 81, 84, 99, 100, 117, 118]). If CL approaches, such as those in [58, 109, 117, 118], are used, and $\Delta f(f(x))$ is also being estimated, then the constant D_θ will depend on

[2] Assumption 6.1 is included to facilitate the use of Assumption 6.2.

[3] A projection algorithm can be used to ensure the estimates remain bounded, see Remark 3.6 or Section 4.4 in [116] for details of the projection operator.

the number of basis used for function approximation. Hence, as data is gathered, purging techniques such as [81] can be used to reduce D_θ. Moreover, for methods such as [58, 109, 117, 118] the stability analysis is divided into disjoint time-intervals: one before learning is achieved and one afterwards. A common Lyapunov function is then used to prove exponential convergence $t \in \mathbb{R}_{\geq t_0}$. Appendix B.1 provides a supplementary proof of a data-based update law which satisfies Assumption 6.2.

6.2.2 Value Function Approximation

Similar to Chapters 4 and 5, the unknown value function $V^*(x)$ is approximated via computationally efficient StaF kernels centered around $x \in \chi \subset \mathbb{R}^{2n}$ [1, 96]. The value function, optimal influencing agent policy, and worst-case roaming agent disturbing policy are

$$V^*(y) = W(x)^T \sigma(y, c(x)) + \epsilon_v(x, y), \tag{6–13}$$

$$u^*(y) = -\frac{R^{-1}G^T(y)}{2}\left(\nabla\sigma(y, c(x))^T W(x) + \epsilon_W(x, y)^T\right), \tag{6–14}$$

and

$$d^*(y) = \frac{K^T}{2\gamma^2}\left(\nabla\sigma(y, c(x))^T W(x) + \epsilon_W^T(x, y)\right), \tag{6–15}$$

respectively, where $W : \chi \to \mathbb{R}^P$, $\sigma : \chi \times \chi \to \mathbb{R}^P$, and $\epsilon_v : \chi \times \chi \to \mathbb{R}$ are defined in Chapters 4 and 5, and $\epsilon_W(x, y) \triangleq \sigma(y, c(x))^T \nabla W(x) + \nabla\epsilon_v(x, y)$. Moreover, the approximate value function $\hat{V} : \mathbb{R}^{3n} \times \mathbb{R}^{3n} \times \mathbb{R}^P \to \mathbb{R}$ is expressed as $\hat{V}\left(y, x, \hat{W}_c\right) = \hat{W}_c^T\sigma(y, c(x))$, and approximate influencing and roaming agent policies $\hat{u} : \mathbb{R}^{2n} \times \mathbb{R}^{2n} \times \mathbb{R}^P \to \mathbb{R}^m$ and $\hat{d} : \mathbb{R}^{2n} \times \mathbb{R}^{2n} \times \mathbb{R}^P \to \mathbb{R}^n$ are expressed as

$$\hat{u}\left(y, x, \hat{W}_a\right) = -\frac{R^{-1}G^T(y)}{2}\nabla\sigma(y, c(x))^T\hat{W}_a, \tag{6–16}$$

$$\hat{d}\left(y, x, \hat{W}_d\right) = \frac{K^T}{2\gamma^2}\nabla\sigma(y, c(x))^T\hat{W}_d, \tag{6–17}$$

where $\hat{W}_c, \hat{W}_a, \hat{W}_d \in \mathbb{R}^P$ denote the critic, actor, and disturbance NN weight estimates, respectively. When $\hat{V}$, (6–16), (6–17), along with the estimates $\hat{\theta}$ are substituted into (6–7), the BE $\delta : \mathbb{R}^{2n} \times \mathbb{R}^{2n} \times \mathbb{R}^P \times \mathbb{R}^P \times \mathbb{R}^P \times \mathbb{R}^{(p_z+p_\eta)\times n} \to \mathbb{R}$ is

$$\delta\left(y,x,\hat{\theta},\hat{W}_c,\hat{W}_a,\hat{W}_d\right)=\nabla\hat{V}\left(y,x,\hat{W}_c\right)\left(F_\theta\left(y,\hat{\theta}\right)+G\left(y\right)\hat{u}\left(y,x,\hat{W}_a\right)+K\hat{d}\left(y,x,\hat{W}_d\right)\right)$$
$$+r\left(y,\hat{u}\left(y,x,\hat{W}_a\right),\hat{d}\left(y,x,\hat{W}_d\right)\right), \tag{6–18}$$

where $F_\theta\left(y,\hat{\theta}\right)=L\Lambda\left(\hat{\theta}^T\right)S\left(y\right)$.

6.2.3 Online Learning

Using the approach in Chapter 5, the BE in (6–18) is evaluated at the values $y=x\left(t\right)$ (i.e. the current state), such that $\delta_t\left(t\right)\triangleq\delta\left(x\left(t\right),x\left(t\right),\hat{\theta}\left(t\right),\hat{W}_c\left(t\right),\hat{W}_a\left(t\right),\hat{W}_d\left(t\right)\right)$. Likewise, the input which the roaming agent utilizes as $u\left(t\right)=\hat{u}\left(x\left(t\right),x\left(t\right),\hat{W}_a\left(t\right)\right)$, while the instantaneous approximate worst-case roaming agent disturbing policy input is given as $\hat{d}\left(t\right)=\hat{d}\left(x\left(t\right),x\left(t\right),\hat{W}_d\left(t\right)\right)$. In addition, sets of functions $\{x_i:\mathbb{R}^{2n}\times\mathbb{R}_{\geq t_0}\to\mathbb{R}^{2n}\}_{i=1}^N$ are selected by the critic such that $x_i\left(x\left(t\right),t\right)\in\overline{B_r\left(x\left(t\right)\right)}$, and the BE and policies are evaluated at $y=x_i\left(x\left(t\right),t\right)$ as $\delta_{ti}\left(t\right)\triangleq\delta\left(x_i\left(x\left(t\right),t\right),x\left(t\right),\hat{\theta}\left(t\right),\hat{W}_c\left(t\right),\hat{W}_a\left(t\right),\hat{W}_d\left(t\right)\right)$, $\hat{u}_i\left(t\right)=\hat{u}\left(x_i\left(x\left(t\right),t\right),x\left(t\right),\hat{W}_a\left(t\right)\right)$, and $\hat{d}_i\left(t\right)=\hat{d}\left(x_i\left(x\left(t\right),t\right),x\left(t\right),\hat{W}_d\left(t\right)\right)$, respectively.

The critic aims to find a set of weights that minimize the BE and is updated according to the modified update laws

$$\dot{\hat{W}}_c\left(t\right)=-\Gamma_c\left(t\right)\frac{k_{c1}}{N+1}\frac{\omega\left(t\right)}{\rho\left(t\right)}\delta_t\left(t\right)-\Gamma_c\left(t\right)\frac{k_{c2}}{N+1}\sum_{i=1}^{N}\frac{\omega_i\left(t\right)}{\rho_i\left(t\right)}\delta_{ti}\left(t\right), \tag{6–19}$$

$$\dot{\Gamma}_c\left(t\right)=\beta_c\Gamma_c\left(t\right)-\Gamma_c\left(t\right)\frac{k_{c1}}{N+1}\frac{\omega\left(t\right)\omega^T\left(t\right)}{\rho^2\left(t\right)}\Gamma_c\left(t\right)-\Gamma_c\left(t\right)\frac{k_{c2}}{N+1}\sum_{i=1}^{N}\frac{\omega_i\left(t\right)\omega_i^T\left(t\right)}{\rho_i^2\left(t\right)}\Gamma_c\left(t\right), \tag{6–20}$$

with $\lambda_{\min}\left\{\Gamma_c\left(t_0\right)\right\}>0$, where $\rho\left(t\right)\triangleq\sqrt{1+\gamma_1\omega^T\left(t\right)\omega\left(t\right)}$, $\rho_i\left(t\right)\triangleq\sqrt{1+\gamma_1\omega_i^T\left(t\right)\omega_i\left(t\right)}$, and $k_{c1},k_{c2},\gamma_1,\beta_c\in\mathbb{R}_{>0}$ are learning gains,

$$\omega(t) = \nabla\sigma(x(t), c(x(t)))\Big(F_\theta\big(x(t), \hat{\theta}(t)\big) + G(x(t))\,\hat{u}\big(x(t), x(t), \hat{W}_a(t)\big) + K\hat{d}\big(x(t), x(t), \hat{W}_d(t)\big)\Big),$$

and $\omega_i(t) = \nabla\sigma_i\left(F_{\theta i} + G_i\hat{u}_i + K_i\hat{d}_i\right)$ with $K_i = K$ since K is constant.[4] [5] Similar to Chapter 5, the states x and x_i in this chapter are assumed to satisfy Assumption 4.5 where ω_k and N are substituted with ω_i and $N+1$, respectively (i.e., $\omega_k = \omega_i$ and $N = N+1$). Using Assumption 4.5, along with the initial condition of $\Gamma_c(t)$, it can shown that $\underline{\Gamma}_c I_P \leq \Gamma_c(t) \leq \overline{\Gamma}_c I_P$, where $\underline{\Gamma}_c, \overline{\Gamma}_c \in \mathbb{R}_{>0}$ are constant bounds [1]. To provide weight estimates for (6–16) and (6–17), the actor and disturbance weight estimates are updated to follow the critic weight estimate as

$$\dot{\hat{W}}_a(t) = \text{proj}\left\{-K_a k_{a1}\left(\hat{W}_a(t) - \hat{W}_c(t)\right)\right\}, \tag{6–21}$$

$$\dot{\hat{W}}_d(t) = \text{proj}\left\{-K_d k_{d1}\left(\hat{W}_d(t) - \hat{W}_c(t)\right)\right\}, \tag{6–22}$$

where $k_{a1}, k_{d1} \in \mathbb{R}_{\geq 0}$ are learning gains, and $K_a, K_d \in \mathbb{R}^{P\times P}$ are positive-definite symmetric matrices.

6.3 Stability Analysis

To facilitate the following stability analysis, define a closed ball of radius ζ centered at the origin as $B_\zeta \subset \mathbb{R}^{2n+(p_\eta+p_z)n+2P}$. To further aid in the subsequent stability analysis, let $Z_L \triangleq \left[x^T, \tilde{W}_c^T, \tilde{W}_a^T, \tilde{W}_d^T, \text{vec}\left(\tilde{\theta}\right)^T\right]^T$, where $\tilde{(\cdot)}$ is defined in Section 2.1. Furthermore, using (6–18) along with (6–7) and (6–14)-(6–17), the BE $\delta_t(t)$ and extrapolated

[4] Similar to Chapter 5, the notation ϕ_i denotes $\phi_i \triangleq \phi(x_i(t), \cdots)$ for an arbitrary function ϕ (see Section 5.2.3).

[5] Compared to previous chapters, the update laws in (6–19) and (6–20) are normalized by $N+1$, which results in a smaller residual bound in the stability analysis.

BEs $\delta_{ti}(t)$ are expressed analytically as $\delta_t = -\omega^T \tilde{W}_c + \frac{1}{4}\tilde{W}_a^T G_\sigma \tilde{W}_a - \frac{1}{4}\tilde{W}_d^T K_\sigma \tilde{W}_d - W^T \nabla\sigma \tilde{F} + \Delta$, and $\delta_{ti} = -\omega_i^T \tilde{W}_c + \frac{1}{4}\tilde{W}_a^T G_{\sigma i} \tilde{W}_a - \frac{1}{4}\tilde{W}_d^T K_{\sigma i} \tilde{W}_d - W^T \nabla\sigma_i \tilde{F}_i + \Delta_i$, respectively, where $\tilde{F} \triangleq F_\theta\left(x, \tilde{\theta}\right)$, $\tilde{F}_i \triangleq F_\theta\left(x_i, \tilde{\theta}\right)$, and the functions Δ, Δ_i are uniformly bounded over χ such that the $\overline{\|\Delta\|}$ and $\overline{\|\Delta_i\|}$ decrease with decreasing $\overline{\|\nabla W\|}$, $\overline{\|\nabla \epsilon_v\|}$, $\overline{\|\nabla \epsilon_{v,i}\|}$, $\overline{\|\varepsilon\|}$, and $\overline{\|\varepsilon_i\|}$. [6]

Assumption 6.3. [119]. The gradient $\nabla V^*(x)$ is bounded such that $\|\nabla V^*(x)\| \leq \alpha \|x\|$, for all $x \in \mathbb{R}^{2n}$ where $\alpha \in \mathbb{R}_{>0}$ is a constant.

Remark 6.3. Compared to results such as [20, 120, 121] which assume $\nabla V^*(x)$ is bounded by a constant such as $\|\nabla V^*(x)\| \leq \alpha$, Assumption 6.3 is a less restrictive in that it does not limit the growth of $\|\nabla V^*(x)\|$. Remark A.1 in Appendix A.4 discusses how the sufficient conditions in (6–23) and the constant ι change when the $\|\nabla V^*(x)\| \leq \alpha$ is assumed. Specifically, the constant ι is larger because the constant $\iota_x \in \mathbb{R}_{>0}$ defined in the appendix can not be reduced by the selection of Q with a large $\underline{q}$.

Theorem 6.1. *Provided Assumptions 4.5, 5.2, 5.3, and 6.1-6.3 are satisfied and the following sufficient conditions hold*

$$\underline{q} \geq \frac{\alpha^2 \overline{\|K_\gamma - G_R\|}}{2}, \qquad \underline{c} \geq \frac{k_{a1} + k_{d1} + \varphi_{c\theta}}{k_{c2}}, \; K_\theta \geq \varphi_{c\theta}, \tag{6–23}$$

$$\sqrt{\frac{\iota}{\kappa}} \leq \underline{v}_l^{-1}\left(\overline{v}_l(\zeta)\right), \tag{6–24}$$

with $\underline{c}, \varphi_{c\theta}, \kappa, \iota \in \mathbb{R}_{>0}$ *defined in Appendix A.4, then the policies in* (6–16) *and* (6–17) *along with the update laws in* (6–19)-(6–22) *for the system in* (6–5) *ensure that the state* Z_L, *and approximate policies* $u(t)$, $\hat{d}(t)$ *remain UUB.*

[6] The functions K_σ and K_γ are defined as $K_\sigma \triangleq \nabla\sigma K_\gamma \nabla\sigma^T$ and $K_\gamma \triangleq \frac{1}{\gamma^2} K K^T$, respectively, while G_σ and G_R are defined in Chapter 5.

Proof. Consider the Lyapunov function candidate $V_L : \mathbb{R}^{2n+2P+(p_\eta+p_z)n} \times \mathbb{R}_{\geq t_0} \to \mathbb{R}_{\geq 0}$ defined as

$$V_L(Z_L,t) \triangleq V^*(x) + \frac{1}{2}\tilde{W}_c^T\Gamma_c^{-1}(t)\tilde{W}_c + \frac{1}{2}\tilde{W}_a^T K_a^{-1}\tilde{W}_a + \frac{1}{2}\tilde{W}_d^T K_d^{-1}\tilde{W}_d + V_\theta\left(\tilde{\theta},t\right), \quad (6\text{–}25)$$

which can be bounded by class $\mathcal{K}$ functions $\underline{v}_l, \overline{v}_l : \mathbb{R} \to \mathbb{R}_{\geq 0}$ as

$$\underline{v}_l(\|Z_L\|) \leq V_L(Z_L,t) \leq \overline{v}_l(\|Z_L\|) \quad (6\text{–}26)$$

for all $Z_L \in \mathbb{R}^{2n+2P+n(p_z+p_\eta)}$ and $t \in \mathbb{R}_{\geq t_0}$. Taking the time-derivative of (6–25) along the system trajectory results in

$$\dot{V}_L(Z_L,t) = \nabla V^*\left(F + G\hat{u} + K\hat{d}\right) + \dot{V}_\theta\left(\tilde{\theta},t\right) + \tilde{W}_c^T\Gamma_c^{-1}\left(\dot{W} - \dot{\hat{W}}_c\right)$$
$$- \frac{1}{2}\tilde{W}_c^T\left(\Gamma_c^{-1}\dot{\Gamma}_c\Gamma_c^{-1}\right)\tilde{W}_c + \tilde{W}_a^T K_a^{-1}\left(\dot{W} - \dot{\hat{W}}_a\right) + \tilde{W}_d^T K_d^{-1}\left(\dot{W} - \dot{\hat{W}}_d\right).$$

Substituting (6–7), (6–8)-(6–9), and (6–16)-(6–22), then using Assumptions 4.5, 5.3, 6.2, 6.3, and along with the fact that $\overline{\left\|\tilde{F}_\theta\right\|} \leq \lambda_{\max}\{L\}\,\overline{S}\left\|\tilde{\theta}\right\|$, and $\overline{\left\|\frac{\omega}{\rho^2}\right\|} \leq \frac{1}{2\sqrt{\gamma_c}}$, and finally applying the sufficient gain conditions in (6–23) yields

$$\dot{V}_L(Z_L,t) \leq -\kappa\|Z_L\|^2 - \left(\kappa\|Z_L\|^2 - \iota\right),$$

and $\dot{V}_L(Z_L,t) \leq -\kappa\|Z_L\|^2, \quad \forall \|Z_L\| \geq \sqrt{\frac{\iota}{\kappa}}$ follows. Then, using (6–24), (6–26), [104, Theorem 4.18] is invoked to conclude that Z_L is uniformly ultimately bounded such that $\limsup_{t\to\infty}\|Z_L(t)\| \leq \underline{v}_l^{-1}\left(\overline{v}_l\left(\sqrt{\frac{\iota}{\kappa}}\right)\right)$. Since $Z_L \in \mathcal{L}_\infty$, then $\tilde{W}_c, \tilde{W}_a, \tilde{W}_d, \tilde{\theta},$ and $x \in \mathcal{L}_\infty$ and therefore $u, \hat{d} \in \mathcal{L}_\infty$. Since $x \in \mathcal{L}_\infty$, $W(x) \in \mathcal{L}_\infty$, and from (6–2) and (6–3) $e_1, e_2 \in \mathcal{L}_\infty$; hence $(z - z_g) \in \mathcal{L}_\infty$ follows. □

Remark 6.4. The sufficient conditions in (6–23) can be satisfied by selecting a function Q with large $\underline{q}$, selecting the gains K_d, K_a, R with large minimum eigenvalues, and by increasing the gains k_1, k_{c2}, and γ. In addition, selecting extrapolation points $x_i(x(t),t)$ so that $\underline{c}$ is large will also aid in (6–23) being satisfied.

6.4 Simulation

6.4.1 Unknown Roaming Agent Basis

To demonstrate the developed method, a simulation is performed for the roaming agent in (6–1) with $z : \mathbb{R}_{\geq 0} \to \mathbb{R}^2$, where

$$f\left(s_1\left(x\left(t\right)\right), s_2\left(x\left(t\right)\right)\right) = A\Phi_1\left(x\left(t\right)\right) + B\Phi_2\left(x\left(t\right)\right), \tag{6–27}$$

$\Phi_1\left(x\left(t\right)\right) \triangleq e_1\left(t\right)\exp\left(-\frac{\|e_1(t)\|^2}{2}\right)$, $\Phi_2\left(x\left(t\right)\right) \triangleq e_g\left(t\right)\exp\left(-\frac{\|e_g(t)\|^2}{2\sigma^2}\right)$, where $A = \begin{bmatrix} 1.5 & -0.6 \\ 0.75 & 1.5 \end{bmatrix}$ $B = \begin{bmatrix} 0.1 & -1 \\ 1.2 & 0.05 \end{bmatrix}$, $\sigma = 0.5$, and $e_g\left(t\right) \triangleq z\left(t\right) - z_g$ is the regulation error. Without a loss of generality, the influencing agent dynamics in (5–2) are selected to evolve with $h\left(z\left(t\right), \eta\left(t\right)\right) = 0_{2\times 1}$, $g\left(\eta\left(t\right)\right) = I_2$, respectively. For parameter identification the basis functions are selected as $S_\eta\left(x\left(t\right)\right) = e_1\left(t\right)$ and $S_z\left(x\left(t\right)\right) = \tanh\left(V_z^T x\right)$, where $V_z \in \mathbb{R}^{4\times 5}$ is a constant weight matrix.[7] The StaF basis is selected as $\sigma\left(x, c\left(x\right)\right) = \left[\sigma_1\left(x, c_1\left(x\right)\right), \ldots, \sigma_5\left(x, c_7\left(x\right)\right)\right]^T$, where $\sigma_i\left(x\left(t\right), c_i\left(x\left(t\right)\right)\right) = x^T\left(t\right)\left(x\left(t\right) + 0.005\nu\left(x\left(t\right)\right)d_i\right)$, $\nu\left(x\left(t\right)\right) = \frac{x^T x}{(1+x^T x)}$, and d_i are the vertices of a 4-simplex. To perform BE extrapolation, a single trajectory $x_i\left(x\left(t\right), t\right)$ is selected at random from a uniform distribution over a $0.1\nu\left(x\left(t\right)\right) \times 0.1\nu\left(x\left(t\right)\right)$ square centered at the current state $x\left(t\right)$. The ICL approach in [55, 109] was used to perform system identification and satisfy Assumption 6.2. The selected initial conditions and parameters are provided in Table 6-1 and the results are shown in Figures 6-1-6-4.

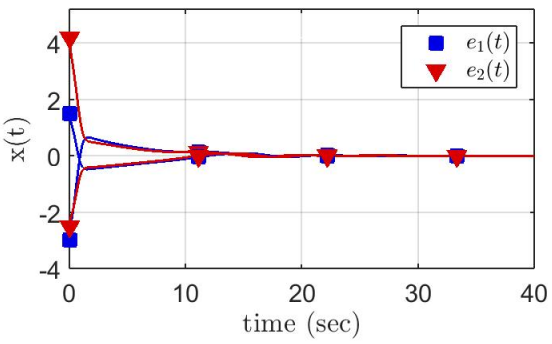
Discussion

Figure 6-1a shows that the concatenated state $x\left(t\right)$ converges to the origin, and hence the roaming agent is regulated to the goal location z_g, as shown in Figure 6-1b. Figures 6-1c and 6-1d show that the influencing agent control policy and approximated

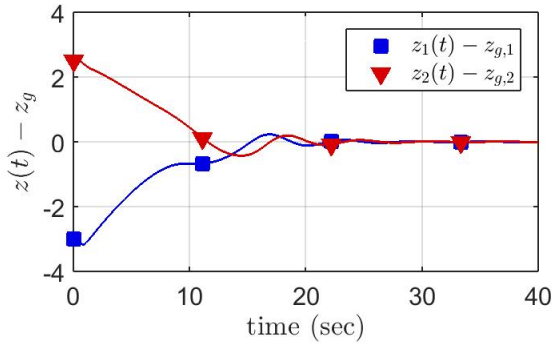
[7] Since the ideal basis to identify the roaming agent dynamics is not exactly known, there is a function approximation error $\varepsilon_z\left(x\left(t\right)\right)$ associated with this approximation.

Table 6-1. Simulation initial conditions and parameters.

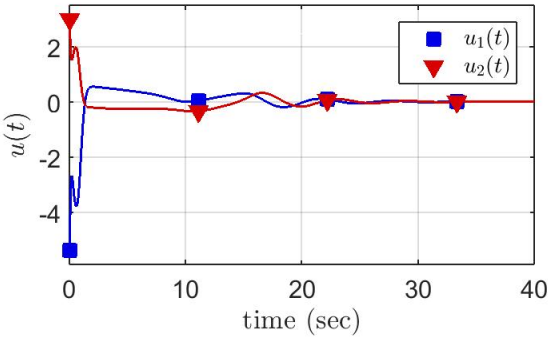
Initial conditions at $t_0 = 0$ $z(0) = [-3.0,\ 2.0]^T$, $\eta(0) = [0, 0.5]^T$, $z_g = [0, -0.5]^T$, $\hat{W}_c(0) = 1 \times 1_{5\times1}$, $\hat{W}_a(0) = 0.75 \times 1_{5\times1}$, $\hat{W}_d(0) = 0.7 \times 1_{5\times1}$, $\Gamma_c(0) = 0.05 I_5$, $\hat{\theta}(0) = U[-2, 3] \times 1_{7\times2}$, $V_z = U[-10, 10] \times 1_{5\times2}$.
Penalizing parameters $Q(x) = x^T Q_x x + \left(\max\left\{0, \|e_1(t)\|^2 - r_a^2\right\}\right)^2 + \left(\max\left\{0, \|e_g(t)\|^2 - r_a^2\right\}\right)^2$, $Q_x = \text{diag}\{1, 1, 3, 3\}$, $R = 5I_2$, $\gamma = 3$, $k_2 = 1.4$, $r_a = 0.02$.
Gains and parameters for ADP update laws $k_{c1} = 0.001$, $k_{c2} = 0.75$, $k_{a1} = 0.5$, $k_{d1} = 0.5$, $\gamma_1 = 0.005$, $\beta_c = 0.001$, $K_a = I_5$, $K_d = I_5$, $N = 1$.



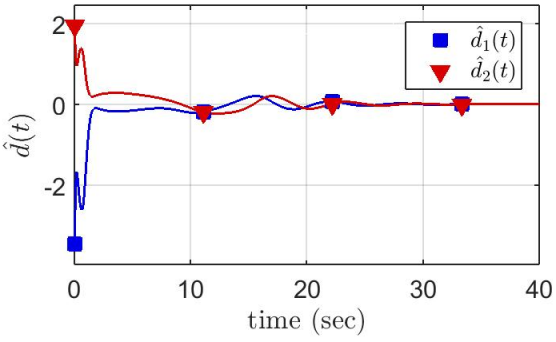
(a) State.



(b) Regulation error.

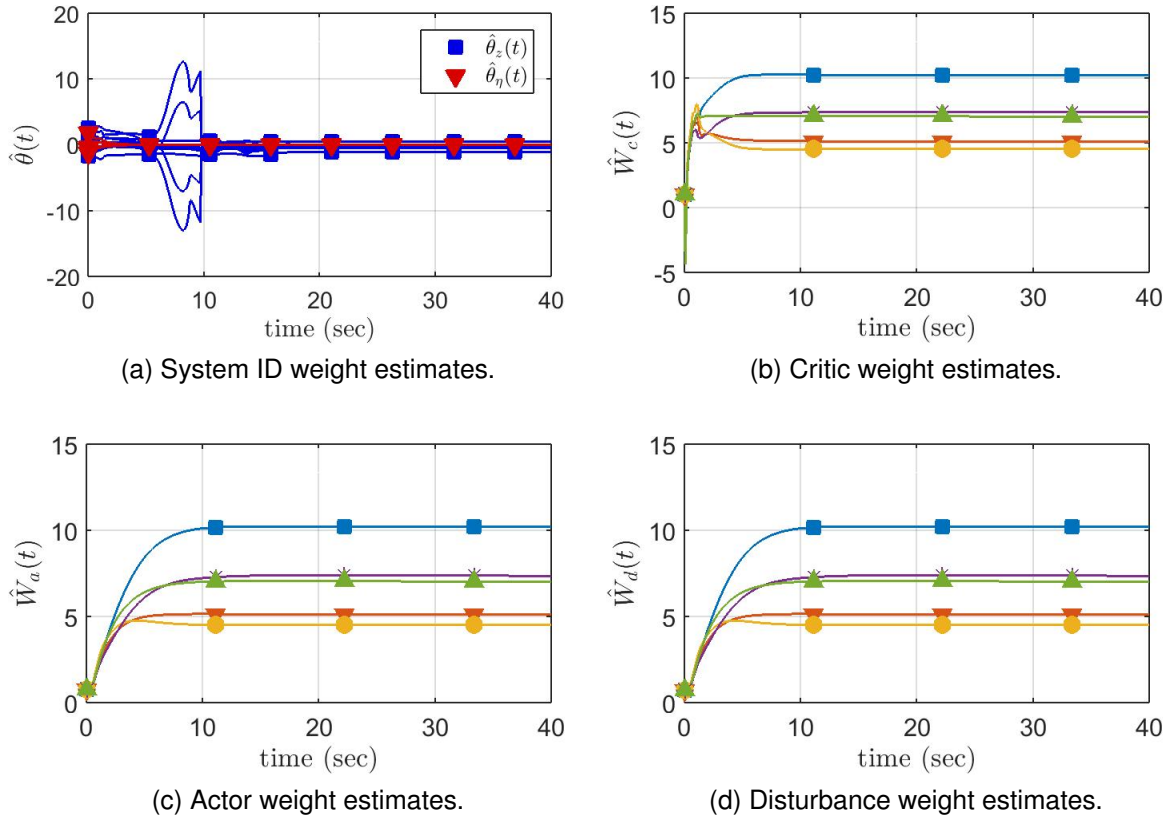


(c) Influencing agent policy.



(d) Approximate roaming agent disturbing policy.

Figure 6-1. The concatenated state $x(t)$, influencing agent policy $u(t)$, and approximate roaming agent disturbing policy $\hat{d}(t)$ all converge to the origin.



(a) System ID weight estimates.

(b) Critic weight estimates.

(c) Actor weight estimates.

(d) Disturbance weight estimates.

Figure 6-2. The critic $\hat{W}_c(t)$, actor $\hat{W}_a(t)$, disturbance $\hat{W}_d(t)$ StaF weight estimates, and system parameters $\hat{\theta}(t)$ remain bounded.

roaming agent disturbing policy remain bounded and converge once the roaming agent is regulated to the desired location. Figure 6-2a shows that the system parameter estimates converge to constant values. The influencing agent is approximated with a known basis, i.e. the true weights are $\theta_\eta = 0_{n\times n}$, and Figure 6-2a shows that the estimates $\hat{\theta}_\eta$ converge to the ideal values. However, the roaming agent is approximated with an unknown basis; hence, the estimates can only be said to be bounded and converge to steady-state values. Figures 6-2b-6-2d show that the critic, actor, and disturbance weight estimates remain bounded. The optimal StaF weights are unknown; thus, the estimates cannot be compared to their ideal values. The positions of the influencing and roaming agents are shown in Figures 6-3 and 6-4. The roaming agent is not motivated to go to the desired location z_g. Because the influencing agent starts

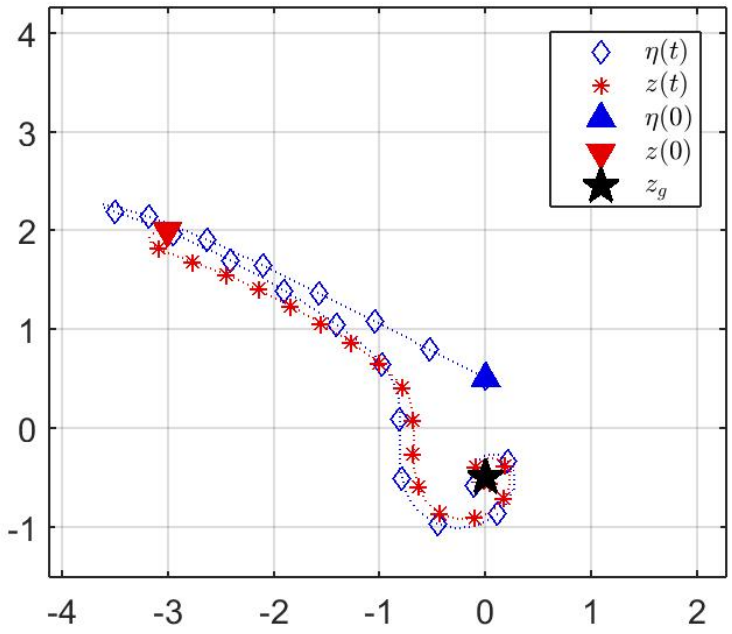


Figure 6-3. Positions of the influencing and roaming agents. The influencing agent (blue diamond) intercepts and drives the roaming agent (red asterisk) to the desired state (black star). The initial condition for the influencing agent is given by the blue triangle and the initial condition for the roaming agent is given by the red triangle.

approaching the roaming agent, the roaming agent starts moving away. The influencing agent then begins to chase the roaming agent and steers it toward z_g. Because the roaming agent first passes z_g, the influencing agent travels on the outside edge of the roaming agent to steer it back in the direction of z_g.

6.4.2 Roaming Agent Partially Known Basis and Worst-case Dynamics

To further demonstrate the result, two additional simulations were performed using the same gain conditions and dynamics in Section 6.4.1, where in one simulation the basis for system identification for roaming agents was partially known and set as $S_z\left(x\left(t\right)\right) = \Phi_2\left(x\left(t\right)\right)$, while in the other simulation the worst-case roaming agent dynamics as discussed in Remark 6.1. The influencing agent dynamics remained the same as in Section 6.4.1. Then, the total cost, and the instantaneous norms of the concatenated state $x\left(t\right)$, the regulation error $e_g\left(t\right)$, the influencing agent input $u\left(t\right)$,

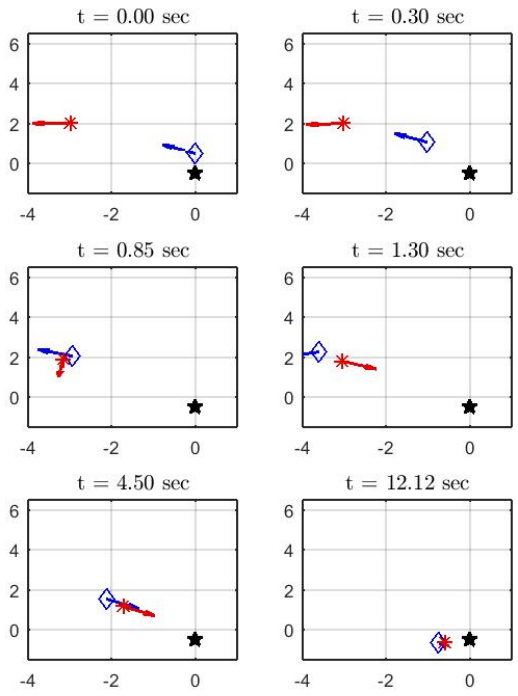
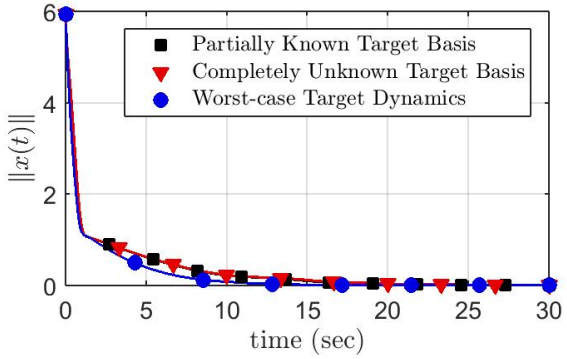


Figure 6-4. Sampled positions of the influencing and roaming agents. The influencing agent (blue diamond) intercepts and drives the roaming agent (red asterisk) to the desired state (black star).

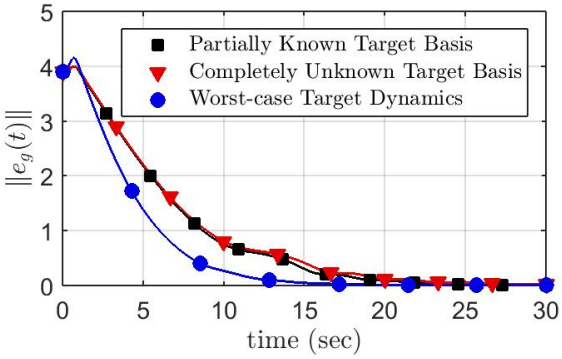
and estimated roaming agent disturbing input $\hat{d}(t)$ were calculated then plotted and shown in Figure 6-5. In addition, the total cost and total RMS values for the norms of the state, denoted by $\|x\|_{RMS}$, the error, denoted by $\|e_g\|_{RMS}$, input, denoted by$\|u\|_{RMS}$, and approximate target input, denoted by $\left\|\hat{d}\right\|_{RMS}$, are calculated and tabulated in Table 6-2.

Discussion

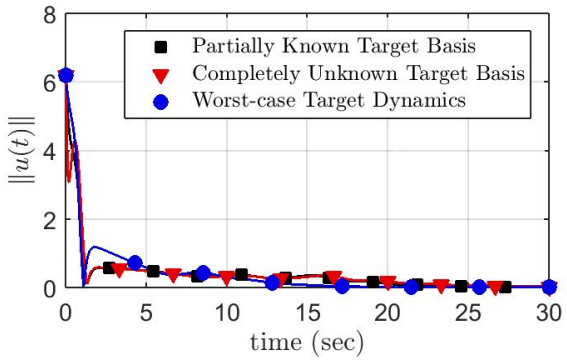
Comparing simulations performed with an unknown roaming basis, a partially known roaming basis, and a worst-case roaming dynamics, the results are shown in Figures 6-5a-6-5d and Table 6-2. Figures 6-5a and 6-5b show that the norms of the simulation using the worst-case roaming dynamics stated in Remark 6.1 results in the lowest state, $x(t)$, and error, $e_g(t)$, norms over the course of the simulation. While the simulations where the basis for the roaming residual dynamics where unknown resulted



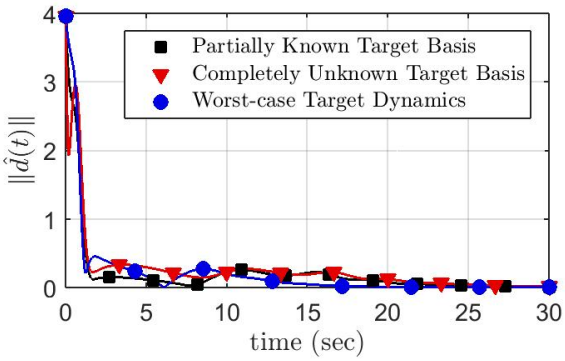
(a) Norm of the state $x\,(t)$.



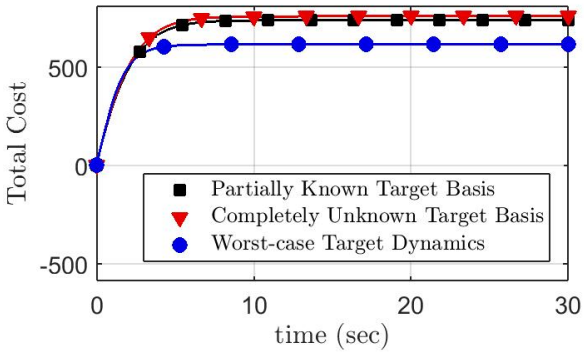
(b) Norm of the error $e_g\,(t)$.



(c) Norm of the input $u\,(t)$.



(d) Norm of the estimated policy $\hat{d}\,(t)$.



(e) Total cost comparison

Figure 6-5. Comparison of the norms for concatenated state $x\,(t)$, regulation error $e_g\,(t)$, influencing agent policy $u\,(t)$, and approximate roaming agent disturbing policy $\hat{d}\,(t)$, and total cost for the simulations in Sections 6.4.1,and 6.4.2.

Table 6-2. The total RMS values and total costs for each case study for the roaming agent dynamics in Sections 6.4.1 and 6.4.2.

	Simulation		
	Partially Known Basis	Unknown Basis	Worst-case Dynamics
Total Cost	737.70	758.10	614.70
$\|x\|_{RMS}$	0.39	0.42	0.36
$\|e_g\|_{RMS}$	0.81	0.82	0.69
$\|u\|_{RMS}$	0.43	0.42	0.49
$\left\|\hat{d}\right\|_{RMS}$	0.28	0.28	0.31

in the highest norms. This is further supported by Table 6-2, where the total RMS values are $\|x\|_{RMS} = 0.36$ for the state $x(t)$, and $\|e_g\|_{RMS} = 0.69$ for the error $e_g(t)$, when assuming the worst-case roaming dynamics. However, when considering a completely unknown basis for the roaming agent, the total RMS values for the norm of the state is $\|x\|_{RMS} = 0.42$ and the norm of the regulation error is $\|e_g\|_{RMS} = 0.82$. When partial information is consider about the roaming (i.e., the basis used for system identification is partially known), the total RMS values for the norms of the concatenated state $x(t)$ and error $e_g(t)$ lie between those of the previously mentioned cases, with values of $\|x\|_{RMS} = 0.39$ and $\|e_g\|_{RMS} = 0.81$, respectively. However, when looking at the norms of the influencing agent policy $u(t)$ and estimated roaming agent disturbing policy $\hat{d}(t)$, in Figures 6-5c and 6-5d, and the total RMS values in Table 6-2, an inverse relationship is seen. Specifically when the worst-case dynamics are considered, the RMS values $\|u\|_{RMS}$ and $\left\|\hat{d}\right\|_{RMS}$ are the highest at 0.49 and 0.31, respectively. When compared to the case when the roaming agent basis is assumed to be unknown, the RMS values are the lowest at $\|u\|_{RMS} = 0.42$ and $\left\|\hat{d}\right\|_{RMS} = 0.28$. This is because when the basis used to approximate the roaming agent dynamics is unknown, the system identification approach from [55, 109] and [109] compensates for some of the uncertainties and the influencing agent. Based on these estimates, the influencing agent does not command a high control effort to regulate the roaming agent. The effect of each assumption on

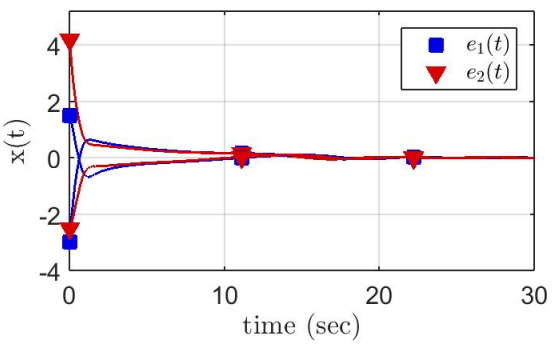
the knowledge of the roaming agent dynamics is further shown in Figure 6-5e, where the total costs are compared for each case in Sections 6.4.1 and 6.4.2. Specifically, it is shown that when the worst-case dynamics are assumed, the total cost is the lowest at 614.70; when the roaming agent basis is unknown, the total cost is the highest at 758.10; and when the roaming agent basis is partially known, the cost is in between the previously two mentioned costs at 737.70. When the worst-case roaming agent dynamics are considered, the influencing agent aims to regulate the roaming agent to z_g as quickly as possible.

6.4.3 Noisy Roaming Agent Dynamics

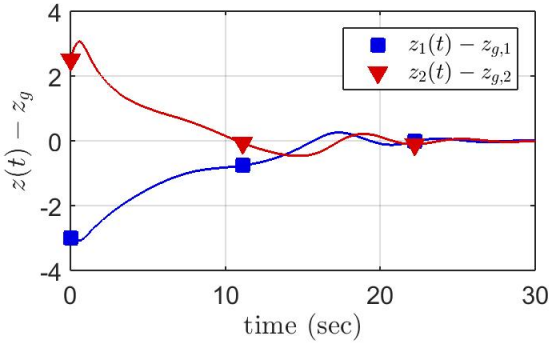
A simulation was also performed where the roaming agent dynamics in Section 6.4.1 are modified such $\Phi_1(x(t)) \triangleq e_1(t) \exp\left(-\frac{\|e_1(t)\|^2}{2}\right) + e_r(t) \exp(\omega_r(t))$, where $e_r(t) \triangleq [e_{1,1}(t)\, e_{g,1}(t),\ e_{1,2}(t)\, e_{g,2}(t)]^T$ and at each time instance $\omega_r(t)$ is selected from $U[-7, 0]\, 1_{1\times 1}$. Compared to the first three simulations, the roaming agent dynamics are now prone to noise. the constant γ was changed to $\gamma = 4$ and the simulation parameters are the same as those in Section 6.4.1. The results are displayed in Figures 6-6 and 6-8. A similar case-study to Section 6.4.2 is performed where the total cost, and total RMS values of the norms of the state $x(t)$, the error $e_g(t)$, the influencing agent input $u(t)$, and approximate roaming agent disturbing policy input $\hat{d}(t)$ were calculated for the three different cases: when the roaming agent's basis is completely unknown and $S_z(x(t)) = \tanh\left(V_z^T x\right)$ is used, when a partially known basis is assumed and set as $S_z(x(t)) = \Phi_2(x(t))$, and when the worst-case dynamics are assumed (i.e. $\Delta f(x(t)) = 0_{2\times 1}$). The results are shown in Figure 6-9 and Table 6-3.

Discussion

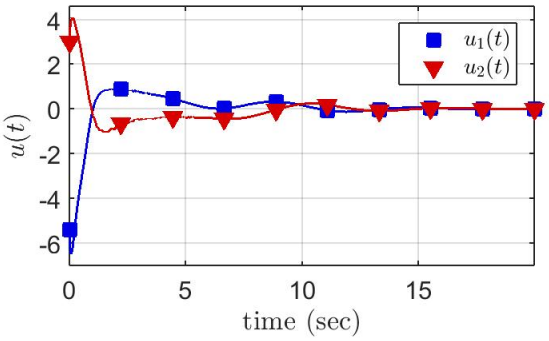
Figure 6-6a and Figure 6-6b show that the concatenated state $x(t)$, the regulation error $e_g(t)$ converge to zero. In addition, the influencing agent policy $u(t)$ and estimated roaming agent disturbing policy $\hat{d}(t)$ are shown to converge in Figures 6-6d and 6-6c, while the critic, actor, and disturbance weight estimates converge to steady-state



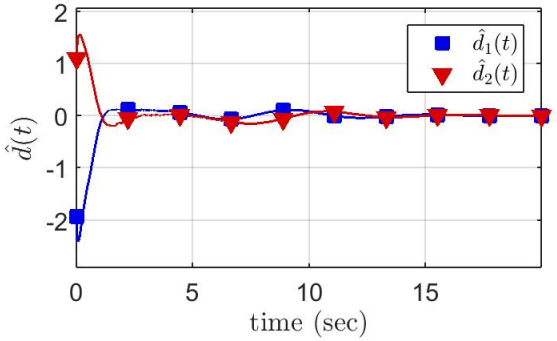
(a) Concatenated state.



(b) Regulation error.



(c) Influencing agent control policy.

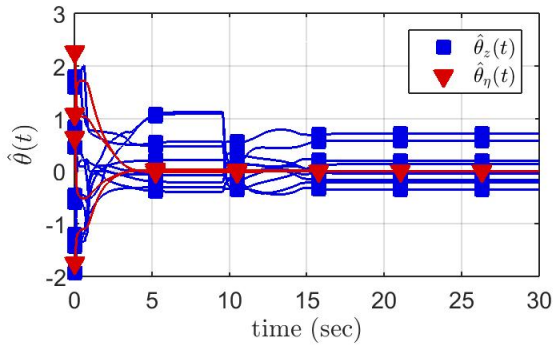


(d) Estimated roaming agent disturbing policy.

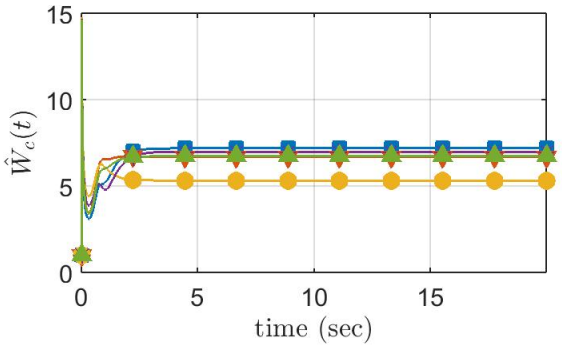
Figure 6-6. The total state $x\left(t\right)$, influencing agent policy $u\left(t\right)$, and approximate roaming agent policy $\hat{d}\left(t\right)$ for the noisy roaming agent dynamics all converge to the origin.

Table 6-3. The total RMS values and total costs for each case study for the noisy roaming agent dynamics in Section 6.4.3

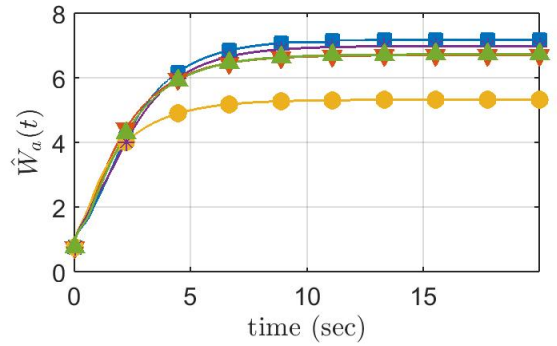
	Simulation		
	Partially Known Basis	Unknown Basis	Worst-case Dynamics
Total Cost	755.83	734.92	738.02
$\|x\|_{RMS}$	0.37	0.36	0.37
$\|e_g\|_{RMS}$	0.78	0.77	0.69
$\|u\|_{RMS}$	0.50	0.52	0.48
$\left\|\hat{d}\right\|_{RMS}$	0.19	0.20	0.17



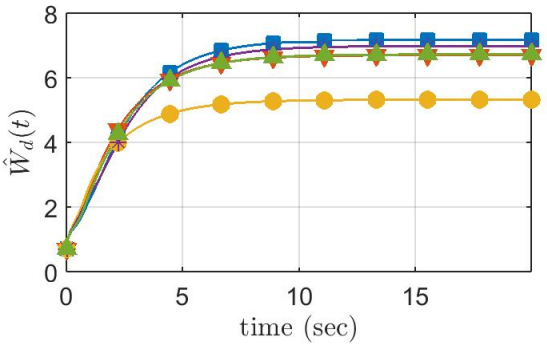
(a) System ID weight estimates.



(b) Critic weight estimates.



(c) Actor weight estimates.



(d) Disturbance weight estimates.

Figure 6-7. The critic, actor, disturbance StaF weight estimates, and the system identification estimates remain bounded.

values and remain bounded, and are shown in Figures 6-7b-6-7d. Moreover, the system parameter estimates are shown to be bounded in Figure 6-7a. The parameter estimates for the influencing agent converge to the ideal weights of $0_{2\times 2}$, which is known because the herding agent is modeled using $h\left(z\left(t\right),\eta\left(t\right)\right) = 0_{2\times 1}$, while the estimates for the roaming agent can only be shown to be bounded. This is because the exact basis is unknown due to the noise in $\Phi_1\left(x\left(t\right)\right)$ as discussed in Section 6.4.3.

Furthermore, Figure 6-9 shows the comparison of the concatenated state, regulation error, influencing agent control, and estimated roaming agent disturbing policy norms in addition to the cost for thirty seconds of the simulation. While Table 6-3 shows the total cost, and total RMS values of the norms of the state x, the error e_g, control policy u, and estimated disturbing policy $\hat{d}$.

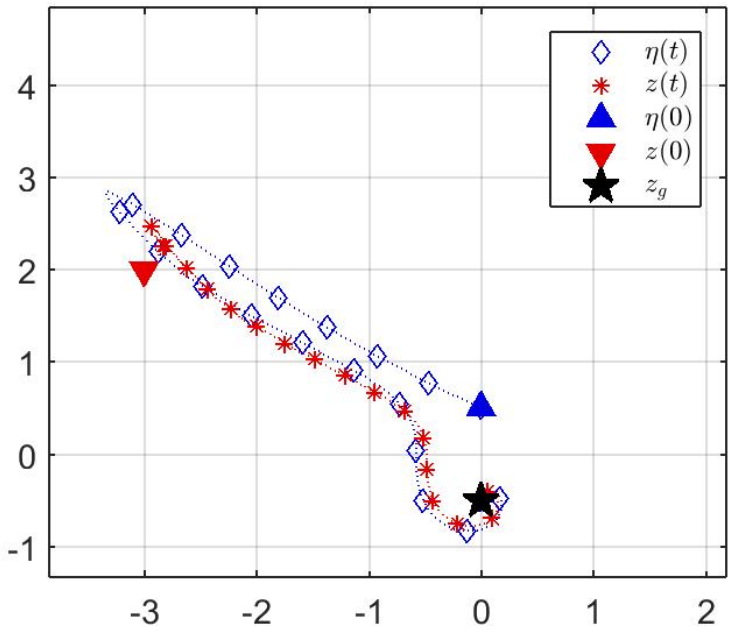
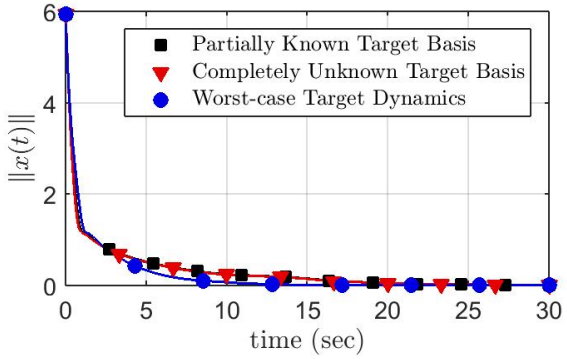
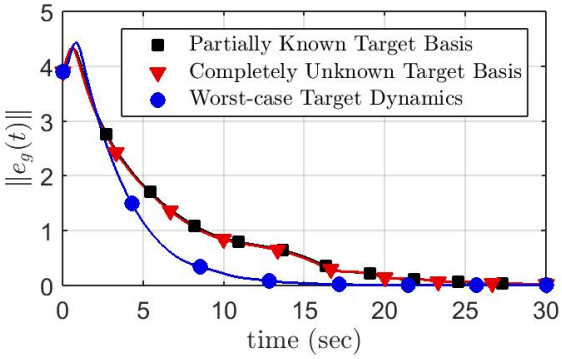


Figure 6-8. Positions of the influencing agent and roaming agent, which is modeled using noisy dynamics. The influencing agent (blue diamonds) intercepts and regulates the roaming agent (red asterisks) to the desired state (black star). The initial influencing agent condition is given by the blue triangle and the initial roaming agent is given by the red triangle.

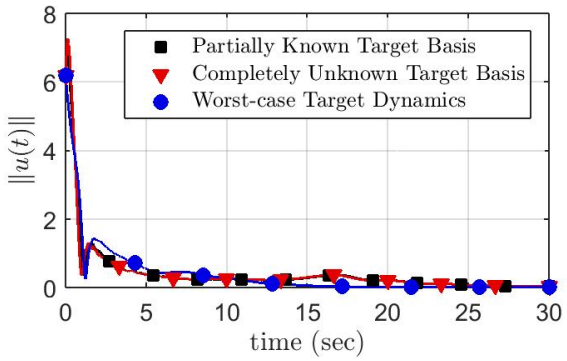
When comparing the three case studies of unknown basis, partially known basis, and worst-case roaming agent dynamics, Figure 6-9e and Table 6-3 show that for each case, the total cost was similar when an unknown basis and worst-case dynamics are considered, with total costs of 734.92 and 738.02, respectively. Moreover, when a partially known basis is considered, the cost is the highest, with a value of 755.83. Unlike the deterministic simulation in Sections 6.4.1 and 6.4.2, Table 6-3 shows that when the worst-case dynamics are considered, the total RMS value for the policies were the smallest, at 0.48, and 0.17, respectively. Also, similar to Sections 6.4.1 and 6.4.2, the total RMS for the regulation error is smallest at 0.37, while the total RMS error for the concatenated state is 0.37, which is the same when a partially known basis is considered.



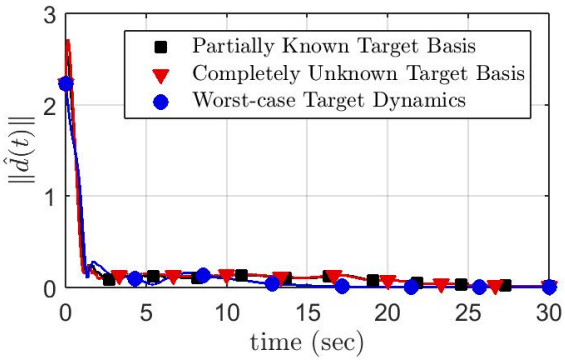
(a) Norm of the state $x\,(t)$.



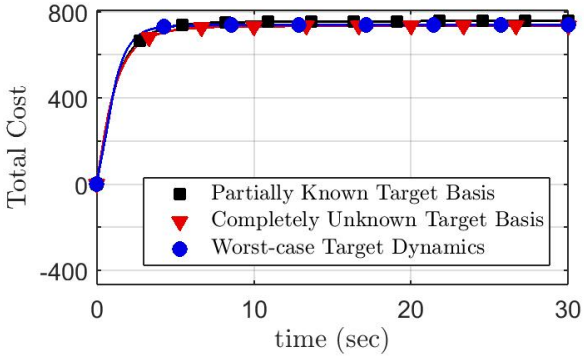
(b) Norm of the error $e_g\,(t)$.



(c) Norm of the herder input $u\,(t)$.



(d) Norm of the approximate target input $\hat{d}\,(t)$.



(e) Total cost comparison

Figure 6-9. Comparison of the norms of the concatenated state, regulation error, influencing agent policy, and approximate worst-case roaming agent policy, and total cost, for the system modeled with noisy roaming agent dynamics.

6.5 Concluding Remarks

A game-based indirect regulation problem is investigated for an influencing agent which tracks, intercepts, and steers a roaming agent to a user-defined goal location. The problem is posed as a minimax optimization problem where the optimal influencing agent policy and worst-case roaming agent disturbing policy are approximated using the StaF ADP-based approximation method. UUB convergence is shown via a Lyapunov stability analysis for the closed-loop error system. Simulation results for a two-state influencing and roaming agents are included which illustrate the performance of the developed method for an uncertain roaming agent.

CHAPTER 7
CONCLUSIONS

RL is an extremely popular and powerful tool for learning uncertainties and optimal policies for systems. While many advances have been made in the field of RL and ADP, challenges still arise when trying to implement such methods online on hardware. Specifically, traditional ADP methods aim to approximate the solution to the optimal control problem over the entire operating domain, which can be computationally infeasible. In this dissertation, a computationally efficient, local StaF approximation method is used to perform local approximations of the value function; which is shown to be implementable through various experimental results. Another challenge in implementing learning-based controllers online, is that in the context of RL, the agent learns the policies from input-output data. Therefore, trade-off between exploration and exploitation needs to be considered when designing RL-based policies. Specifically, traditional approaches aim to provide exploration by trying to satisfy the PE condition. One method to satisfy this condition is to inject noisy potentially destabilizing signals into the physical system. However, even with the injection of noisy signals, the PE condition cannot be shown to be satisfied. However, in this dissertation, virtual excitation is performed where off-policy trajectories are selected in a neighborhood of the agent's state, and then extrapolated BEs are used to provide exploration for the system. Compared the PE condition, the virtual excitation can be checked online by measuring the minimum eigenvalue of the extrapolated BE history stack, and since the off-policy trajectories are excited, the system is not potentially destabilized.

In Chapter 3, a novel framework is developed where the value function is approximated via combination of traditional function approximation techniques (i.e., R-MBRL) and local approximation method (i.e., StaF). The operating domain is divided into two sets: a set A, which is a desired region where the agent will end up, and a set $B = \chi \setminus A$, which is the complement. The traditional R-MBRL method is used to approximate the

value function in A before the agent arrives there, while the StaF method is utilized in B to provide the agent with local stability. A Lyapunov-based stability analysis shows that under certain conditions, the closed-loop system is stable such that the states, policy, and weight estimation errors are uniformly ultimately bounded. Extensive simulation results are provided, and show that the choice of approximation method depends on many factors. A limitation in Chapter 3 is that while using StaF, when an agent visits an area more than once, it needs to relearn the weights, which is inefficient. Motivated by the fact that the R-MBRL gives a global policy, it is desired to learn the value function over the entire operating domain. However, the computational complexities associated with this can limit R-MBRL from being implemented on hardware. This motivates the use of sparse NNs, such as [122, 123], to approximation the value function in segmented parts of the operating domain as the agent enters them. Sparse representations of the value function allow only a certain part of the basis to be active, depending on the agents location, thus potentially reducing computational issues. Further research is required to derive such ADP methods using sparse NNs.

In Chapter 4, motivated by the local approximation nature of the StaF method, a path-planning strategy is developed for an agent which encounters uncertain mobile obstacles. The problem is posed as an autonomous infinite-horizon optimal control problem with an agent subject to control saturations and collision penalizations. First, both the number of obstacles and the dynamics are considered to be known. Motivated by the fact that the agent may encounter a given obstacles more than once, a basis is given to each obstacle. A basis is turned off when the corresponding obstacle is not sensed. However, knowing the number of obstacles and their dynamics is limiting in application; hence, an extension is provided to alleviate these two assumptions. Specifically, the value function is interpreted as a time-varying map by removing the direct consideration of the obstacle states. Then, time is mapped to a compact set and and the mapping is used as the input to the StaF approximation. A Lyapunov-based

stability analysis shows UUB convergence of the closed-loop system, and based on the design of the collision penalizing function, collisions are also avoided. Experimental results are provided to show the validity of the result. In the experiments, a Parrot Bebop 2.0 quadcopter was used to navigate to the origin, while avoiding virtual obstacles. A limitation of the developed approach in Chapter 4 is that continuous agent dynamics are considered. However, in practice many systems have discontinuous or hybrid dynamics. For instance, the agent's initial aim may be to navigate to a desired location, but as it senses an obstacle, the dynamics and problem may switch such that the agent is required to flee from the obstacle to prevent collision. Tools from formal methods can be utilized to develop strategies where the system and control problem is no longer continuous. Methods such as [124] exist which aim to use ADP with formal methods for hybrid states. However, the computational complexity associated with them is still an open problem; hence, the development and analysis of such methods with applications in real-time RL-based systems remains a subject for future research.

In Chapters 5 and 6, indirect regulation problems are considered. Specifically, a policy is generated for an influencing agent to regulate a roaming agent to a goal location (i.e., indirectly herd a roaming agent). In such problems, the roaming agent is not motivated to go that location itself. Hence, the influencing agent both pursues and then guides the roaming agent to the goal location through inter-agent interaction dynamics. In Chapter 5, an indirect approach is taken where a virtual state with a virtual policy are designed to regulate the roaming agent to the goal. The influencing agent is then tasked with tracking the virtual state and desired input. Both the influencing and roaming agents are considered to have uncertain dynamics. To overcome this challenge, a data-based parameter identification approach, called integral concurrent learning, is used to learn the functions online from input-output data. ICL alleviates the needs to inject a probing signal to facilitate learning because the method uses a history stack of input-output data, and the minimum eigenvalue corresponds to data richness. A

stability analysis shows that the closed-loop system is UUB and that the roaming agent is regulated to a neighborhood of the goal location. Simulation and experimental results are provided to show the performance of the system.

Compared to the indirect approach in Chapter 5, a game-based approach is taken in Chapter 6. Specifically, two error systems are developed, one which the influencing agent aims to intercept the roaming agent, and the second where the influencing agent aims to drive the roaming agent to the goal location. The problem is posed as a minimax optimization problem, where the influencing agent policy is the minimizing policy and the roaming agents policy is the maximizing policy. By forming the problem in this form, the goal was to find the worst-case roaming agent disturbing policy. An actor-critic-disturber method was used to estimate the unknown StaF weights, a Lyapunov-based stability analysis shows that the closed loop system is UUB, and a signal chasing argument shows that the roaming agent is regulated to the goal location. Simulations are provided to show the performance of the developed strategy. Specifically, three cases are considered in the simulation: the cases when the roaming agent basis used for function approximation is unknown and partially known, and the case when the worst-case roaming agent dynamics are considered. Another simulation is performed where the roaming agent is subject to noise in its dynamics, and the results show the influencing agent still guides the roaming agent to the goal location.

A limitation of Chapters 5 and 6 is that only one influencing and one roaming agent are considered. However, it may be beneficial to consider problems when there are multiple of each type of agent. Specifically, the problem when there are more roaming agents compared to influencing agents is a challenging one. Methods that use adaptive or robust control formulations exist for such problems (cf. [55–57, 94]), however they do not considers optimality. Moreover, methods that consider optimality with continuous-time and space RL to indirectly regulate multiple roaming agents in real-time have yet to be explored. The challenge in such a problem is that the influencing agents would

need to switch between the selected roaming agent they are trying to regulate. Future research still needs to be conduct to study the effects of switched systems based formulations of optimal control problems where learning is involved.

In all chapters of this dissertation, the considered systems are continuous and deterministic. However, this is a limiting factor as most “real-world” systems are prone to stochastic or hybrid dynamics. In addition, implementation of RL during learning in real-time systems is difficult as safety and reliability become much more important. Formal methods, which focus on analyzing hybrid systems and their stability by the use of computational tools, provide certain guarantees for systems. Therefore, they may be a valuable tool to address safety and reliability in real-time data-based learning systems. This motivates an investigation into using formal methods in conjunction with computationally efficient function approximation techniques to develop real-time data-based approximately optimal controllers with safety and performance guarantees for autonomous agents. With the massive rise of autonomy over the last several years, besides the applications in this dissertation, data-based learning methods can play a major role in incorporating more intelligence-driven autonomy in everyday systems.

APPENDIX: A
AUXILIARY TERMS AND SUFFICIENT CONDITIONS

A.1 Auxiliary terms and Sufficient Conditions for Chapter 3

In the following, $\overline{\|h\|} \triangleq \sup_{\xi \in B_\zeta} \|h(\xi)\|$, for some continuous function $h : \mathbb{R}^n \to \mathbb{R}^k$, where $B_\zeta \subset \mathbb{R}^{n+2L+2P}$ denotes a closed ball with radius ζ centered at the origin. The sufficient conditions that facilitate the stability analysis in Section 3.5 are given by

$$\frac{(k_{a1}+k_{a2})}{2} \geq \left(\left(\frac{k_{c1}}{\sqrt{\gamma_2}}+\frac{\eta_{c1}}{\sqrt{\gamma_1}}\right)\vartheta_5+\frac{k_{c1}}{\sqrt{\gamma_2}}\vartheta_6\right)\overline{\nu}_l\left(\|Z(t_0)\|\right)+2\vartheta_1+\vartheta_2+\frac{\vartheta_4\overline{\|W_2\|}}{\sqrt{\gamma_2}}, \tag{A–1}$$

$$\frac{(\eta_{a1}+\eta_{a2})}{2} \geq \left(\frac{\eta_{c1}}{\sqrt{\gamma_1}}\vartheta_6+\left(\frac{\eta_{c1}}{\sqrt{\gamma_1}}+\frac{k_{c1}}{\sqrt{\gamma_2}}\right)\vartheta_7\right)\overline{\nu}_l\left(\|Z(t_0)\|\right)+\left(\frac{1}{\underline{\Gamma_2}}+1\right)\vartheta_2+\frac{\vartheta_3\overline{W_1}}{\sqrt{\gamma_1}}, \tag{A–2}$$

$$\frac{k_{c2}\underline{b}}{4} \geq \max\left\{\frac{\vartheta_2}{2\underline{\Gamma_2}}+\frac{k_{c1}(\vartheta_5+\vartheta_6+\vartheta_7)}{4\sqrt{\gamma_2}}+\frac{\vartheta_{10}}{2},\frac{\left(k_{a1}+\vartheta_9\right)^2}{(k_{a1}+k_{a2})}\right\}, \tag{A–3}$$

$$\frac{\eta_{c2}\underline{c}}{4} \geq \max\left\{\frac{\eta_{c1}(\vartheta_5+\vartheta_6+\vartheta_7)}{4\sqrt{\gamma_1}}+\frac{\vartheta_{10}}{2},\frac{\left(\eta_{a1}+\frac{\vartheta_3}{2\sqrt{\gamma_1}}\overline{W_1}\right)^2}{(\eta_{a1}+\eta_{a2})}\right\}, \tag{A–4}$$

and

$$\nu_l^{-1}(\iota) < \overline{\nu_l}^{-1}(\underline{\nu_l}(\zeta)). \tag{A–5}$$

In (A–1)-(A–5), the constants $\iota, \{\vartheta_i | i = 1, \ldots 12\} \in \mathbb{R}_{>0}$ are defined as

$$\begin{aligned}
\vartheta_1 &= \frac{\overline{\|G_{\nabla W_2 \nabla \phi}\|}}{2}+\frac{\overline{\|G_{\nabla W_2 \nabla \lambda \phi^T}\|}}{2},\\
\vartheta_2 &= \frac{\overline{\|G_{\nabla W_2 \nabla \sigma}\|}}{2}+\frac{\overline{\|G_{\nabla W_2 \nabla \lambda \sigma^T}\|}}{2},\\
\vartheta_3 &= \frac{\eta_{c1}+\eta_{c2}}{4}\overline{\|G_{\nabla\sigma}\|}, \qquad \vartheta_4 = \frac{k_{c1}+k_{c2}}{4}\overline{\|G_{\nabla\phi}\|},\\
\vartheta_5 &= \frac{\overline{\|G_{\nabla\phi}\|}}{2}+\frac{\overline{\|\phi G_{\nabla\lambda\nabla\phi}\|}}{4}+\frac{\overline{\|\phi G_{\nabla\lambda\nabla\lambda\phi^T}\|}}{4},\\
\vartheta_6 &= \frac{\overline{\|G_{\nabla\phi\nabla\sigma}\|}}{2}+\frac{\overline{\|\phi G_{\nabla\lambda\nabla\lambda\sigma^T}\|}}{2}+\frac{\overline{\|\phi G_{\nabla\lambda\nabla\sigma}\|}}{2}+\frac{\overline{\|\sigma G_{\nabla\lambda\nabla\phi}\|}}{2},\\
\vartheta_7 &= \frac{\overline{\|G_{\nabla\sigma}\|}}{2}+\frac{\overline{\|\sigma G_{\nabla\lambda\nabla\lambda\sigma^T}\|}}{4}+\frac{\overline{\|\sigma G_{\nabla\lambda\nabla\sigma}\|}}{2},
\end{aligned}$$

$$\vartheta_8 = \overline{\|\nabla W_2 f\|} + \frac{1}{2}\overline{\|G_{\nabla V^* \nabla \phi}\|} + \frac{1}{2}\overline{\|G_{\nabla V^* \nabla \lambda \phi^T}\|} + \vartheta_2 \overline{W_1} + \vartheta_1 \overline{\|W_2\|} + \frac{\vartheta_4}{2\sqrt{\gamma_2}} \overline{\|W_2\|}^2,$$

$$\vartheta_9 = \frac{\vartheta_1}{\underline{\Gamma_2}} + \frac{\vartheta_4}{2\sqrt{\gamma_2}} \overline{\|W_2\|},$$

$$\vartheta_{10} = \frac{k_{c1}}{2\sqrt{\gamma_2}} \overline{\|\omega_{\nabla \sigma}\|} + \frac{\eta_{c1}}{2\sqrt{\gamma_1}} \overline{\|\omega_{\nabla \phi}\|},$$

$$\vartheta_{11} = \frac{\vartheta_2}{\underline{\Gamma_2}} \overline{W_1} + \frac{\vartheta_1}{\underline{\Gamma_2}} \overline{\|W_2\|} + \frac{\overline{\|\nabla W_2 f\|}}{\underline{\Gamma_2}},$$

$$\vartheta_{12} = \frac{\overline{\|G_{\nabla V^* \nabla \sigma}\|}}{2} + \frac{\overline{\|G_{\nabla V^* \nabla \lambda \sigma^T}\|}}{2} + \eta_{a2} \overline{W_1} + \frac{\vartheta_3}{2\sqrt{\gamma_1}} \overline{W_1}^2,$$

and

$$\iota = \frac{\overline{\|G_{\nabla V^* \nabla W_2 \phi}\|}}{2} + \frac{\overline{\|G_{\nabla V^* \nabla \epsilon}\|}}{2} + \frac{\left(k_{a2} + \vartheta_8\right)^2}{(k_{a1} + k_{a2})} + \frac{\left(\frac{\eta_{c1}}{2\sqrt{\gamma_1}} \left(2\overline{\|\Delta_1\|} + \overline{\|\Delta_2\|} + \overline{\|\Delta_3\|}\right)\right)^2}{\eta_{c2} \underline{c}}$$
$$+ \frac{\left(\vartheta_{12}\right)^2}{(\eta_{a1} + \eta_{a2})} + \frac{\left(\frac{k_{c1}}{2\sqrt{\gamma_2}} \left(\overline{\|\Delta_1\|} + 2\overline{\|\Delta_2\|} + \overline{\|\Delta_3\|}\right) + \vartheta_{11}\right)^2}{k_{c2} \underline{b}}.$$

The sufficient condition in (A–1) can be satisfied by increasing the gain k_{a2}. This will not affect the sufficient conditions in (A–2) and (A–4) and it may decrease the sufficient condition in (A–3). The sufficient condition in (A–2) can be satisfied without affecting the sufficient conditions (A–1) and (A–3) by increasing the gain η_{a2}. The sufficient condition in (A–3) can be satisfied by selecting points for BE extrapolation in $B \subset \chi \setminus A$ so that the minimum eigenvalue $\underline{b}$ in (3–27) is large enough and by increasing the gain k_{a2}. By selecting points for BE extrapolation in $A \subset \chi$ such that the minimum eigenvalue, $\underline{c}$, is large enough, and a large η_{a2}, the sufficient condition in (A–4) can be satisfied. Provided the transition function λ is selected such that $\overline{\nabla \lambda}$ is small, the basis functions used for approximation are selected such that $\overline{\|\epsilon\|}$, $\overline{\|\nabla \epsilon\|}$, and $\overline{\|\nabla W_2\|}$ are small, and k_{a2}, η_{a2}, $\underline{c}$,

and $\underline{b}$ are selected to be sufficiently large, then the sufficient condition in (A–5) can be satisfied. [1]

A.2 Auxiliary Terms for Chapter 4

In Section 4.4 the positive constants $\iota, \varphi_{ac} \in \mathbb{R}_{>0}$ are introduced, which are defined as

$$\iota \triangleq \frac{\iota_c^2}{k_{c2}\underline{c}} + \frac{\left(\iota_{a1}+\iota_{a2}\right)^2}{k_{a1}+k_{a2}} + \overline{\left\|W^T\nabla\sigma + \sigma^T\nabla W + \nabla\epsilon + \nabla P_a\right\|}\frac{G_R}{2}\overline{\left\|\nabla W^T\sigma + \nabla\varepsilon^T\right\|}m,$$

and

$$\varphi_{ac} \triangleq k_{a1} + \frac{\overline{\|\nabla W\|}}{\underline{\Gamma}}\frac{G_R}{2}\overline{\|\nabla\sigma^T\|} + k_{c1}\sqrt{\frac{m}{\gamma_1}}\mu_{sat}\overline{\|\nabla\sigma G\|} + \frac{k_{c2}}{N}\sum_{k=1}^{N}\sqrt{\frac{m}{\gamma_1}}\mu_{sat}\overline{\|\nabla\sigma_k G_k\|},$$

where

$$\begin{aligned}
\iota_{a1} &\triangleq \overline{\left\|W^T\nabla\sigma + \sigma^T\nabla W + \nabla\epsilon + \nabla P_a\right\|}\frac{G_R}{2}\overline{\|\nabla\sigma^T\|} + \frac{\overline{\|\nabla W F\|}}{\lambda_{\min}\{\Gamma_a\}} \\
&\quad + \frac{\overline{\|\nabla W\|}}{\lambda_{\min}\{\Gamma_a\}}\frac{G_R}{2}\overline{\|\nabla P_a + \nabla\sigma^T W\|}, \\
\iota_{a2} &\triangleq k_{a2}\|W\| + k_{c1}\mu_{sat}\overline{\|\nabla\sigma G\|}\sqrt{\frac{m}{\gamma_1}}\|W\| + \frac{k_{c2}}{N}\sum_{k=1}^{N}\mu_{sat}\overline{\|\nabla\sigma_k G_k\|}\sqrt{\frac{m}{\gamma_1}}\overline{\|W\|}, \\
\iota_c &\triangleq \frac{\overline{\|\nabla W F\|}}{\underline{\Gamma}} + \frac{\overline{\|\nabla W\|}}{\underline{\Gamma}}\frac{G_R}{2}\overline{\|\nabla P_a + \nabla\sigma^T W\|} + k_{c1}\frac{1}{2\sqrt{\gamma_1}}\overline{\|\Delta\|} + \frac{k_{c2}}{N}\sum_{k=1}^{N}\frac{1}{2\sqrt{\gamma_1}}\overline{\|\Delta_k\|}.
\end{aligned}$$

[1] The minimum eigenvalue of $\frac{1}{N}\sum_{i=1}^{N}\frac{\omega_{\nabla\sigma i}(t)\omega_{\nabla\sigma i}^T(t)}{\rho_{1i}^2(t)}$ can be increased by collecting redundant data, i.e. selecting $N \gg P$ in the area of interest. The bound on the gradient of λ, i.e. $\overline{\nabla\lambda}$, can be decreased by selecting larger transition regions $A' \setminus A$.

A.3 Auxiliary Terms for Chapter 5

To facilitate the analysis in Section 5.3, $\kappa \in \mathbb{R}_{>0}$ is defined as $\kappa \triangleq \min\left\{\frac{q}{2}, \frac{k_\theta c_{\theta 1}}{16}, \frac{k_{c2}\underline{c}}{8}, \frac{(k_{a1}+k_{a2})}{8}\right\}$ where the constants $\varphi_a, \varphi_{ac}, \varphi_{c\theta} \in \mathbb{R}_{>0}$ are defined as

$$\varphi_a \triangleq \frac{3\sqrt{3}\left(k_{c1}+k_{c2}\right)}{64}\frac{\overline{\|G_\sigma\|\|W\|}}{\sqrt{\gamma_1}} + \frac{\overline{\|\nabla W G_R \nabla \sigma^T\|}}{2\lambda_{\min}\{K_a\}},$$

$$\varphi_{ac} \triangleq k_{a1} + \frac{3\sqrt{3}\left(k_{c1}+k_{c2}\right)}{64}\frac{\overline{\|G_\sigma\|\|W\|}}{\sqrt{\gamma_1}} + \frac{\overline{\|\nabla W\|\|G_R\|\|\nabla \sigma^T\|}}{2\underline{\Gamma}_c},$$

and

$$\varphi_{c\theta} \triangleq \frac{3\sqrt{3}\left(k_{c1}+k_{c2}\right)\left(\overline{\|W\|\|\nabla\sigma\|\|S\|}\left(1+\frac{1}{k_d}+\overline{\|g\|\|g^+\|}\right)\right)}{16\sqrt{\gamma_1}}.$$

Furthermore, the constant $\iota \in \mathbb{R}_{>0}$ is defined as $\iota \triangleq \frac{1}{2}\overline{\|\nabla V^*\|\|G_R\|\|\nabla W^T\sigma + \nabla\epsilon^T\|} + \frac{(\iota_{a1}+\iota_{a2})^2}{(k_{a1}+k_{a2})} + \frac{\iota_c^2}{k_{c2}\underline{c}} + \frac{k_\theta v_1^2}{c_{\theta 1}}$, where

$$\iota_c \triangleq \frac{\overline{\|\nabla W\|}\left(1+\frac{1}{k_d}+\overline{\|g\|\|g^+\|}\right)\left(\overline{\theta S}+\varepsilon\right)}{\underline{\Gamma}_c} + \frac{\overline{\|\nabla W\|\|G_R\|\|\nabla\sigma^T W\|}}{2\underline{\Gamma}_c} + \frac{3\sqrt{3}\left(k_{c1}+k_{c2}\right)\overline{\|\Delta\|}}{16\sqrt{\gamma_1}},$$

$$\iota_{a1} \triangleq \frac{\overline{\|\nabla W\|}\left(1+\frac{1}{k_d}+\overline{\|g\|\|g^+\|}\right)\left(\overline{\theta S}+\bar{\varepsilon}\right)}{\lambda_{\min}\{K_a\}} + \frac{\overline{\|\nabla W\|\|G_R\|\|\nabla\sigma^T W\|}}{2\lambda_{\min}\{K_a\}},$$

and

$$\iota_{a2} \triangleq k_{a2}\overline{\|W\|} + \frac{3\sqrt{3}\left(k_{c1}+k_{c2}\right)}{64}\frac{\overline{\|G_\sigma\|\|W\|}^2}{\sqrt{\gamma_1}} + \frac{\overline{\|\nabla V^*\|\|G_R\|\|\nabla\sigma\|}}{2}.$$

A.4 Auxiliary Terms for Chapter 6

To facilitate the analysis in Section 6.3, $\kappa, \varphi_{c\theta}, \iota \in \mathbb{R}_{>0}$ are defined as $\kappa \triangleq \frac{1}{8}\min\left\{\underline{q}, k_{c2}\underline{c}, k_{a1}, k_{d1}, K_\theta\right\}$, $\varphi_{c\theta} \triangleq \max\left\{k_{c1}, k_{c2}\right\}\frac{\lambda_{\max}\{L\}}{2\sqrt{\gamma_c}}\overline{\|W\|\|\nabla\sigma\|}\overline{S}$, and

$\iota \triangleq \frac{\iota_x^2}{\underline{q}} + \frac{\iota_c^2}{k_{c2}} + \frac{\iota_a^2}{k_{a1}} + \frac{\iota_d^2}{k_{d1}} + \frac{D_\theta^2}{K_\theta}$. Furthermore, the constants $\underline{c}, \iota_x, \iota_c, \iota_a, \iota_a \in \mathbb{R}_{>0}$ are defined as $\underline{c} \triangleq \left(\frac{\beta_c}{2k_{c2}\overline{\Gamma}_c} + \frac{c_1}{2}\right)$,

$$\iota_x \triangleq \frac{\alpha}{2}\overline{\left\|K_\gamma \nabla\sigma^T \hat{W}_d - G_R \nabla\sigma^T \hat{W}_a\right\|},$$

$$\iota_c \triangleq \frac{\|\nabla W\|}{\underline{\Gamma}_c}\left(\overline{S\theta} + \overline{\varepsilon}_\eta + \frac{\overline{\left\|K_\gamma \nabla\sigma^T \hat{W}_d - G_R \nabla\sigma^T \hat{W}_a\right\|}}{2}\right)$$
$$+ \frac{\max\{k_{c1}, k_{c2}\}}{2\sqrt{\gamma_c}}\left(\overline{\|\Delta\|} + \frac{1}{4}\overline{\left\|\tilde{W}_a\right\|}\,\overline{\|G_\sigma\|}\,\overline{\left\|\tilde{W}_a\right\|} + \frac{1}{4}\overline{\left\|\tilde{W}_d\right\|}\,\overline{\|K_\sigma\|}\,\overline{\left\|\tilde{W}_d\right\|}\right),$$

$$\iota_a \triangleq \frac{\lambda_{\max}\{L\}}{\lambda_{\min}\{K_a\}}\|\nabla W\|\left(\overline{S\theta} + \overline{\varepsilon}_\eta + \frac{\overline{\left\|K_\gamma \nabla\sigma^T \hat{W}_d - G_R \nabla\sigma^T \hat{W}_a\right\|}}{2\lambda_{\max}\{L\}}\right),$$

and

$$\iota_d \triangleq \frac{\lambda_{\max}\{L\}}{\lambda_{\min}\{K_d\}}\|\nabla W\|\left(\overline{S\theta} + \overline{\varepsilon}_\eta + \frac{\overline{\left\|K_\gamma \nabla\sigma^T \hat{W}_d - G_R \nabla\sigma^T \hat{W}_a\right\|}}{2\lambda_{\max}\{L\}}\right).$$

Remark A.1. If the value function is assumed to be bounded as in Remark 6.3, the first sufficient conditions in (6–23) reduce to $\underline{c} \geq \frac{k_{a1}+k_{d1}+\varphi_{c\theta}}{k_{c2}}$ and $K_\theta \geq \varphi_{c\theta}$. However, the bound ι changes to $\iota = \iota_x + \frac{\iota_c^2}{k_{c2}} + \frac{\iota_a^2}{k_{a1}} + \frac{\iota_d^2}{k_{d1}} + \frac{D_\theta^2}{K_\theta}$, where ι_x is redefined as $\iota_x = \frac{1}{4}\alpha^2\overline{\|(K_\gamma - G_R)\|} + \frac{\alpha}{2}\overline{\left\|K_\gamma \nabla\sigma^T \hat{W}_d - G_R \nabla\sigma^T \hat{W}_a\right\|}$. This bound is larger compared to the previously defined bound since ι can not be made arbitrarily small as previously defined by increasing $\underline{q}$.

APPENDIX: B
PROOF OF SUPPORTING ASSUMPTIONS (CH. 6)

B.1 ICL-based Parameter Estimate

To ensure convergence under a finite excitation condition using input-output data to satisfy Assumption 6.2, an ICL-based parameter update law can be used to update the estimates (cf., [58]) To facilitate the analysis, let $p = (p_z + p_\eta)$, then (5–2) and (6–4) can be represented as

$$\dot{\breve{x}}(t) = \breve{S}(x)\theta + \breve{G}(x(t), u(t)) + \breve{D}(t) + \breve{\varepsilon}(x(t)), \tag{B–1}$$

where $\breve{x}(t) \triangleq [k_1 z(t), \eta(t)]^T \in \mathbb{R}^{2\times n}$, $\breve{G}(x(t), u(t)) \triangleq \begin{bmatrix} 0_{n\times 1}, & g(s_2(x(t)))u(t) \end{bmatrix}^T \in \mathbb{R}^{2\times n}$, $\breve{S}(x) \triangleq \begin{bmatrix} S_z^T(x(t)) & 0_{1\times p_\eta} \\ 0_{1\times p_z} & S_\eta^T(x(t)) \end{bmatrix} \in \mathbb{R}^{2\times p}$, $\breve{D}(t) \triangleq \begin{bmatrix} k_1 d(t), & 0_{n\times 1} \end{bmatrix}^T \in \mathbb{R}^{2\times n}$, and $\breve{\varepsilon}(x(t)) \triangleq \begin{bmatrix} \varepsilon_z(x(t)) & \varepsilon_\eta(x(t)) \end{bmatrix}^T \in \mathbb{R}^{2\times n}$. Based on the ICL strategy in [58], for a time $t_i \in [\Delta t_\theta, t]$, (B–1) can be expressed as

$$\breve{x}(t_i) - \breve{x}(t_i - \Delta t_\theta) = \mathcal{S}_i\theta + \mathcal{G}_i + \mathcal{D}_i + \mathcal{E}_i \tag{B–2}$$

where $\Delta t_\theta \in \mathbb{R}_{>0}$ denotes an integration time-window, $\mathcal{G}_i \triangleq \int_{t_i-\Delta t_\theta}^{t_i} \breve{G}(x(\tau), u(\tau))\, d\tau$, $\mathcal{S}_i \triangleq \int_{t_i-\Delta t_\theta}^{t_i} \breve{S}(x(\tau))\, d\tau$, $\mathcal{E}_i \triangleq \int_{t_i-\Delta t_\theta}^{t_i} \breve{\varepsilon}(x(\tau))\, d\tau$, $\mathcal{D}_i = \int_{t_i-\Delta t_\theta}^{t_i} \breve{D}(\tau)\, d\tau$, and $\mathcal{E}_i \triangleq \int_{t_i-\Delta t_\theta}^{t_i} \varepsilon(x(\tau))\, d\tau$. Using this expression, a least-squares parameter update law is designed as

$$\dot{\hat{\theta}}(t) = \text{proj}\Big\{ k_{CL}\Gamma_\theta(t) \sum_{i=1}^{M} \mathcal{S}_i^T \big(\breve{x}(t_i) - \breve{x}(t_i - \Delta t_\theta) - \mathcal{G}_i - \mathcal{S}_i\hat{\theta}(t) \big) \Big\}, \tag{B–3}$$

$$\dot{\Gamma}_\theta(t) = \beta_\theta \Gamma_\theta(t) - k_{CL}\Gamma_\theta(t) \sum_{i=1}^{M} \mathcal{S}_i^T \mathcal{S}_i \Gamma_\theta(t), \tag{B–4}$$

where $k_{CL}, \beta_\theta \in \mathbb{R}_{>0}$ are user-defined gains, and $M \in \mathbb{Z}_{\geq \frac{p}{2}}$ denotes the number of collected data points. To show that the update laws in (B–3) and (B–4) satisfy Assumption 6.2, the following analysis is provided.

ICL Analysis

The summation $\sum_{i=1}^{M} \mathcal{S}_i^T \mathcal{S}_i$ is assumed to satisfy Assumption 5.4, which can be verified online by checking the minimum eigenvalue of $\sum_{i=1}^{M} \mathcal{S}_i^T \mathcal{S}_i$. Using Assumption 5.4 and a similar argument to [101, Corollary 4.3.2], provided $\lambda_{\min}\{\Gamma_\theta(t_0)\} > 0$, Γ_θ satisfies $\underline{\Gamma}_\theta I_p \leq \Gamma_\theta(t) \leq \overline{\Gamma}_\theta I_p$, where $\underline{\Gamma}_\theta, \overline{\Gamma}_\theta \in \mathbb{R}_{>0}$. Let $Z_\theta \triangleq \text{vec}\left(\tilde{\theta}\right)$ denote the vector of estimation errors and let $V_\theta : \mathbb{R}^{np} \times \mathbb{R}_{\geq 0} \to \mathbb{R}$ denote the candidate Lyapunov function

$$V_\theta(Z_\theta, t) \triangleq \frac{1}{2}\text{tr}\left(\tilde{\theta}^T \Gamma_\theta^{-1}(t)\tilde{\theta}\right). \tag{B–5}$$

Using the bounds on $\Gamma_\theta(t)$, (B–5) can be bounded as $\frac{1}{2\overline{\Gamma}_\theta}\|Z_\theta\|^2 \leq V_\theta(Z_\theta, t) \leq \frac{1}{2\underline{\Gamma}_\theta}\|Z_\theta\|^2$, for all $t \in \mathbb{R}_{\geq t_0}$ and $Z_\theta \in \mathbb{R}^{np}$.

Using Assumptions 5.3 and 6.1, $\mathcal{S}_i$, $\mathcal{D}_i$, and $\mathcal{E}_i$ can be bounded as $\sup_{t\in\mathbb{R}_{\geq 0}}\|\mathcal{S}_i\| \leq \overline{S}\Delta t_\theta$, $\sup_{t\in\mathbb{R}_{\geq 0}}\|\mathcal{D}_i\| \leq k_1\overline{d}\Delta t_\theta$, and $\sup_{t\in\mathbb{R}_{\geq 0}}\|\mathcal{E}_i\| \leq \overline{\varepsilon}\Delta t_\theta$, respectively. Taking the time-derivative of (B–5), substituting in (B–2)-(B–4), and using the fact that $0 \leq \lambda_{\min}\left\{\sum_{i=1}^{M} \mathcal{S}_i^T \mathcal{S}_i\right\}$ for $t < T_1$, yields

$$\dot{V}_\theta(Z_\theta, t) \leq -\frac{1}{2}k_{CL}c_{\theta 1}\|Z_\theta\|^2 + k_{CL}\nu_\theta\|Z_\theta\|,$$

where $c_{\theta 1} \triangleq \frac{\beta_\theta}{k_{CL}\overline{\Gamma}_\theta}$ and $\nu_\theta \triangleq N\overline{S}(\Delta t_\theta)^2\left(k_1\overline{d} + \overline{\varepsilon}\right)$. Completing the square, using the bounds on (B–5), and invoking the Comparison Lemma [104, Lemma 3.4] yields

$$V_\theta(t) \leq V_\theta(t_0)e^{-\frac{k_{CL}c_{\theta 1}}{2}\underline{\Gamma}_\theta(t-t_0)} + \left(1 - e^{-\frac{k_{CL}c_{\theta 1}}{2}\underline{\Gamma}_\theta(t-t_0)}\right)\frac{2}{c_{\theta 1}^2}\frac{\nu_\theta^2}{\underline{\Gamma}_\theta},$$

for all $t \in [t_0, T_1)$; and hence, $V_\theta(t) \leq V_\theta(t_0) + \frac{2}{c_{\theta 1}^2}\frac{\nu_\theta^2}{\underline{\Gamma}_\theta}$ for all $t \in \mathbb{R}_{\geq 0}$.

After gathering enough data such that Assumption 5.4 is satisfied, the time-derivative of (B–5), along with Assumption 5.4 and (B–2)-(B–4) are used to yield

$$\dot{V}_\theta(Z_\theta, t) \leq -\frac{1}{2}k_{CL}c_{\theta_2}\|Z_\theta\|^2 + k_{CL}\|Z_\theta\|\nu_\theta,$$

for all $t \geq T_1$, where $c_{\theta_2} \triangleq c_{\theta_1} + \lambda_1$. Using the Comparison Lemma [104, Lemma 3.4] and the bounds on (B–5) yields

$$\|Z_\theta\| \leq c_\Gamma \sqrt{(c_M e^{-\lambda_\theta (t-T_1)} + (1 - e^{-\lambda_\theta (t-T_1)})\, c_B)},$$

for all $t \geq T_1$, where $c_M \triangleq \left(\|Z_\theta(t_0)\|^2 + \frac{4\nu_\theta^2}{c_{\theta 1}^2} \right)$, $c_\Gamma \triangleq \sqrt{\frac{\overline{\Gamma}_\theta}{\underline{\Gamma}_\theta}}$, $\lambda_\theta \triangleq \frac{k_{CL} c_{\theta 2} \underline{\Gamma}_\theta}{2}$, and $c_B \triangleq \frac{4\upsilon_1^2}{c_{\theta 2}^2}$. Hence, the designed update law provides an exponential bound on the weight estimation errors and satisfies Assumption 6.2.

REFERENCES

[1] R. Kamalapurkar, J. Rosenfeld, and W. E. Dixon, "Efficient model-based reinforcement learning for approximate online optimal control," *Automatica*, vol. 74, pp. 247–258, Dec. 2016.

[2] G. S. Aoude, B. D. Luders, J. M. Joseph, N. Roy, and J. P. How, "Probabilistically safe motion planning to avoid dynamic obstacles with uncertain motion patterns," *Auton. Robot.*, vol. 35, no. 1, pp. 51–76, 2013.

[3] A. V. Rao, D. A. Benson, C. L. Darby, M. A. Patterson, C. Francolin, and G. T. Huntington, "Algorithm 902: GPOPS, A MATLAB software for solving multiple-phase optimal control problems using the Gauss pseudospectral method," *ACM Trans. Math. Softw.*, vol. 37, no. 2, pp. 1–39, 2010.

[4] K. Yang, S. K. Gan, and S. Sukkarieh, "An efficient path planning and control algorithm for RUAVs in unknown and cluttered environments," *J. Intell. Robot Syst.*, vol. 57, pp. 101–122, 2010.

[5] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Rob. R*, vol. 30, pp. 846–894, 2011.

[6] A. Alvarez, A. Caiti, and R. Onken, "Evolutionary path planning for autonomous underwater vehicles in a variable ocean," *IEEE J. Ocean. Eng.*, vol. 29, pp. 418–429, 2004.

[7] S. M. LaValle and P. Konkimalla, "Algorithms for computing numerical optimal feedback motion strategies," *Int. J. Robot. Res.*, vol. 20, pp. 729–752, 2001.

[8] C. Petres, Y. Pailhas, P. Patron, Y. Petillot, J. Evans, and D. Lane, "Path planning for autonomous underwater vehicles," *IEEE Trans. Robot.*, vol. 23, no. 2, pp. 331–341, 2007.

[9] A. Shum, K. Morris, and A. Khajepour, "Direction-dependent optimal path planning for autonomous vehicles," *Robot. and Auton. Syst.*, 2015.

[10] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin, "A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games," *IEEE Trans. Autom. Control*, vol. 50, no. 7, pp. 947–957, 2005.

[11] H. Zhang, D. Liu, Y. Luo, and D. Wang, *Adaptive Dynamic Programming for Control Algorithms and Stability*, ser. Communications and Control Engineering. London: Springer-Verlag, 2013.

[12] H. Zhang, L. Cui, and Y. Luo, "Near-optimal control for nonzero-sum differential games of continuous-time nonlinear systems using single-network adp," *IEEE Trans. Cybern.*, vol. 43, no. 1, pp. 206–216, 2013.

[13] 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 Trans. Neural Netw.*, vol. 22, no. 12, pp. 2226–2236, Dec. 2011.

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

[15] K. G. Vamvoudakis, M. F. Miranda, and J. P. Hespanha, "Asymptotically stable adaptive–optimal control algorithm with saturating actuators and relaxed persistence of excitation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 11, pp. 2386–2398, 2016.

[16] X. Yang, D. Liu, D. Wang, and H. Ma, "Constrained online optimal control for continuous-time nonlinear systems using neuro-dynamic programming," in *Proc. IEEE Chin. Control Conf.*, 2014, pp. 8717–8722.

[17] H. Modares, F. L. Lewis, and M.-B. Naghibi-Sistani, "Integral reinforcement learning and experience replay for adaptive optimal control of partially-unknown constrained-input continuous-time systems," *Automatica*, vol. 50, no. 1, pp. 193–202, 2014.

[18] R. Kamalapurkar, P. Walters, and W. E. Dixon, "Model-based reinforcement learning for approximate optimal regulation," *Automatica*, vol. 64, pp. 94–104, 2016.

[19] 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 Trans. Cybern.*, vol. 45, no. 7, pp. 1372–1385, 2015.

[20] D. Wang, D. Liu, Q. Zhang, and D. Zhao, "Data-based adaptive critic designs for nonlinear robust optimal control with uncertain dynamics," *IEEE Trans. Syst. Man Cybern., Syst.*, vol. 46, no. 11, pp. 1544–1555, 2016.

[21] R. Kamalapurkar, L. Andrews, P. Walters, and W. E. Dixon, "Model-based reinforcement learning for infinite-horizon approximate optimal tracking," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 3, pp. 753–758, 2017.

[22] G. Leitmann and J. Skowronski, "Avoidance control," *J. Optim. Theory App.*, vol. 23, no. 4, pp. 581–591, 1977.

[23] E. Rimon and D. Koditschek, "Exact robot navigation using artificial potential functions," *IEEE Trans. Robot. Autom.*, vol. 8, no. 5, pp. 501–518, Oct. 1992.

[24] D. E. Koditschek and E. Rimon, "Robot navigation functions on manifolds with boundary," *Adv. Appl. Math.*, vol. 11, pp. 412–442, Dec. 1990.

[25] Z. Kan, A. Dani, J. M. Shea, and W. E. Dixon, "Network connectivity preserving formation stabilization and obstacle avoidance via a decentralized controller," *IEEE Trans. Autom. Control*, vol. 57, no. 7, pp. 1827–1832, 2012.

[26] Z. Kan, J. R. Klotz, E. Doucette, J. Shea, and W. E. Dixon, "Decentralized rendezvous of nonholonomic robots with sensing and connectivity constraints," *ASME J. Dyn. Syst. Meas. Control*, vol. 139, no. 2, pp. 024 501–1–024 501–7, 2017.

[27] T.-H. Cheng, Z. Kan, J. A. Rosenfeld, and W. E. Dixon, "Decentralized formation control with connectivity maintenance and collision avoidance under limited and intermittent sensing," in *Proc. Am. Control Conf.*, 2014, pp. 3201–3206.

[28] E. J. Rodríguez-Seda, C. Tang, M. W. Spong, and D. M. Stipanović, "Trajectory tracking with collision avoidance for nonholonomic vehicles with acceleration constraints and limited sensing," *Int. J. of Robot. Res.*, vol. 33, no. 12, pp. 1569–1592, 2014.

[29] E. J. Rodríguez-Seda, D. M. Stipanović, and M. W. Spong, "Guaranteed collision avoidance for autonomous systems with acceleration constraints and sensing uncertainties," *J. Optim. Theory Appl.*, vol. 168, no. 3, pp. 1014–1038, 2016.

[30] E. J. Rodríguez-Seda and M. W. Spong, "Guaranteed safe motion of multiple lagrangian systems with limited actuation," in *Proc. IEEE Conf. Decis. Control*, 2012, pp. 2773–2780.

[31] D. M. Stipanović, P. F. Hokayem, M. W. Spong, and D. D. Šiljak, "Cooperative avoidance control for multiagent systems," *J. Dyn. Sys., Meas., and Control*, vol. 129, no. 5, pp. 699–707, 2007.

[32] P. Walters, R. Kamalapurkar, and W. E. Dixon, "Approximate optimal online continuous-time path-planner with static obstacle avoidance," in *Proc. IEEE Conf. Decis. Control*, 2015, pp. 650–655.

[33] J. F. Fisac, M. Chen, C. J. Tomlin, and S. S. Sastry, "Reach-avoid problems with time-varying dynamics, targets and constraints," in *Proc. Int. Conf. Hybrid Syst.: Comp. Control*. ACM, 2015, pp. 11–20.

[34] J. Ding, E. Li, H. Huang, and C. J. Tomlin, "Reachability-based synthesis of feedback policies for motion planning under bounded disturbances," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 2160–2165.

[35] R. Takei, H. Huang, J. Ding, and C. J. Tomlin, "Time-optimal multi-stage motion planning with guaranteed collision avoidance via an open-loop game formulation," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2012, pp. 323–329.

[36] J. von Neumann and O. Morgenstern, *Theory of Games and Economic Behavior*. Princeton University Press, 1980.

[37] R. Isaacs, *Differential Games: A Mathematical Theory with Applications to Warfare and Pursuit, Control and Optimization*, ser. Dover Books on Mathematics. Dover Publications, 1999.

[38] T. H. Chung, G. A. Hollinger, and V. Isler, "Search and pursuit-evasion in mobile robotics," *Autonomous robots*, vol. 31, no. 4, p. 299, 2011.

[39] S. S. Kumkov, S. Le Ménec, and V. S. Patsko, "Zero-sum pursuit-evasion differential games with many objects: survey of publications," *Dyn. Games Appl.*, vol. 7, no. 4, pp. 609–633, 2017.

[40] R. Vidal, O. Shakernia, H. Kim, D. Shim, and S. Sastry, "Probabilistic pursuit-evasion games: theory, implementation, and experimental evaluation," *IEEE Trans. Robot. and Autom.*, vol. 18, no. 5, pp. 662–669, Oct. 2002.

[41] A. D. Khalafi and M. R. Toroghi, "Capture zone in the herding pursuit evasion games," *Appl. Math. Sci.*, vol. 5, no. 39, pp. 1935–1945, 2011.

[42] P. Kachroo, S. A. Shedied, J. S. Bay, and H. Vanlandingham, "Dynamic programming solution for a class of pursuit evasion problems: the herding problem," *IEEE Trans. Syst. Man Cybern.*, vol. 31, no. 1, pp. 35–41, Feb. 2001.

[43] M. Chen, Z. Zhou, and C. J. Tomlin, "Multiplayer reach-avoid games via pairwise outcomes," *IEEE Trans. Autom. Control*, vol. 62, no. 3, pp. 1451–1457, 2017.

[44] J. Chen, W. Zha, Z. Peng, and D. Gu, "Multi-player pursuit–evasion games with one superior evader," *Automatica*, vol. 71, pp. 24–32, 2016.

[45] M. V. Ramana and M. Kothari, "Pursuit-evasion games of high speed evader," *J. Intell. Rob. Syst.*, vol. 85, no. 2, pp. 293–306, 2017.

[46] F. Yan, J. Jiang, K. Di, Y. Jiang, and Z. Hao, "Multiagent pursuit-evasion problem with the pursuers moving at uncertain speeds," *J. Intell. Rob. Syst.*, pp. 1–17, 2018.

[47] R. Isaacs, *Differential Games*. John Wiley, 1967.

[48] E. Garcia, D. W. Casbeer, and M. Pachter, "Active target defence differential game: fast defender case," *IET Control Theory Appl.*, vol. 11, no. 17, pp. 2985–2993, 2017.

[49] ——, "Design and analysis of state-feedback optimal strategies for the differential game of active defense," *IEEE Trans Autom. Contol*, 2018.

[50] H. Huang, J. Ding, W. Zhang, and C. J. Tomlin, "Automation-assisted capture-the-flag: A differential game approach," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 3, pp. 1014–1028, 2015.

[51] A. S. Gadre, "Learning strategies in multi-agent systems-applications to the herding problem," *M.S. thesis, Dept. Elect. Comput. Eng., Virginia Tech, Blacksburg, VA, USA*, 2001.

[52] Z. Lu, "Cooperative optimal path planning for herding problems," Ph.D. dissertation, Texas A&M University, 2006.

[53] S. A. Shedied, "Optimal control for a two player dynamic pursuit evasion game; the herding problem," Ph.D. dissertation, Virginia Polytechnique Institute, 2002.

[54] R. Licitra, Z. Hutcheson, E. Doucette, and W. E. Dixon, "Single agent herding of n-agents: A switched systems approach," in *IFAC World Congr.*, 2017, pp. 14 374–14 379.

[55] R. Licitra, Z. I. Bell, E. Doucette, and W. E. Dixon, "Single agent indirect herding of multiple targets: A switched adaptive control approach," *IEEE Control Syst. Lett.*, vol. 2, no. 1, pp. 127–132, January 2018.

[56] A. Pierson and M. Schwager, "Controlling noncooperative herds with robotic herders," *IEEE Trans. Robot.*, vol. 34, no. 2, pp. 517–525, 2018.

[57] M. Bacon and N. Olgac, "Swarm herding using a region holding sliding mode controller," *J. Vib. Control*, vol. 18, no. 7, pp. 1056–1066, 2012.

[58] A. Parikh, R. Kamalapurkar, and W. E. Dixon, "Integral concurrent learning: Adaptive control with parameter convergence using finite excitation," *Int J Adapt Control Signal Process*, to appear.

[59] P. E. Gill, W. Murray, and M. A. Saunders, "Snopt: An sqp algorithm for large-scale constrained optimization," *SIAM REV.*, vol. 47, no. 1, pp. 99–131, 2005.

[60] P. E. Gill, E. Wong, W. Murray, and M. A. Saunders, "User's guide for snopt version 7: Software for large-scale nonlinear programming," https://ccom.ucsd.edu/ optimizers/static/pdfs/snopt7-7.pdf, 2015, department of Mathematics, University of California, San Diego, La Jolla, CA 92093-0112.

[61] R. Kamalapurkar, P. S. Walters, J. A. Rosenfeld, and W. E. Dixon, *Reinforcement learning for optimal feedback control: A Lyapunov-based approach*. Springer, 2018.

[62] K. G. Vamvoudakis, H. Modares, B. Kiumarsi, and F. L. Lewis, "Game theory-based control system algorithms with real-time reinforcement learning: How to solve multiplayer games online," *IEEE Control Syst.*, vol. 37, no. 1, pp. 33–52, 2017.

[63] B. Kiumarsi, K. G. Vamvoudakis, H. Modares, and F. L. Lewis, "Optimal and autonomous control using reinforcement learning: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, 2017.

[64] 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.

[65] Q.-Y. Fan and G.-H. Yang, “Nearly optimal sliding mode fault-tolerant control for affine nonlinear systems with state constraints,” *Neurocomputing*, vol. 216, pp. 78–88, 2016.

[66] H. Modares, F. L. Lewis, and M.-B. N. Sistani, “Online solution of nonquadratic two-player zero-sum games arising in the H_∞ control of constrained input systems,” *Int. J. Adapt. Control Signal Process.*, vol. 28, no. 3-5, pp. 232–254, 2014.

[67] P. Walters, R. Kamalapurkar, L. Andrews, and W. E. Dixon, “Online approximate optimal path-following for a mobile robot,” in *Proc. IEEE Conf. Decis. Control*, 2014, pp. 4536–4541.

[68] T. Dierks, B. Thumati, and S. Jagannathan, “Optimal control of unknown affine nonlinear discrete-time systems using offline-trained neural networks with proof of convergence,” *Neural Netw.*, vol. 22, no. 5-6, pp. 851–860, 2009.

[69] K. Doya, “Reinforcement learning in continuous time and space,” *Neural Comput.*, vol. 12, no. 1, pp. 219–245, 2000.

[70] J. A. Rosenfeld, R. Kamalapurkar, and W. E. Dixon, “State following (StaF) kernel functions for function approximation Part I: Theory and motivation,” in *Proc. Am. Control Conf.*, 2015, pp. 1217–1222.

[71] K. G. Vamvoudakis and F. L. Lewis, “Online synchronous policy iteration method for optimal control,” in *Recent Advances in Intelligent Control Systems*, W. Yu, Ed. London, UK: Springer, 2009, pp. 357–374.

[72] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 1998.

[73] D. Vrabie and F. L. Lewis, “Neural network approach to continuous-time direct adaptive optimal control for partially unknown nonlinear systems,” *Neural Netw.*, vol. 22, no. 3, pp. 237–246, 2009.

[74] P. Mehta and S. Meyn, “Q-learning and pontryagin’s minimum principle,” in *Proc. IEEE Conf. Decis. Control*, Dec. 2009, pp. 3598–3605.

[75] Q.-Y. Fan and G.-H. Yang, “Adaptive actor–critic design-based integral sliding-mode control for partially unknown nonlinear systems with input disturbances,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 1, pp. 165–177, 2016.

[76] G. Chowdhary, “Concurrent learning for convergence in adaptive control without persistency of excitation,” Ph.D. dissertation, Georgia Institute of Technology, Dec. 2010.

[77] G. Chowdhary and E. Johnson, “A singular value maximizing data recording algorithm for concurrent learning,” in *Proc. Am. Control Conf.*, 2011, pp. 3547–3552.

[78] 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. 89–92, Jan. 2013.

[79] Y. Lv, J. Na, Q. Yang, X. Wu, and Y. Guo, “Online adaptive optimal control for continuous-time nonlinear systems with completely unknown dynamics,” *Int. J. Control*, vol. 89, no. 1, pp. 99–112, 2016.

[80] G. Chowdhary and E. Johnson, “Concurrent learning for convergence in adaptive control without persistency of excitation,” in *Proc. IEEE Conf. Decis. Control*, 2010, pp. 3674–3679.

[81] R. Kamalapurkar, B. Reish, G. Chowdhary, and W. E. Dixon, “Concurrent learning for parameter estimation using dynamic state-derivative estimators,” *IEEE Trans. Autom. Control*, vol. 62, no. 7, pp. 3594–3601, July 2017.

[82] Z. I. Bell, H.-Y. Chen, A. Parikh, and W. E. Dixon, “Single scene and path reconstruction with a monocular camera using integral concurrent learning,” in *Proc. IEEE Conf. Decis. Control*, 2017, pp. 3670–3675.

[83] Z. Bell, A. Parikh, J. Nezvadovitz, and W. E. Dixon, “Adaptive control of a surface marine craft with parameter identification using integral concurrent learning,” in *Proc. IEEE Conf. Decis. Control*, 2016, pp. 389–394.

[84] S. B. Roy, S. Bhasin, and I. N. Kar, “Combined mrac for unknown mimo lti systems with parameter convergence,” *IEEE Trans. Autom. Control*, vol. 63, no. 1, pp. 283–290, Jan. 2018.

[85] K. G. Vamvoudakis and J. P. Hespanha, “Cooperative q-learning for rejection of persistent adversarial inputs in networked linear quadratic systems,” *IEEE Trans. Autom. Control*, vol. 63, no. 4, pp. 1018–1031, 2018.

[86] H. Modares, F. L. Lewis, W. Kang, and A. Davoudi, “Optimal synchronization of heterogeneous nonlinear systems with unknown dynamics,” *IEEE Trans. Autom. Control*, vol. 63, no. 1, pp. 117–131, 2018.

[87] D. Wang, D. Liu, C. Mu, and H. Ma, “Decentralized guaranteed cost control of interconnected systems with uncertainties: a learning-based optimal control strategy,” *Neurocomputing*, vol. 214, pp. 297–306, 2016.

[88] R. Kamalapurkar, J. R. Klotz, P. Walters, and W. E. Dixon, “Model-based reinforcement learning for differential graphical games,” *IEEE Trans. Control Netw. Syst.*, vol. 5, no. 1, pp. 423–433, 2018.

[89] J. Sun, C. Liu, and Q. Ye, "Robust differential game guidance laws design for uncertain interceptor-target engagement via adaptive dynamic programming," *Int. J. Control*, vol. 90, no. 5, pp. 990–1004, 2017.

[90] G. Wen, S. S. Ge, and F. Tu, "Optimized backstepping for tracking control of strict-feedback systems," *IEEE Trans. Neural Netw. Learn. Syst.*, 2018.

[91] Z. Wang, X. Liu, K. Liu, S. Li, and H. Wang, "Backstepping-based Lyapunov function construction using approximate dynamic programming and sum of square techniques," *IEEE Trans. Cybern.*, vol. 47, no. 10, pp. 3393–3403, 2017.

[92] Y. Lin and E. D. Sontag, "A universal formula for stabilization with bounded controls," *Systems & Control Letters*, vol. 16, no. 6, pp. 393–397, 1991.

[93] P. Deptula, Z. I. Bell, F. Zegers, R. Licitra, and W. E. Dixon, "Single agent indirect herding via approximate dynamic programming," in *Proc. IEEE Conf. Decis. Control*, Dec. 2018, pp. 7136–7141.

[94] W. Lee and D. Kim, "Autonomous shepherding behaviors of multiple target steering robots," *Sens.*, vol. 17, no. 12, p. 2729, 2017.

[95] J. A. Rosenfeld, R. Kamalapurkar, and W. E. Dixon, "The state following (staf) approximation method," *IEEE Trans. on Neural Netw. Learn. Syst.*, to appear.

[96] P. Deptula, J. Rosenfeld, R. Kamalapurkar, and W. E. Dixon, "Approximate dynamic programming: Combining regional and local state following approximations," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 6, pp. 2154–2166, June 2018.

[97] J. A. Farrell and M. M. Polycarpou, *Adaptive approximation based control: Unifying neural, fuzzy and traditional adaptive approximation approaches*, ser. Adaptive and Learning Systems for Signal Processing, Communications and Control Series. John Wiley & Sons, 2006, vol. 48.

[98] F. L. Lewis, R. Selmic, and J. Campos, *Neuro-Fuzzy Control of Industrial Systems with Actuator Nonlinearities*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2002.

[99] S. Basu Roy, S. Bhasin, and I. N. Kar, "Composite adaptive control of uncertain euler-lagrange systems with parameter convergence without pe condition," *Asian J. Control*, 2019.

[100] N. Cho, H.-S. Shin, Y. Kim, and A. Tsourdos, "Composite model reference adaptive control with parameter convergence under finite excitation," *IEEE Trans. Autom. Control*, vol. 63, no. 3, pp. 811–818, Mar. 2017.

[101] P. Ioannou and J. Sun, *Robust Adaptive Control*. Prentice Hall, 1996.

[102] P. S. Walters, "Guidance and control of marine craft: An adaptive dynamic programming approach," Ph.D. dissertation, University of Florida, 2015.

[103] V. Stepanyan and N. Hovakimyan, "Robust adaptive observer design for uncertain systems with bounded disturbances," *IEEE Trans. Neur. Netw.*, vol. 18, no. 5, pp. 1392–1403, 2007.

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

[105] P. M. Patre, W. MacKunis, K. Kaiser, and W. E. Dixon, "Asymptotic tracking for uncertain dynamic systems via a multilayer neural network feedforward and RISE feedback control structure," *IEEE Trans. Autom. Control*, vol. 53, no. 9, pp. 2180–2185, 2008.

[106] 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, Jan. 2015.

[107] P. Deptula, Z. I. Bell, E. Doucette, J. W. Curtis, and W. E. Dixon, "Data-based reinforcement learning approximate optimal control for an uncertain nonlinear system with partial loss of control effectiveness," in *Proc. Am. Control Conf.*, 2018, pp. 2521–2526.

[108] Z. Bell, P. Deptula, H.-Y. Chen, E. Doucette, and W. E. Dixon, "Velocity and path reconstruction of a moving object using a moving camera," in *Proc. Am. Control Conf.*, 2018, pp. 5256–5261.

[109] A. Parikh, R. Kamalapurkar, and W. E. Dixon, "Target tracking in the presence of intermittent measurements via motion model learning," *IEEE Trans. Robot.*, vol. 34, no. 3, pp. 805–819, 2018.

[110] "bebop_autonomy library," http://bebop-autonomy.readthedocs.io.

[111] P. Deptula, Z. I. Bell, F. M. Zegers, R. A. Licitra, and W. E. Dixon, "Approximate optimal influence over an agent through an uncertain interaction dynamic experiment," https://youtu.be/JeK4jTDuImo, Feb. 2019.

[112] Q. Jiao, H. Modares, S. Xu, F. L. Lewis, and K. G. Vamvoudakis, "Multi-agent zero-sum differential graphical games for disturbance rejection in distributed control," *Automatica*, vol. 69, pp. 24–34, 2016.

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

[114] S. Xue, B. Luo, and D. Liu, "Event-triggered adaptive dynamic programming for zero-sum game of partially unknown continuous-time nonlinear systems," *IEEE Trans. Syst. Man Cybern., Syst.*, no. 99, pp. 1–11, 2018.

[115] D. Wang, H. He, and D. Liu, "Improving the critic learning for event-based nonlinear H_∞ control design," *IEEE Trans. Cybern.*, vol. 47, no. 10, pp. 3417–3428, 2017.

[116] W. E. Dixon, A. Behal, D. M. Dawson, and S. Nagarkatti, *Nonlinear Control of Engineering Systems: A Lyapunov-Based Approach*. Birkhauser: Boston, 2003.

[117] G. Chowdhary, M. Mühlegg, J. How, and F. Holzapfel, "Concurrent learning adaptive model predictive control," in *Advances in Aerospace Guidance, Navigation and Control*, Q. Chu, B. Mulder, D. Choukroun, E.-J. van Kampen, C. de Visser, and G. Looye, Eds. Springer Berlin Heidelberg, 2013, pp. 29–47.

[118] G. Chowdhary, M. Mühlegg, and E. Johnson, "Exponential parameter and tracking error convergence guarantees for adaptive controllers without persistency of excitation," *Int. J. Control*, vol. 87, no. 8, pp. 1583–1603, 2014.

[119] H. Jiang, H. Zhang, Y. Luo, and J. Han, "Neural-network-based robust control schemes for nonlinear multiplayer systems with uncertainties via adaptive dynamic programming," *IEEE Trans. Syst. Man Cybern., Syst*, Mar. 2018.

[120] K. G. Vamvoudakis and F. L. Lewis, "Multi-player non-zero-sum games: Online adaptive learning solution of coupled hamilton-jacobi equations," *Automatica*, vol. 47, pp. 1556–1569, 2011.

[121] K. G. Vamvoudakis, "Non-zero sum nash q-learning for unknown deterministic continuous-time linear systems," *Automatica*, vol. 61, pp. 274–281, 2015.

[122] S. A. Nivison and P. Khargonekar, "Improving long-term learning of model reference adaptive controllers for flight applications: A sparse neural network approach," in *Proc. AIAA Guid. Navig. Control Conf.*, Jan. 2017.

[123] S. A. Nivison and P. P. Khargonekar, "Development of a robust deep recurrent neural network controller for flight applications," in *Proc. Am. Control Conf.* IEEE, 2017, pp. 5336–5342.

[124] I. Papusha, J. Fu, U. Topcu, and R. M. Murray, "Automata theory meets approximate dynamic programming: Optimal control with temporal logic constraints," in *Proc. Conf. Dec. Control*. IEEE, 2016, pp. 434–440.

BIOGRAPHICAL SKETCH

Patryk Deptuła was born in 1992 in Ostrołęka, Poland. He received his bachelor's degree in mechanical engineering in 2014 from Central Connecticut State University, New Britain, CT. In 2015, Patryk was awarded the Henry Barnard Scholar Award in the State of Connecticut, as well as the Dean's Citation and Departmental Honors awards in the School of Engineering, Science, and Technology at Central Connecticut State University. Between the time that Patryk graduated from his undergraduate university to starting his graduate studies, he worked as an engineer in the Control and Diagnostic Systems group at Belcan, LLC in Windsor, CT, where he performed verification and validation of aircraft engine software. That same year, Patryk joined the Nonlinear Controls and Robotics (NCR) group under the advisement of Dr. Warren W. Dixon to pursue his doctoral studies. He received his master's degree in Mechanical Engineering from University of Florida in 2017 under the supervision of Dr. Warren E. Dixon. Patryk's research interests include learning-based and adaptive control for robotic, autonomous, and dynamical systems.